# Table of Contents

Foreword 0

**Part I Welcome to Phidgets**

**Part II Introduction** 14

1. Understanding Phidgets ................................................................. 14
2. Platform Support ........................................................................ 15

**Part III Programming Concept** 17

1. Getting Started ........................................................................... 17
2. Event Handler ............................................................................ 21
3. Multiple Devices .......................................................................... 24

**Part IV Phidgets Common** 26

1. ErrorHandler.vi ........................................................................ 26
2. PhidgetClose.vi ......................................................................... 27
3. PhidgetDelete.vi ......................................................................... 28
4. PhidgetEventCloseOnError.vi ..................................................... 29
5. PhidgetEventCreateOnError.vi ..................................................... 30
6. PhidgetEventExeOnError.vi ......................................................... 31
7. PhidgetGetDeviceClass.vi ............................................................ 33
8. PhidgetGetDeviceID.vi ................................................................. 34
9. PhidgetGetDeviceLabel.vi ............................................................. 36
10. PhidgetGetDeviceName.vi ............................................................ 37
11. PhidgetGetDeviceType.vi ............................................................. 38
12. PhidgetGetDeviceVersion.vi ....................................................... 39
13. PhidgetGetServerAddress.vi ....................................................... 41
14. PhidgetGetServerStatus.vi .......................................................... 42
15. PhidgetGetServiceID.vi ............................................................... 43
16. PhidgetLibraryVersion.vi ............................................................. 45
17. PhidgetOpen.vi ......................................................................... 46
18. PhidgetOpenRemote.vi ................................................................. 47
19. PhidgetOpenRemoteIP.vi .............................................................. 49
20. PhidgetSetDeviceLabel.vi ............................................................ 51
21. Subvi ....................................................................................... 52
   _AttachCHK.vi ........................................................................ 52
   _ChkError.vi ............................................................................ 53
   _Close.vi ............................................................................... 55
   _Delete.vi ............................................................................. 56

© 2014 Phidgets Inc.
Part V  Specific Modules  

1  Accelerometer  

2  AdvancedServo

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>_Open.vi</td>
<td>57</td>
</tr>
<tr>
<td>_OpenRemote.vi</td>
<td>58</td>
</tr>
<tr>
<td>_OpenRemoteIP.vi</td>
<td>59</td>
</tr>
<tr>
<td>_SerialReturn.vi</td>
<td>61</td>
</tr>
<tr>
<td>_WaitAttach.vi</td>
<td>62</td>
</tr>
<tr>
<td>EventCloseIntDouble.vi</td>
<td>63</td>
</tr>
<tr>
<td>EventCloseIntInt.vi</td>
<td>65</td>
</tr>
<tr>
<td>EventExeIntDouble.vi</td>
<td>66</td>
</tr>
<tr>
<td>EventExeIntInt.vi</td>
<td>68</td>
</tr>
<tr>
<td>AcceAxisCount.vi</td>
<td>70</td>
</tr>
<tr>
<td>AcceCreate.vi</td>
<td>70</td>
</tr>
<tr>
<td>AcceEventClose.vi</td>
<td>71</td>
</tr>
<tr>
<td>AcceEventCreate.vi</td>
<td>72</td>
</tr>
<tr>
<td>AcceEventExe.vi</td>
<td>74</td>
</tr>
<tr>
<td>AcceGetData.vi</td>
<td>75</td>
</tr>
<tr>
<td>AcceGetMax.vi</td>
<td>76</td>
</tr>
<tr>
<td>AcceGetMin.vi</td>
<td>78</td>
</tr>
<tr>
<td>AcceGetTrigger.vi</td>
<td>79</td>
</tr>
<tr>
<td>AcceOpen.vi</td>
<td>81</td>
</tr>
<tr>
<td>AcceSetTrigger.vi</td>
<td>82</td>
</tr>
<tr>
<td>AdvServoCount.vi</td>
<td>85</td>
</tr>
<tr>
<td>AdvServoCreate.vi</td>
<td>86</td>
</tr>
<tr>
<td>AdvServoEventClose.vi</td>
<td>87</td>
</tr>
<tr>
<td>AdvServoEventCreateCrtChange.vi</td>
<td>89</td>
</tr>
<tr>
<td>AdvServoEventCreatePosChange.vi</td>
<td>90</td>
</tr>
<tr>
<td>AdvServoEventCreateVelChange.vi</td>
<td>91</td>
</tr>
<tr>
<td>AdvServoEventExe.vi</td>
<td>93</td>
</tr>
<tr>
<td>AdvServoGetAcce.vi</td>
<td>94</td>
</tr>
<tr>
<td>AdvServoGetAcceMax.vi</td>
<td>96</td>
</tr>
<tr>
<td>AdvServoGetAcceMin.vi</td>
<td>97</td>
</tr>
<tr>
<td>AdvServoGetCurrent.vi</td>
<td>99</td>
</tr>
<tr>
<td>AdvServoGetEngaged.vi</td>
<td>100</td>
</tr>
<tr>
<td>AdvServoGetPos.vi</td>
<td>102</td>
</tr>
<tr>
<td>AdvServoGetPosMax.vi</td>
<td>103</td>
</tr>
<tr>
<td>AdvServoGetPosMin.vi</td>
<td>104</td>
</tr>
<tr>
<td>AdvServoGetRampingState.vi</td>
<td>106</td>
</tr>
<tr>
<td>AdvServoGetServoType.vi</td>
<td>107</td>
</tr>
<tr>
<td>AdvServoGetVel.vi</td>
<td>109</td>
</tr>
<tr>
<td>AdvServoGetVelLmt.vi</td>
<td>110</td>
</tr>
<tr>
<td>AdvServoGetVelMax.vi</td>
<td>112</td>
</tr>
<tr>
<td>AdvServoGetVelMin.vi</td>
<td>113</td>
</tr>
<tr>
<td>AdvServoOpen.vi</td>
<td>115</td>
</tr>
<tr>
<td>AdvServoSetAcce.vi</td>
<td>116</td>
</tr>
<tr>
<td>AdvServoSetEngaged.vi</td>
<td>117</td>
</tr>
<tr>
<td>AdvServoSetPos.vi</td>
<td>119</td>
</tr>
<tr>
<td>AdvServoSetPosMax.vi</td>
<td>120</td>
</tr>
<tr>
<td>AdvServoSetPosMin.vi</td>
<td>122</td>
</tr>
<tr>
<td>AdvServoSetRampingState.vi</td>
<td>123</td>
</tr>
<tr>
<td>AdvServoSetServoParameters.vi</td>
<td>125</td>
</tr>
</tbody>
</table>
### Encoder

- EncoderCreate.vi .......................................................... 131
- EncoderGetEnabled.vi .................................................... 132
- EncoderGetVoltage.vi ..................................................... 133
- EncoderGetVoltageMax.vi ................................................ 135
- EncoderGetVoltageMin.vi ................................................ 136
- EncoderOpen.vi ............................................................. 138
- EncoderOutputCount.vi ................................................... 139
- EncoderSetEnabled.vi ..................................................... 141
- EncoderSetVoltage.vi ..................................................... 142

### Bridge

- BridgeCreate.vi ............................................................ 144
- BridgeEventClose.vi ....................................................... 145
- BridgeEventCreate.vi ..................................................... 146
- BridgeEventExe.vi ........................................................ 147
- BridgeGetDataRate.vi ...................................................... 149
- BridgeGetDataRateMax.vi ............................................... 150
- BridgeGetDataRateMin.vi ............................................... 151
- BridgeGetEnable.d.vi ...................................................... 153
- BridgeGetGain.vi .......................................................... 154
- BridgeGetValue.vi ........................................................ 156
- BridgeGetValueMax.vi ..................................................... 157
- BridgeGetValueMin.vi ..................................................... 159
- BridgeInputCount.vi ....................................................... 160
- BridgeOpen.vi .............................................................. 161
- BridgeSetDataRate.vi ...................................................... 163
- BridgeSetEnabled.vi ....................................................... 164
- BridgeSetGain.vi .......................................................... 166

### Analog

- AnalogCreate.vi ........................................................... 131
- AnalogGetEnabled.vi ..................................................... 132
- AnalogGetVoltage.vi ..................................................... 133
- AnalogGetVoltageMax.vi ................................................ 135
- AnalogGetVoltageMin.vi ................................................ 136
- AnalogOpen.vi ............................................................. 138
- AnalogOutputCount.vi ................................................... 139
- AnalogSetEnabled.vi ..................................................... 141

### Frequency

- FreqCreate.vi .............................................................. 190
- FreqEventClose.vi ........................................................ 191
- FreqEventCreate.vi ....................................................... 192
- FreqEventExe.vi ........................................................... 193
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFSetOutputState.vi</td>
<td>195</td>
</tr>
<tr>
<td>IFGetTrig.vi</td>
<td>196</td>
</tr>
<tr>
<td>IFGetOutputState.vi</td>
<td>197</td>
</tr>
<tr>
<td>IFGetInputState.vi</td>
<td>198</td>
</tr>
<tr>
<td>IFGetInputCount.vi</td>
<td>200</td>
</tr>
<tr>
<td>IFGetDateTime.vi</td>
<td>201</td>
</tr>
<tr>
<td>IFGetCount.vi</td>
<td>204</td>
</tr>
<tr>
<td>FreqSetOutputState.vi</td>
<td>202</td>
</tr>
<tr>
<td>FreqGetTrig.vi</td>
<td>204</td>
</tr>
<tr>
<td>FreqGetOutputState.vi</td>
<td>205</td>
</tr>
<tr>
<td>FreqGetInputState.vi</td>
<td>207</td>
</tr>
<tr>
<td>FreqGetInputCount.vi</td>
<td>208</td>
</tr>
<tr>
<td>FreqGetDateTime.vi</td>
<td>208</td>
</tr>
<tr>
<td>FreqGetCount.vi</td>
<td>210</td>
</tr>
<tr>
<td>FreqSetEnabled.vi</td>
<td>211</td>
</tr>
<tr>
<td>FreqSetFilter.vi</td>
<td>212</td>
</tr>
<tr>
<td>FreqSetTimeout.vi</td>
<td>214</td>
</tr>
</tbody>
</table>

7 GPS

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPSCreate.vi</td>
<td>215</td>
</tr>
<tr>
<td>GPSEventCloseFixStatus.vi</td>
<td>216</td>
</tr>
<tr>
<td>GPSEventClosePosition.vi</td>
<td>218</td>
</tr>
<tr>
<td>GPSEventCreateFixStatus.vi</td>
<td>219</td>
</tr>
<tr>
<td>GPSEventCreatePosition.vi</td>
<td>220</td>
</tr>
<tr>
<td>GPSEventFixStatus.vi</td>
<td>221</td>
</tr>
<tr>
<td>GPSEventFixPosition.vi</td>
<td>223</td>
</tr>
<tr>
<td>GPSetAltitude.vi</td>
<td>225</td>
</tr>
<tr>
<td>GPSetDate.vi</td>
<td>226</td>
</tr>
<tr>
<td>GPSetFixStatus.vi</td>
<td>228</td>
</tr>
<tr>
<td>GPSetHeading.vi</td>
<td>229</td>
</tr>
<tr>
<td>GPSetLatitude.vi</td>
<td>231</td>
</tr>
<tr>
<td>GPSetLongitude.vi</td>
<td>232</td>
</tr>
<tr>
<td>GPSetTime.vi</td>
<td>233</td>
</tr>
<tr>
<td>GPSetVelocity.vi</td>
<td>235</td>
</tr>
<tr>
<td>GPSSOpen.vi</td>
<td>236</td>
</tr>
</tbody>
</table>

8 InterfaceKit

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICreate.vi</td>
<td>238</td>
</tr>
<tr>
<td>IFEventClose.vi</td>
<td>239</td>
</tr>
<tr>
<td>IFEventCreateInput.vi</td>
<td>240</td>
</tr>
<tr>
<td>IFEventCreateOutput.vi</td>
<td>241</td>
</tr>
<tr>
<td>IFEventCreateSensor.vi</td>
<td>243</td>
</tr>
<tr>
<td>IFEventExe.vi</td>
<td>244</td>
</tr>
<tr>
<td>IFGetDataRate.vi</td>
<td>246</td>
</tr>
<tr>
<td>IFGetDataRateMax.vi</td>
<td>247</td>
</tr>
<tr>
<td>IFGetDataRateMin.vi</td>
<td>249</td>
</tr>
<tr>
<td>IFGetInputCount.vi</td>
<td>250</td>
</tr>
<tr>
<td>IFGetInputState.vi</td>
<td>251</td>
</tr>
<tr>
<td>IFGetOutputCount.vi</td>
<td>253</td>
</tr>
<tr>
<td>IFGetOutputState.vi</td>
<td>254</td>
</tr>
<tr>
<td>IFGetRatio.vi</td>
<td>256</td>
</tr>
<tr>
<td>IFGetSensorCount.vi</td>
<td>257</td>
</tr>
<tr>
<td>IFGetSensorValue.vi</td>
<td>259</td>
</tr>
<tr>
<td>IFGetSensorValueRaw.vi</td>
<td>260</td>
</tr>
<tr>
<td>IFGetTrig.vi</td>
<td>261</td>
</tr>
<tr>
<td>IFOpen.vi</td>
<td>263</td>
</tr>
<tr>
<td>IFSetDataRate.vi</td>
<td>264</td>
</tr>
<tr>
<td>IFSetOutputState.vi</td>
<td>266</td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>IFSetRatio.vi</td>
<td>267</td>
</tr>
<tr>
<td>IFSetTrig.vi</td>
<td>269</td>
</tr>
<tr>
<td>IR</td>
<td>270</td>
</tr>
<tr>
<td>IRCreate.vi</td>
<td>270</td>
</tr>
<tr>
<td>IREventCloseOnCode.vi</td>
<td>271</td>
</tr>
<tr>
<td>IREventCloseOnLearn.vi</td>
<td>272</td>
</tr>
<tr>
<td>IREventCloseOnRawData.vi</td>
<td>274</td>
</tr>
<tr>
<td>IREventCreateOnCode.vi</td>
<td>275</td>
</tr>
<tr>
<td>IREventCreateOnLearn.vi</td>
<td>276</td>
</tr>
<tr>
<td>IREventCreateOnRawData.vi</td>
<td>278</td>
</tr>
<tr>
<td>IREventExeOnCode.vi</td>
<td>279</td>
</tr>
<tr>
<td>IREventExeOnLearn.vi</td>
<td>281</td>
</tr>
<tr>
<td>IREventExeOnRawData.vi</td>
<td>284</td>
</tr>
<tr>
<td>IRGetLastCode.vi</td>
<td>286</td>
</tr>
<tr>
<td>IRGetLastLearnedCode.vi</td>
<td>287</td>
</tr>
<tr>
<td>IRGetRawData.vi</td>
<td>291</td>
</tr>
<tr>
<td>IROpen.vi</td>
<td>292</td>
</tr>
<tr>
<td>IRTransmit.vi</td>
<td>294</td>
</tr>
<tr>
<td>IRTransmitRaw.vi</td>
<td>297</td>
</tr>
<tr>
<td>IRTransmitRepeat.vi</td>
<td>299</td>
</tr>
<tr>
<td>LED</td>
<td>301</td>
</tr>
<tr>
<td>LEDCount.vi</td>
<td>301</td>
</tr>
<tr>
<td>LEDCreate.vi</td>
<td>302</td>
</tr>
<tr>
<td>LEDGetBrightness.vi</td>
<td>303</td>
</tr>
<tr>
<td>LEDGetCurrentLimit.vi</td>
<td>304</td>
</tr>
<tr>
<td>LEDGetCurrentLimitIndexed.vi</td>
<td>306</td>
</tr>
<tr>
<td>LEDGetVoltage.vi</td>
<td>307</td>
</tr>
<tr>
<td>LEDOpen.vi</td>
<td>309</td>
</tr>
<tr>
<td>LEDSetBrightness.vi</td>
<td>310</td>
</tr>
<tr>
<td>LEDSetCurrentLimit.vi</td>
<td>311</td>
</tr>
<tr>
<td>LEDSetCurrentLimitIndexed.vi</td>
<td>313</td>
</tr>
<tr>
<td>LEDSetVoltage.vi</td>
<td>314</td>
</tr>
<tr>
<td>MotorControl</td>
<td>316</td>
</tr>
<tr>
<td>MCCreate.vi</td>
<td>316</td>
</tr>
<tr>
<td>MCEventCloseCurrent.vi</td>
<td>317</td>
</tr>
<tr>
<td>MCEventCloseEMF.vi</td>
<td>318</td>
</tr>
<tr>
<td>MCEventCloseInput.vi</td>
<td>319</td>
</tr>
<tr>
<td>MCEventClosePositionChange.vi</td>
<td>320</td>
</tr>
<tr>
<td>MCEventClosePositionUpdate.vi</td>
<td>322</td>
</tr>
<tr>
<td>MCEventCloseSensorUpdate.vi</td>
<td>323</td>
</tr>
<tr>
<td>MCEventCloseVelocity.vi</td>
<td>324</td>
</tr>
<tr>
<td>MCEventCreateCurrentChange.vi</td>
<td>325</td>
</tr>
<tr>
<td>MCEventCreateCurrentUpdate.vi</td>
<td>326</td>
</tr>
<tr>
<td>MCEventCreateEMF.vi</td>
<td>328</td>
</tr>
<tr>
<td>MCEventCreateInput.vi</td>
<td>329</td>
</tr>
<tr>
<td>MCEventCreatePositionChange.vi</td>
<td>330</td>
</tr>
<tr>
<td>MCEventCreatePositionUpdate.vi</td>
<td>331</td>
</tr>
<tr>
<td>MCEventCreateSensorUpdate.vi</td>
<td>333</td>
</tr>
<tr>
<td>MCEventCreateVelocity.vi</td>
<td>334</td>
</tr>
<tr>
<td>MCEventExeCurrent.vi</td>
<td>335</td>
</tr>
<tr>
<td>MCEventExeEMF.vi</td>
<td>337</td>
</tr>
<tr>
<td>MCEventExinInput.vi</td>
<td>339</td>
</tr>
<tr>
<td>MCEventExePositionChange.vi</td>
<td>340</td>
</tr>
</tbody>
</table>
MCEventExePositionUpdate.vi ................................................................. 342
MCEventExeSensorUpdate.vi ................................................................. 344
MCEventExeVelocity.vi ................................................................. 345
MCGetAcceleration.vi .................................................................. 347
MCGetAccelerationMax.vi ................................................................. 348
MCGetAccelerationMin.vi ................................................................. 350
MCGetBackEMFValue.vi ................................................................. 351
MCGetBraking.vi ........................................................................... 353
MCGetCurrent.vi ........................................................................... 354
MCGetEMFState.vi ........................................................................... 355
MCGetEncoderCount.vi ................................................................... 357
MCGetEncoderPosition.vi ................................................................. 358
MCGetInputCount.vi ........................................................................ 360
MCGetInputState.vi ........................................................................ 361
MCGetMotorCount.vi ....................................................................... 363
MCGetRatiometricState.vi ................................................................. 364
MCGetSensorCount.vi ....................................................................... 365
MCGetSensorValue.vi ....................................................................... 367
MCGetSensorValueRAW.vi ................................................................. 368
MCGetSupplyVoltage.vi ................................................................. 370
MCGetVelocity.vi ........................................................................... 371
MCOpen.vi ..................................................................................... 372
MCSetAcceleration.vi ....................................................................... 374
MCSetBraking.vi ............................................................................... 375
MCSetEMFState.vi ........................................................................... 377
MCSetEncoderPosition.vi ................................................................. 378
MCSetRatiometricState.vi ................................................................. 380
MCSetVelocity.vi ............................................................................... 381

12 PHSensor..................................................................................... 382
PHCreate.vi ................................................................. 382
PHEventClose.vi ................................................................. 383
PHEventCreate.vi ................................................................. 385
PHEventExe.vi ................................................................. 386
PHGetPH.vi .................................................................................. 388
PHGetPHMax.vi ........................................................................... 389
PHGetPHMin.vi ............................................................................... 390
PHGetPHTrigger.vi ........................................................................ 392
PHGetPotential.vi ........................................................................... 393
PHGetPotentialMax.vi .................................................................... 394
PHGetPotentialMin.vi .................................................................... 396
PHOpen.vi ..................................................................................... 397
PHSetTemperature.vi ................................................................. 398
PHSetTrig.vi .................................................................................. 400

13 RFID.......................................................................................... 401
RFIDCreate.vi ................................................................. 401
RFIDEventClose.vi ................................................................. 402
RFIDEventClose2.vi ................................................................. 404
RFIDEventCloseOutput.vi ............................................................. 405
RFIDEventCreateOutput.vi .............................................................. 406
RFIDEventCreateTag.vi ................................................................. 407
RFIDEventCreateTag2.vi ............................................................... 409
RFIDEventCreateTagLost.vi ........................................................... 410
RFIDEventCreateTagLost2.vi .......................................................... 411

© 2014 Phidgets Inc.
RFIDEventExe.vi ................................................................. 412
RFIDEventExe2.vi ............................................................. 414
RFIDEventExeOutput.vi .................................................. 416
RFIDGetAntennaState.vi ................................................ 417
RFIDGetLastTag.vi .......................................................... 419
RFIDGetLastTag2.vi ......................................................... 420
RFIDGetLEDState.vi ......................................................... 422
RFIDGetOutputCount.vi .................................................. 423
RFIDGetOutputState.vi ................................................... 424
RFIDGetTagState.vi ........................................................ 426
RFIDOpen.vi ................................................................ 427
RFIDSetAntennaState.vi ................................................ 429
RFIDSetLEDState.vi ....................................................... 430
RFIDSetOutputState.vi ................................................... 432
RFIDWriteTag.vi ............................................................. 433

14 Servo ............................................................................. 435
ServoCount.vi ................................................................. 435
ServoCreate.vi ............................................................... 436
ServoEventClose.vi ....................................................... 437
ServoEventCreate.vi ...................................................... 438
ServoEventExe.vi ........................................................... 440
ServoGetEngaged.vi ....................................................... 441
ServoGetPos.vi ............................................................... 443
ServoGetPosMax.vi ........................................................ 444
ServoGetPosMin.vi ........................................................ 446
ServoGetServoType.vi .................................................... 447
ServoOpen.vi ................................................................. 448
ServoSetEngaged.vi ....................................................... 450
ServoSetPos.vi ............................................................... 451
ServoSetServoParameters.vi ........................................... 453
ServoSetServoType.vi .................................................... 454

15 Spatial ........................................................................... 456
SpatialCreate.vi .............................................................. 456
SpatialEventClose.vi ..................................................... 457
SpatialEventCreate.vi .................................................... 458
SpatialEventExe.vi ........................................................ 460
SpatialGetAcc.vi ............................................................ 462
SpatialGetAccAxisCount.vi .............................................. 463
SpatialGetAccMax.vi ...................................................... 465
SpatialGetAccMin.vi ...................................................... 466
SpatialGetAngRate.vi ...................................................... 468
SpatialGetAngRateMax.vi ................................................. 469
SpatialGetAngRateMin.vi ................................................ 471
SpatialGetCompassAxisCount.vi ..................................... 472
SpatialGetDataRate.vi .................................................... 473
SpatialGetDataRateMax.vi .............................................. 475
SpatialGetDataRateMin.vi .............................................. 476
SpatialGetGyroAxisCount.vi .......................................... 477
SpatialGetMagField.vi .................................................... 479
SpatialGetMagFieldMax.vi .............................................. 480
SpatialGetMagFieldMin.vi .............................................. 482
SpatialOpen.vi .............................................................. 483
SpatialResetCompassCorrectionParameters.vi ................. 485
### Stepper

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>StepperCreate.vi</td>
<td>490</td>
</tr>
<tr>
<td>StepperEventCloseCurrentVelocity.vi</td>
<td>491</td>
</tr>
<tr>
<td>StepperEventCloseInput.vi</td>
<td>493</td>
</tr>
<tr>
<td>StepperEventClosePosition.vi</td>
<td>494</td>
</tr>
<tr>
<td>StepperEventClosePosition71.vi</td>
<td>495</td>
</tr>
<tr>
<td>StepperEventCreateCurrent.vi</td>
<td>496</td>
</tr>
<tr>
<td>StepperEventCreateInput.vi</td>
<td>498</td>
</tr>
<tr>
<td>StepperEventCreatePosition.vi</td>
<td>499</td>
</tr>
<tr>
<td>StepperEventCreatePosition71.vi</td>
<td>500</td>
</tr>
<tr>
<td>StepperEventCreateVelocity.vi</td>
<td>501</td>
</tr>
<tr>
<td>StepperEventExeCurrentVelocity.vi</td>
<td>503</td>
</tr>
<tr>
<td>StepperEventExeInput.vi</td>
<td>504</td>
</tr>
<tr>
<td>StepperEventExePosition.vi</td>
<td>506</td>
</tr>
<tr>
<td>StepperEventExePosition71.vi</td>
<td>508</td>
</tr>
<tr>
<td>StepperGetAcceleration.vi</td>
<td>509</td>
</tr>
<tr>
<td>StepperGetAccelerationMax.vi</td>
<td>511</td>
</tr>
<tr>
<td>StepperGetAccelerationMin.vi</td>
<td>512</td>
</tr>
<tr>
<td>StepperGetCurrent.vi</td>
<td>514</td>
</tr>
<tr>
<td>StepperGetCurrentLimit.vi</td>
<td>515</td>
</tr>
<tr>
<td>StepperGetCurrentMax.vi</td>
<td>516</td>
</tr>
<tr>
<td>StepperGetCurrentMin.vi</td>
<td>518</td>
</tr>
<tr>
<td>StepperGetCurrentPosition.vi</td>
<td>519</td>
</tr>
<tr>
<td>StepperGetCurrentPosition71.vi</td>
<td>521</td>
</tr>
<tr>
<td>StepperGetEngaged.vi</td>
<td>522</td>
</tr>
<tr>
<td>StepperGetPositionMax.vi</td>
<td>524</td>
</tr>
<tr>
<td>StepperGetPositionMax71.vi</td>
<td>525</td>
</tr>
<tr>
<td>StepperGetPositionMin.vi</td>
<td>527</td>
</tr>
<tr>
<td>StepperGetPositionMin71.vi</td>
<td>528</td>
</tr>
<tr>
<td>StepperGetTargetPosition.vi</td>
<td>530</td>
</tr>
<tr>
<td>StepperGetTargetPosition71.vi</td>
<td>531</td>
</tr>
<tr>
<td>StepperGetVelocity.vi</td>
<td>532</td>
</tr>
<tr>
<td>StepperGetVelocityLimit.vi</td>
<td>534</td>
</tr>
<tr>
<td>StepperGetVelocityMax.vi</td>
<td>535</td>
</tr>
<tr>
<td>StepperGetVelocityMin.vi</td>
<td>537</td>
</tr>
<tr>
<td>StepperInputCount.vi</td>
<td>538</td>
</tr>
<tr>
<td>StepperInputState.vi</td>
<td>540</td>
</tr>
<tr>
<td>StepperMotorCount.vi</td>
<td>541</td>
</tr>
<tr>
<td>StepperOpen.vi</td>
<td>542</td>
</tr>
<tr>
<td>StepperSetAcceleration.vi</td>
<td>544</td>
</tr>
<tr>
<td>StepperSetCurrentLimit.vi</td>
<td>545</td>
</tr>
<tr>
<td>StepperSetCurrentPosition.vi</td>
<td>547</td>
</tr>
<tr>
<td>StepperSetCurrentPosition71.vi</td>
<td>548</td>
</tr>
<tr>
<td>StepperSetEngaged.vi</td>
<td>550</td>
</tr>
<tr>
<td>StepperSetTargetPosition.vi</td>
<td>551</td>
</tr>
<tr>
<td>StepperSetTargetPosition71.vi</td>
<td>553</td>
</tr>
<tr>
<td>StepperSetVelocityLimit.vi</td>
<td>554</td>
</tr>
<tr>
<td>StepperStoppedState.vi</td>
<td>555</td>
</tr>
</tbody>
</table>

### TemperatureSensor

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TempCreate.vi</td>
<td>557</td>
</tr>
</tbody>
</table>

© 2014 Phidgets Inc.
Part VI Phidgets Constants

1 CodeInfo .......................................................... 616

2 Frequency Filter Mode ........................................ 616

3 IREncoding .......................................................... 616

4 IRLength ............................................................... 617

5 ServoType ........................................................... 617

6 ThermocoupleType ................................................ 618

© 2014 Phidgets Inc.
Index 0
1 Welcome to Phidgets

LabVIEW Help

July 2014, Version 2.1.8

This help system includes information about LabVIEW programming for each Phidget device. It contains programming concepts, step-by-step instructions, and reference information about VIs, functions and palettes.

To navigate this help system, use the Contents and Search tabs to the left of this window.

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

This help system may link to Portable Document Format (PDF) versions of documents. You must have Adobe Reader installed to view or search the PDF versions of these manuals.

Note: (Mac OS X) Phidgets recommends that you use Safari 1.3.2 or later or Firefox 1.0.2 or later to view the Help. (Linux) Phidgets recommends that you use Mozilla 1.2 or later or Firefox 1.0.2 or later to view the Help.
2 Introduction

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

This manual documents the Phidgets software programming model in National Instruments Labview language. The Programming Concept should be the first section to be read for someone beginning to use Phidgets. After the concepts described are understood, users can read Phidgets Common and Specific Modules for function reference and device documentation in general. Note that these sections are light on function documentation - generally only containing specific reference information and basic function information.

For a more detailed introduction, please refer to Understanding Phidgets and Platform Support.

2.1 Understanding Phidgets

Hardware Model

All Phidgets are connected to the computer using USB. Most computers support up to 127 USB devices (or more), so it is easy to connect as many Phidgets as are required for almost any project. Phidgets can be connected either directly to a computer or through Hubs, but there are some limitations.

The maximum cable length for USB is 15 feet. This is a maximum distance between device and computer, even if there are one or more Hubs in between. There are cable extenders available on the market, but these can be unreliable and are not endorsed by Phidgets Inc. Users should never try to run USB over anything other then a certified USB cable, and should never try to run it longer than the spec.

Phidgets run as USB 1.1 low speed or full speed devices, and are supported by both USB 1.1 and USB 2.0 hosts.

Software Model

The Phidgets Labview library is written under the C library - phidget21, which implements the low-level protocols necessary to communicate with the Phidgets, and exports a unified interface to the software programmer. This also makes the Phidgets Labview library cross-platform.

The Phidget Labview library contains only glue logic for interfacing with the C library, thus making maintenance much easier. It should be noted the library employs threading and events extensively. (See Programming Concept for more information.)
2.2 Platform Support

Operating System Support

Windows

Microsoft Windows 2000 and later are supported, including 64-bit editions. The Windows libraries are installed using an MSI installer that can be found on the Phidgets web site. This installs the C library, the .NET library, the COM library, the Java library, the Phidget Web Service and the Phidget Control Panel.

The Phidget Control Panel is represented by a “Ph” icon that runs in the system tray (usually on the right end of the Windows task bar). This program can be used to list and control any Phidgets attached to the system, and to control the Web Service.

Mac OS X

Mac OS X 10.3.9 and newer on Intel and PPC are supported. The Mac libraries are distributed in a .dmg and are installed using a standard Mac package installer. This installs the C library, the Kernel driver, the Java library, the Phidget Web Service and the Phidget Preference Pane.

The Phidget Preference Pane is a preference pane which resides in System Preferences. This program can be used to list and control any Phidgets attached to the system, and to control the Web Service.

Linux

Linux version 2.4 is supported, including 64-bit editions, but 2.6.7 or newer is recommended. The Linux libraries are distributed as source. The source for the C library, with optional JNI (Java support) extensions and the source for the Phidget Web Service are available as a .tar.gz. The included Makefile makes it easy to build and install the libraries on most Linux distribution.

Other

Other Operating System support is not currently available.

Labview Version Support

The Phidgets Labview library supports 32-bit Labview version 7.1.1 or higher. It also supports 64-bit Labview version 2009 or higher.

Note: If you use Labview 64-bit, you need to install the Phidgets Labview 64-bit.
library.
3 Programming Concept

This manual is designed such that both novice and expert users can quickly reference the various Phidget LabVIEW functions.

The manual is subdivided into 6 sections: Getting Started, Event Handler, Multiple Devices, Phidgets Common, Specific Modules and Phidgets Constants.

Each section is defined as follow:

Getting Started: tells users how to communicate with phidgets and perform some basic functions. Use the Getting Started manual as a tutorial to familiarize yourself with Phidget LabVIEW functions and basic features you use to build data acquisition and instrument control applications.

Event Handler: needs only be used in applications that need to receive events. Use the Event Handler manual as a tutorial to learn how to construct the event and use the handler.

Multiple Devices: needs only be used in applications that involve multiple phidgets. Use the Multiple Devices manual as a tutorial to configure and control many phidgets in one VI.

Phidgets Common: contains the common functions for all phidgets.

Specific Modules: contains all the functions for specific phidgets.

Phidgets Constants: explains all Phidgets pre-defined constants.

Note: It is important for user to upgrade the Phidgets21 library to the most recent version. Click here to check and download the latest version library.

3.1 Getting Started

Phidgets are an easy to use set of building blocks for low cost sensing and control from your PC. Using the Universal Serial Bus (USB) as the basis for all Phidgets, the complexity is managed behind this easy to use and robust Application Program Interface (API) library.

As such, the Phidgets Labview VI features a very simple and easy-to-use set of VIs. At the simplest level, all you have to do to control a Phidget is explained in this section.

For illustration purposes, Phidget Accelerometer will be used.
Phidgets Accelerometer Example

Note: For a more specific illustration on different phidgets, users can refer to different examples accordingly.

Open the diagram of the "Single control example.vi" under the "Accelerometer" folder.
Phidgets can be programmed into 3 steps:

*<Step 1>* Initialize the Phidget. This includes opening a Phidget hardware, creating a Phidget handler or setting up parameters of a Phidget.

Users can also acquire other information in this step. For more details, please refer to the example called "Remote Example.vi" under "TemperatureSensor" folder.

If users want to open Phidgets over the network, a PhidgetOpenRemote and AcceCreate functions will be called instead of AcceOpen.
<Step 2> Perform any execution related with the Phidget. This includes data acquisition, device control, event execution etc.

<Step 3> Close the Phidget. This may include closing the device, releasing all the resources or freeing a Phidget handle and an error handler.
3.2 Event Handler

This demonstrates how to call a Phidgets event and how to use them.

For illustration purposes, Phidget Spatial will be used.

Note: For a more specific illustration on different phidgets, users can refer to different examples accordingly.
Open the diagram of the "Spatial event example.vi" under the "Spatial" folder.

Phidgets Event can be programmed into 3 steps:

<Step1> Create an event handler.

<Step2> Perform any event execution related with the Phidget.
<Step 3> Close the related event.

Note: When execute an event, please make sure "create", "execute" and "close" the same event. For example, user wants to run an Event called A. He has to place "CreateEventA.vi", "ExeEventA.vi" and "CloseEventA.vi" on the block diagram. He cannot place "CloseEventB.vi" instead of "CloseEventA.vi". However, for some Phidgets event, they share same VIs. (For more details, please refer to specific Phidgets.)

Phidget Error Event:

Another useful example will be "Temp On Error.vi" under "TemperatureSensor" folder.
In this example, the error event is actually located in "Common Functions" folder. It will detect asynchronous errors from Phidgets.

>Note: Phidgets can also support multiple events. For an example, please refer to "Event handler example.vi" under the "InterfaceKit" folder.

### 3.3 Multiple Devices

Phidgets Labview can control multiple phidgets. As long as the handlers are different, different phidgets can run in parallel.

For a more detailed illustration, please refer to the example of "Multiple control example (Parallel).vi" and "Multiple control example (Advanced).vi" under the "Accelerometer" folder.
Note: Another example will be "TemperatureDisplay.vi" under "TextLCD" folder. This example shows how to combined use different phidgets.
4 Phidgets Common

This section describes the VI functions used by all Phidgets. The SubVI folder contains advanced LabVIEW VIs

4.1 ErrorHandler.vi

ErrorHandler.vi

Create a Phidget Error handler

error in  

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error?

TRUE if error occurs

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
4.2 PhidgetClose.vi

PhidgetClose.vi

Close a Phidget device

Device In  
Device # Identification.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out
	error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

4.3 PhidgetDelete.vi

PhidgetDelete.vi

Delete a Phidget device handle

Device In

error in

error out

Device In

Device # Identification.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

4.4 PhidgetEventCloseOnError.vi

PhidgetEventCloseOnError.vi

Close a Phidget error event handle

Device In

Device # Identification.

error in (no error)

derror in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

4.5 PhidgetEventCreateOnError.vi

**PhidgetEventCreateOnError.vi**

Create a Phidget error event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code is the error or warning code.

source describes the origin of the error or warning.

Device Out

Same as Device In

code is the error or warning code.

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

4.6 PhidgetEventExeOnError.vi

PhidgetEventExeOnError.vi

This is called when an asynchronous error occurs.
Device In

Device # Identification.

error in (no error)

type in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

ErrorCode

The error code to get the description of.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

ErrorString

Contain the error description string.

4.7 PhidgetGetDeviceClass.vi

PhidgetGetDeviceClass.vi

Get the class of a Phidget

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DeviceClass

Returns the device class constant

4.8 PhidgetGetDeviceID.vi

PhidgetGetDeviceID.vi

Get the device ID of a Phidget
Device In
Device # Identification.

*error in (no error)*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

*status*

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

*code*

*code* is the error or warning code.

*source*

*source* describes the origin of the error or warning.

Device Out
Same as Device In

*error out*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

*status*

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

*code*
code is the error or warning code.

source

source describes the origin of the error or warning.

DeviceID

Returns the device ID constant

### 4.9 PhidgetGetDeviceLabel.vi

**PhidgetGetDeviceLabel.vi**

Get the label of a Phidget

- **error in**
- **error out** passes error or warning information out of a VI to be used by other VIs.
- **status**
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
  - code is the error or warning code.
- **source**
  - source string describes the origin of the error or warning.

**Device In**

Device # Identification.

**error out**

- error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

DeviceLabel

Returns the device label

4.10 PhidgetGetDeviceName.vi

PhidgetGetDeviceName.vi

Get the specific name of a Phidget

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error
or warning.

Device In
Device # Identification.

error out
error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source string describes the origin of the error or warning.

Device Out
Same as Device In

DeviceName
Returns the device name

4.11 PhidgetGetDeviceType.vi

PhidgetGetDeviceType.vi
Get the type (class) of a Phidget

error in
error in passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

DeviceType

Returns the device type

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.12 PhidgetGetDeviceVersion.vi

PhidgetGetDeviceVersion.vi

Get the firmware version of a Phidget
**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device In**

Device # Identification.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In
Version

Returns the device version

4.13 PhidgetGetServerAddress.vi

PhidgetGetServerAddress.vi

Get the address and port of a remotely opened Phidget. This will fail if the Phidget was opened locally.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

**code** is the error or warning code.

source

**source** string describes the origin of the error or warning.

Device Out

Same as Device In

**ServerAddress**

Returns the address

**port**

Returns the port number

### 4.14 PhidgetGetServerStatus.vi

**PhidgetGetServerStatus.vi**

Get the connected to server status of a remotely opened Phidget. This will fail if the Phidget was opened locally

Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

**code** is the error or warning code.

source

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**ServerStatus**

Returns the server status. Possible values are 0 for unattached, 1 for attached and others for undefined

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

### 4.15 PhidgetGetServiceID.vi

**PhidgetGetServiceID.vi**
Get the server ID of a remotely opened Phidget. This will fail if the Phidget was opened locally.

**Device In**

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device In**

Device # Identification.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**
4.16 PhidgetLibraryVersion.vi

PhidgetLibraryVersion.vi

Get the library version. This contains a version number and a build date.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source string describes the origin of the error or warning.

LibraryVersion

Returns the library version

4.17 PhidgetOpen.vi

PhidgetOpen.vi

Open a Phidget locally

Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In
Device # Identification. This function will create a new device identification if it's 0

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

estaterror out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.18 PhidgetOpenRemote.vi

PhidgetOpenRemote.vi

Open a Phidget remotely by ServerID. Note that this requires Bonjour (mDNS) to be running on both the host and the server.
milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

Serial Number

Serial Number. Specify -1 to open any.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid

serverID

Server ID. Specify NULL to open any

password

Password. Can be NULL if the server is running without password

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

### 4.19 PhidgetOpenRemoteIP.vi

**PhidgetOpenRemoteIP.vi**

Open a Phidget remotely by address and port

- **port address**
- **Device In**
- **Serial Number**
- **milliseconds**
- **error in**
- **Device Out**
- **Serial Number Return**
- **Attached?**
- **error out**

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**Serial Number**

Serial Number. Specify -1 to open any.

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid.

address

Address. This can be a hostname or IP address

password

Password. Can be NULL if the server is running without password

port

Port number. Default is 5001

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

code

code passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
4.20 PhidgetSetDeviceLabel.vi

PhidgetSetDeviceLabel.vi

Set the label of a Phidget. Note that this is not supported on very old Phidgets, and not yet supported in Windows.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

DeviceLabel
The label to be set

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.21 Subvi

4.21.1 _AttachCHK.vi

.AttachCHK.vi

Get the attach state of a Phidget

(error in)

(error out)

(error in) passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

4.21.2 _ChkError.vi

_ChkError.vi

Check the error of a Phidget

error in (no error) OK

error out

Return Code

error in (no error)
**Return Code**

The error code to get the description of.

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.
4.21.3 _Close.vi

_Close.vi

Close a Phidget device

Device In  Device Out

Device In

Device # Identification.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
4.21.4 _Delete.vi

_Device.vi

Delete a Phidget handler

Device In

Device # Identification.

error in

error in passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
4.21.5 _Open.vi

_Open.vi

Open a Phidget device

_device In

_device Out

Serial Number

error in

error out

passes error or warning information out of a VI to be used by other VIs.

status

is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

Serial Number

Serial Number. Specify -1 to open any.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.21.6 _OpenRemote.vi

_OpenRemote.vi

Open a Phidget remotely by ServerID. Note that this requires Bonjour (mDNS) to be running on both the host and the server

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.
Device In

Device # Identification.

Serial Number

Serial Number. Specify -1 to open any.

serverID

Server ID. Specify NULL to open any

password

Password. Can be NULL if the server is running without password

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.21.7  _OpenRemoteIP.vi

_OpenRemoteIP.vi

Open a Phidget remotely by address and port
error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

Serial Number

Serial Number. Specify -1 to open any.

address

Address. This can be a hostname or IP address

password

Password. Can be NULL if the server is running without password

port

Port number. Default is 5001

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.21.8 _SerialReturn.vi

_SerialReturn.vi

Return the serial number of a Phidget

error in

eerror out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In
Device # Identification.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

**Serial Number Return**

Serial Number of the opened phidget

### 4.21.9 _WaitAttach.vi

_WidgetAttach.vi_

Wait until a Phidget attached

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred. 
code

code is the error or warning code. 
source

source string describes the origin of the error or warning.

Device In

Device # Identification.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)
error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred. 
code

code is the error or warning code. 
source

source string describes the origin of the error or warning.

Device Out

Same as Device In

4.21.10 EventCloseIntDouble.vi

EventCloseIntDouble.vi

Close the event handler which contains an integer and a double event variables. This should not be directly accessed by users
Device In

Device # Identification.

error in (no error)

time severity

time can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

time severity

time passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

describes the origin of the error or warning.

4.21.11 EventCloseIntInt.vi

EventCloseIntInt.vi

Close the event handler which contains an integer and an integer event variables. This should not be directly accessed by users

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out
Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

### 4.21.12 EventExeIntDouble.vi

**EventExeIntDouble.vi**

Call the event handler which contains an integer and a double event variables. This should not be directly accessed by users

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

acceleration

The double

index

The integer

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification
Event?

Returns TRUE if the event has executed, or FALSE otherwise.

### 4.21.13 EventExeIntInt.vi

**EventExeIntInt.vi**

Call the event handler which contains an integer and an integer event variables. This should not be directly accessed by users.

- **Device In**
  
  Device # Identification.

- **error in (no error)**
  
  `error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  
  `status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  
  `code` is the error or warning code.

- **source**
  
  `source` describes the origin of the error or warning.

- **Event Registration Refnum In**
  
  Event # Identification
The 1st integer

Device Out

Same as Device In

error out

code is the error or warning code.

source string describes the origin of the error or warning.

Event Registration Refnum

Event # Identification

The 2nd integer

Event?

Returns TRUE if the event has executed, or FALSE otherwise.
5 Specific Modules

This section describes each of the VI function used by different Phidgets. All the VI functions are located in its dll folder correspondingly.

Note: Refer to the Product manual for your Phidget and the C Programming Manual for more detailed, language unspecific API documentation.

Please select a specified module accordingly.

5.1 Accelerometer

5.1.1 AcceAxisCount.vi

AcceAxisCount.vi

Get the number of acceleration axes supported by this accelerometer.

Device In

Device In

Device # Identification.

error in (no error)

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

count

The number of axes

5.1.2 AcceCreate.vi

AcceCreate.vi

Create a Phidget Accelerometer handle.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

The created Device # ID

5.1.3 AcceEventClose.vi

AcceEventClose.vi

Close the acceleration change event handle.

Device In

Device # Identification.

error in (no error)

Device Out

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.
5.1.4  AcceEventCreate.vi

AcceEventCreate.vi

Set up an acceleration change event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.1.5 AcceEventExe.vi

AcceEventExe.vi

This is called when the acceleration changes by more than the change trigger.

Device In

Device # Identification.

eroerror in (no error)

eroerror in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

Event Registration Refnum In

Event # Identification

acceleration

The acceleration

index

The acceleration index.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.1.6 AcceGetData.vi

AcceGetData.vi
Get the current acceleration data of an axis.

Device In

Device # Identification.

Channel

Channel of the device to open

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

acceleration

The acceleration

error out

error out can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
5.1.7 AcceGetMax.vi

AcceGetMax.vi

Get the maximum acceleration supported by an axis.

Device In

Device # Identification.

Channel

The acceleration index. (x, y, z)

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
code is the error or warning code.

source

describes the origin of the error or warning.

Device Out

Same as Device In

acce_max

The maximum acceleration.

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

c is the error or warning code.

source

describes the origin of the error or warning.

5.1.8  AcceGetMin.vi

AcceGetMin.vi

Get the minimum acceleration supported by an axis.
Device # Identification.

**Channel**

The acceleration index. \((x, y, z)\)

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**acce_min**

The minimum acceleration.

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.1.9 AcceGetTrigger.vi

AcceGetTrigger.vi

Get the change trigger for an axis.

Device In

Device # Identification.

Channel

The acceleration index. (x, y, z)

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out
Same as Device In

**Trigger Out**

The change trigger.

**error out**

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

### 5.1.10 AcceOpen.vi

**AcceOpen.vi**

Open a Phidget Accelerometer.

**Serial Number**

Serial number. Specify -1 to open any.

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**error in**
error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out
5.1.11 AcceSetTrigger.vi

AcceSetTrigger.vi

Set the change trigger for an axis.

Device In

Device # Identification.

Channel

Channel of the device to open

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Trigger In

The change trigger.

Device Out

Same as Device In
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

## 5.2 AdvancedServo

### 5.2.1 AdvServoCount.vi

**AdvServoCount.vi**

Gets the number of motors supported by this controller.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error in

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

count

The motor count.

5.2.2 AdvServoCreate.vi

AdvServoCreate.vi

Create a Phidget Advanced Servo handle.
5.2.3 AdvServoEventClose.vi

AdvServoEventClose.vi
Close the Phidget Advanced Servo event handle.

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

*error out* passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.2.4 AdvServoEventCreateCrtChange.vi

AdvServoEventCreateCrtChange.vi

Set up a current change event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

eror out
error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.2.5 AdvServoEventCreatePosChange.vi

AdvServoEventCreatePosChange.vi

Set up a postion change event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
Specific Modules

**Device Out**

Same as Device In

**error out**

*error out* passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* describes the origin of the error or warning.

---

5.2.6 **AdvServoEventCreateVelChange.vi**

AdvServoEventCreateVelChange.vi

Set up a velocity change event handle

Device In

error in (no error)

Event Registration Refnum Out

Device Out

Device # Identification

Event # Identification
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification
5.2.7 AdvServoEventExe.vi

AdvServoEventExe.vi

This is called when the Phidget Advanced Servo event changes.

### Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### Event Registration Refnum In

Event # Identification

**Value**

The return value of related event. (E.g.: For a position change event, this value is position.)

**index**
The motor index.

**Device Out**

Same as Device In

**error out**

`error out` passes error or warning information out of a VI to be used by other VIs.

**status**

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

`code` is the error or warning code.

**source**

`source` string describes the origin of the error or warning.

**Event Registration Refnum Out**

Same as the Event Registration Refnum In.

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

### 5.2.8 AdvServoGetAcce.vi

**AdvServoGetAcce.vi**

Get the last set acceleration of a motor

**Device In**

Device # Identification.

**error in (no error)**
**Specific Modules**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.2.9 AdvServoGetAcceMax.vi

Get the maximum acceleration supported by a motor.

**Device In**
- Device # Identification.
- **error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
- **code** is the error or warning code.

**source**
- **source** describes the origin of the error or warning.

**index**
- The motor index

**Device Out**
- Same as Device In

**error out**

The acceleration
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

accemax_out

The maximum acceleration

5.2.10 AdvServoGetAcceMin.vi

AdvServoGetAcceMin.vi

Get the minimum acceleration supported by a motor.

Device In

Device # Identification.

error in (no error)

Device Out

accmin_out

error out

Device In

Device # Identification.

error in (no error)
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**accemin_out**

The minimum acceleration
5.2.11 AdvServoGetCurrent.vi

AdvServoGetCurrent.vi

Get the current current draw for a motor.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**current_out**

The current

5.2.12 AdvServoGetEngaged.vi

**AdvServoGetEngaged.vi**

Get the engaged state of a motor. This is whether the motor is powered or not

- **Device In**
  - Device # Identification.
  - **error in (no error)**
  - **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EngagedState_out

The engaged state. Possible values are 0 for False, 1 for True and others for undefined

Engaged?

The engaged state. Possible values are True for Engaged and False for Not Engaged
5.2.13 AdvServoGetPos.vi

AdvServoGetPos.vi

Get the current position of a motor.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**pos_out**

The position.

### 5.2.14 AdvServoGetPosMax.vi

**AdvServoGetPosMax.vi**

Get the maximum position that a motor can go to.

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

posmax_out

The maximum position

5.2.15 AdvServoGetPosMin.vi

AdvServoGetPosMin.vi

The minimum position
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

posmin_out

The minimum position

5.2.16 AdvServoGetRampingState.vi

AdvServoGetRampingState.vi

Get the speed ramping state for a motor. This is whether or not velocity and acceleration are used

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

RampingState_out

The speed ramping state. Possible values are 0 for False, 1 for True and others for undefined.

RampingState?

The speed ramping state (Boolean type).

5.2.17 AdvServoGetServoType.vi

AdvServoGetServoType.vi

Get the servo type of a motor

© 2014 Phidgets Inc.
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

GetServoType

Returns the servo type. This is an enum. Please refer to [Phidgets Constants -> ServoType](#).

## 5.2.18 AdvServoGetVel.vi

### AdvServoGetVel.vi

Get the current velocity of a motor.

![Diagram](#)

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**
source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

vel_out

The current velocity

5.2.19 AdvServoGetVelLmt.vi

AdvServoGetVelLmt.vi

Get the last set velocity limit of a motor.

Device In

velLmt

Device Out

vellim_out

error out

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source describes the origin of the error or warning.

vellim_out

The velocity limit

5.2.20 AdvServoGetVelMax.vi

AdvServoGetVelMax.vi

Get the maximum velocity that can be set for a motor

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index
Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

velmax_out

The maximum velocity

5.2.21 AdvServoGetVelMin.vi

AdvServoGetVelMin.vi

Get the minimum velocity that can be set for a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**velmin_out**

The minimum velocity
5.2.22  AdvServoOpen.vi

AdvServoOpen.vi

Serial Number
Serial Number. Specify -1 to open any.

milliseconds
Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in

error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source string describes the origin of the error or warning.

Device In
Device # identification. This function will create a new device identification if it's 0 or invalid.

Serial Number Return
Serial Number of the opened phidget

Attached?
Returns TRUE is the device successfully attached, or FALSE otherwise.
error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.2.23 AdvServoSetAcce.vi

AdvServoSetAcce.vi

Set the acceleration for a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

accel

The acceleration

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.2.24 AdvServoSetEngaged.vi

AdvServoSetEngaged.vi
Set the engaged state of a motor. This is whether the motor is powered or not.

**Device In**
- Device # Identification.
- **error in (no error)**
- **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**
- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
- **code** is the error or warning code.
- **source**
- **source** describes the origin of the error or warning.

**Device Out**
- Same as Device In

**error out**
- **error out** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
5.2.25 AdvServoSetPos.vi

AdvServoSetPos.vi

Set the position of a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
source describes the origin of the error or warning.

pos_in

The position

index

The motor index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.2.26 AdvServoSetPosMax.vi

AdvServoSetPosMax.vi

Set the maximum position that a motor can go to.

Device In

Device Out

posmax_in (no error)

error in

Far Max by

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**posmax_in**

The maximum position

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.2.27  AdvServoSetPosMin.vi

**AdvServoSetPosMin.vi**

Set the minimum position that a motor can go to.

**Device In**

Device # Identification.

**error in (no error)**

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

`code` is the error or warning code.

**source**

`source` describes the origin of the error or warning.

**posmin_in**

The minimum position...
index

The motor index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.2.28 AdvServoSetRampingState.vi

AdvServoSetRampingState.vi

Set the speed ramping state for a motor. This is whether or not velocity and acceleration are used

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**SetRampingState**

The speed ramping state. ( 0 = False 1 = True )

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.2.29  AdvServoSetServoParameters.vi

AdvServoSetServoParameters.vi

Set the servo parameters of a motor.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

min_us

The minimum supported PCM in microseconds
max_us
The maximum supported PCM in microseconds

degrees
The degrees of rotation defined by the given PCM range

velocity_max
The maximum velocity in degrees/second

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

5.2.30 AdvServoSetServoType.vi

AdvServoSetServoType.vi
Set the servo type of a motor
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

setServoType

The servo type. This is an enum. Please refer to Phidgets Constants -> ServoType

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

### 5.2.31 AdvServoSetVelLmt.vi

**AdvServoSetVelLmt.vi**

Set the velocity limit for a motor

![Diagram](image)

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**vel_in**
The velocity limit

index

The motor index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.2.32 AdvServoStoppedState.vi

AdvServoStoppedState.vi

Get the stopped state of a motor. This is true when the motor is not moving and there are no outstanding commands

Device In

Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.3 Analog

5.3.1 AnalogCreate.vi

AnalogCreate.vi

Create a Phidget Analog device

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

The created device # ID

5.3.2 AnalogGetEnabled.vi

AnalogGetEnabled.vi

Get the enabled state of the device

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source describes the origin of the error or warning.

index

The analog output index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EnabledState

The enabled state. Possible values are 0 for False, 1 for True and others for undefined

Enabled?

The enabled state. Possible values are True for Enabled and False for Not Enabled

5.3.3 AnalogGetVoltage.vi

AnalogGetVoltage.vi
Get the measured voltage

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The analog output index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

voltage

The voltage

5.3.4 AnalogGetVoltageMax.vi

AnalogGetVoltageMax.vi

Gets the maximum voltage that can be output

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

index

The analog output index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

VoltageMax

The maximum voltage

5.3.5 AnalogGetVoltageMin.vi

AnalogGetVoltageMin.vi

Gets the minimum voltage that can be output
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The analog output index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
5.3.6 AnalogOpen.vi

**AnalogOpen.vi**

Opens a Phidget Analog device

- **Serial Number**
  - Serial Number. Specify -1 to open any.

- **milliseconds**
  - Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

- **error in (no error)**
  - **error in (no error)** passes error or warning information out of a VI to be used by other VIs.

- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  - **code** is the error or warning code.

- **source**
  - **source** string describes the origin of the error or warning.

- **Device In**
Device # identification. This function will create a new device identification if it's 0 or invalid.

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

5.3.7 *AnalogOutputCount.vi*

**AnalogOutputCount.vi**

Gets the number of analog outputs on the device

**Device In**

Device # Identification.
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count
5.3.8  **AnalogSetEnabled.vi**

The number of analog outputs on the device

**AnalogSetEnabled.vi**

Set the enabled state of the device

- **Device In**
  - Device # Identification.
- **error in (no error)**
  - *error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**
  - *status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
  - *code* is the error or warning code.
- **source**
  - *source* describes the origin of the error or warning.
- **index**
  - The analog output index
- **SetEnabled**
  - The enabled state
- **Device Out**
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.3.9 AnalogSetVoltage.vi

**AnalogSetVoltage.vi**

Sets the voltage to output

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**voltage**

The voltage to output

**index**

The analog output index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.4 Bridge
5.4.1 BridgeCreate.vi

**BridgeCreate.vi**

Create a phidget bridge device.

- **error in**
- **error out** passes error or warning information out of a VI to be used by other VIs.
- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code**
  - **code** is the error or warning code.
  - **source**
  - **source** string describes the origin of the error or warning.

- **error out** passes error or warning information out of a VI to be used by other VIs.
- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
  - **code**
  - **code** is the error or warning code.
  - **source**
  - **source** string describes the origin of the error or warning.
5.4.2 BridgeEventClose.vi

BridgeEventClose.vi

Close a Phidget Bridge event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out
**5.4.3 BridgeEventCreate.vi**

Create a Phidget Bridge event handle

- **Device In**
  - Device # Identification.
- **error in (no error)**
  - **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
  - **code** is the error or warning code.
- **source**
  - **source** string describes the origin of the error or warning.
source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.4.4 BridgeEventExe.vi

BridgeEventExe.vi

Returns the value of the selected input

Device In

Device # Identification.
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Value

The value of the selected input (mV/V)

index

The bridge input index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Same as Event # Registration Refnum In

5.4.5 BridgeGetDataRate.vi

BridgeGetDataRate.vi

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**DataRate**

### 5.4.6 BridgeGetDataRateMax.vi

**BridgeGetDataRateMax.vi**

Gets the maximum supported data rate, in ms.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**DataRateMax**

Maximum data rate

---

5.4.7  **BridgeGetDataRateMin.vi**

**BridgeGetDataRateMin.vi**
Gets the minimum supported data rate, in ms.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.4.8 BridgeGetEnabled.vi

BridgeGetEnabled.vi

Get the enabled state of a bridge input.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
index
The bridge input index

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

EnabledState
The enabled state. Possible values are 0 for False, 1 for True and others for undefined

Enabled?
The enabled state. Possible values are True for Enabled and False for Not Enabled

5.4.9 BridgeGetGain.vi

BridgeGetGain.vi

Gets the gain setting of the bridge input
Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The bridge input index

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source
describes the origin of the error or warning.

Gain
The gain setting

5.4.10 BridgeGetValue.vi

BridgeGetValue.vi

Gets the value of the selected bridge input

Device In

Device Out

Device In

index

Device Out

value

error in (no error)

error out

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
describes the origin of the error or warning.

index

The bridge input index
Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

value

The value on the bridge input

5.4.11 BridgeGetValueMax.vi

BridgeGetValueMax.vi

The maximum measurable bridge value

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The bridge input index

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**ValueMax**

The maximum value
5.4.12 BridgeGetValueMin.vi

BridgeGetValueMin.vi

The minimum measurable bridge value.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The bridge input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

ValueMin

The minimum value

5.4.13 BridgInputCount.vi

BridgInputCount.vi

Gets the number of bridge inputs on the device

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count

The number of bridge inputs on the device

5.4.14 BridgeOpen.vi

BridgeOpen.vi

Device In
Serial Number
milliseconds
error in (no error)

Device Out
Serial Number Return
Attached?
error out

Serial Number

© 2014 Phidgets Inc.
Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification. This function will create a new device identification if it's 0

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.4.15 BridgeSetDataRate.vi

BridgeSetDataRate.vi

Sets the data rate for the bridge input. (ms)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
DataRate
The data rate

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

5.4.16 BridgeSetEnabled.vi

BridgeSetEnabled.vi
Sets the enabled state of the bridge input

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The bridge input index

**SetEnabled**

The enabled state

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.4.17 BridgeSetGain.vi

**BridgeSetGain.vi**

Sets the bridge gain

- **Device In**: Device # Identification.
- **error in (no error)**: error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**: status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**: code is the error or warning code.

- **source**: source describes the origin of the error or warning.

- **gain**: The bridge gain

- **index**: The bridge input index

- **Device Out**
Same as Device In

**error in**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.5 Encoder

#### 5.5.1 EncoderCreate.vi

**EncoderCreate.vi**

Create a Phidget Encoder handle

**error in**

**error in** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

The created device # ID

5.5.2 EncoderEventCloseInput.vi

EncoderEventCloseInput.vi

Close the input change event handle

Device In

Device # Identification.

error in (no error)

Device Out

t

t

error out

t

t

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.5.3 EncoderEventClosePosition.vi

EncoderEventClosePosition.vi
Close the encoder position change event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.5.4 EncoderEventCreateInput.vi

**EncoderEventCreateInput.vi**

Set up an input change event handle

![Diagram](image)

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

---

### 5.5.5 EncoderEventCreatePosition.vi

**EncoderEventCreatePosition.vi**

Set up an encoder position change event handle

```
Device In ———> Encoder ———> Device Out
error in (no error) ———> Event Registration Refnum Out ———> error out
```

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.5.6 EncoderEventExeInput.vi

**EncoderEventExeInput.vi**

Thos occurs on a Phidget Encoder Input event
**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device In**

Device # Identification.

**index**

The encoder index

**Value**

The input state value. Possible values are 0 for False, 1 for True and others for undefined

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.5.7 EncoderEventExePosition.vi

EncoderEventExePosition.vi

This is called when an encoder position changes.

code

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

© 2014 Phidgets Inc.
code is the error or warning code.

source

describes the origin of the error or warning.

Device In

Device # Identification.

Event Registration Refnum In

Event # Identification

error out

passes error or warning information out of a VI to be used by other VIs.

status

is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

string describes the origin of the error or warning.

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Same as the Event Registration Refnum In

index

The encoder index

time

© 2014 Phidgets Inc.
The time in ms since the last position change event

position

The current position of the encoder. (This is a relative not absolute position.)

5.5.8 EncoderGetCount.vi

EncoderGetCount.vi

Get the number of encoders supported by this board

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In
error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other
VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EncoderCount

The encoder input count

5.5.9 EncoderGetEnabledState.vi

EncoderGetEnabledState.vi

Get the enabled state of an encoder. This is whether the encoder is powered or not. Please note that
1057 doesn't support this function

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other
VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EncoderState

The enabled state. Possible values are 0 for False, 1 for True and others for undefined Enabled?
The enabled state. Possible values are True for Enabled and False for Not Enabled

5.5.10 EncoderGetIndexPosition.vi

EncoderGetIndexPosition.vi

Get the position of the last index pulse, as referenced to Phidget Encoder -> EncoderGetPosition. The function will return an error (EPHIDGET_UNKNOWN) if there hasn't been an index event, or if the encoder doesn't support index

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

Device Out

Same as Device In
error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

IndexPosition

The index position

5.5.11 EncoderGetInputCount.vi

EncoderGetInputCount.vi

Get the number of encoders supported by this board

Device In

Device In

error in (no error)

Device # Identification.

test in (no error)

test in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
**5.5.12 EncoderGetInputState.vi**

**EncoderGetInputState.vi**

Get the state of a digital input
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

input_state

The input state. Possible values are 0 for False, 1 for True and others for undefined

InputState

The input state (Boolean type). Possible values are True for Engaged and False for Not Engaged

5.5.13 EncoderGetPosition.vi

EncoderGetPosition.vi

Get the current position of an encoder

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Position

The current position

5.5.14 EncoderOpen.vi

EncoderOpen.vi

Open a PhidgetEncoder
Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

group box

group box

group box

group box

group box

error in (no error)

group box

group box

group box

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

group box

group box

group box

group box

group box

Device In

Device # Identification. This function will create a new device identification if it's 0

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.5.15 EncoderSetEnabled.vi

EncoderSetEnabled.vi

Set the enabled state of an encoder. This is whether the encoder is powered or not. Please note that 1057 doesn't support this function

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

SetEnable

The encoder state.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.5.16 EncoderSetPosition.vi

EncoderSetPosition.vi

Set the position of an encoder
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The encoder index

setPosition

The new position

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.6 Frequency

5.6.1 FreqCreate.vi

FreqCreate.vi

Create a Phidget Frequency input handle

error in

error in passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

The created Device # ID

5.6.2 FreqEventClose.vi

FreqEventClose.vi

Close a Phidget Frequency input event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.6.3 FreqEventCreate.vi

FreqEventCreate.vi

Create a Phidget Frequency input event handle

Device In

Device Out

error in (no error)

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

5.6.4  **FreqEventExe.vi**

**FreqEventExe.vi**
Executes whenever some counts have been detected

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Device In**

Device # Identification.

**Event Registration Refnum In**

**error out**

*error out* passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
**FreqGetCount.vi**

Gets the total number of pulses detected on the specified channel since the Phidget was opened, or since the last reset.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count

The total number of pulses
5.6.6  FreqGetCount71.vi

FreqGetCount71.vi

Gets the total number of pulses detected on the specified channel since the Phidget was opened, or since the last reset.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Count**

The number of pulses

### 5.6.7 FreqGetEnabled.vi

**FreqGetEnabled.vi**

Get the enabled state of the device

![Diagram of FreqGetEnabled.vi](image)

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
Specific Modules

that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EnabledState

The enabled state. Possible values are 0 for False, 1 for True and others for undefined

Enabled?

The enabled state. Possible values are True for Enabled and False for Not Enabled
5.6.8  FreqGetFilter.vi

FreqGetFilter.vi

Get the channel filter mode.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Filter Type

The filter type. This is an enum. Please refer to Phidgets Constants -> Frequency Filter Mode

5.6.9 FreqGetTime.vi

FreqGetTime.vi

Gets the total elapsed time since Phidget was opened, or since the last reset, in microseconds.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.6.10 FreqGetTime71.vi

**FreqGetTime71.vi**

Gets the total elapsed time since Phidget was opened, or since the last reset, in microseconds.
### Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

### Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Time

The elapsed time

5.6.11 FreqGetTimeout.vi

FreqGetTimeout.vi

Gets or set the Timeout value, in microseconds

Device In

index

code

Device Out

Timeout

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Timeout**

The timeout value

5.6.12 FreqGetValue.vi

**FreqGetValue.vi**

Get the frequency measured by the device.
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source describes the origin of the error or warning.

Frequency

The measured frequency

5.6.13 FreqInputCount.vi

FreqInputCount.vi

The number of frequency inputs on the device

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In
error out

(error in) can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count

The number of inputs

5.6.14 FreqOpen.vi

FreqOpen.vi

Opens a Phidget Frequency Input device

Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to
wait forever. (Default is 5000)

(error in (no error))

time error out

error out passes error or warning information
out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

### Device In

Device # Identification. This function will create a new device identification if it's 0

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

### Device Out
5.6.15  FreqReset.vi

FreqReset.vi

Resets the TotalCount and TotalTime counters to 0 for the specified channel.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

### 5.6.16 FreqSetEnabled.vi

**FreqSetEnabled.vi**

Sets the enabled state of the channel.

- **Device In**
  - Device # Identification.

- **error in (no error)**
  - error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

SetEnabled

The enabled state of the channel

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.6.17 FreqSetFilter.vi

FreqSetFilter.vi

Set the channel filter mode. This controls the type of signal that the frequency counter will respond to - either a zero-centered signal, or a logic level signal.
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Filter

The filter type. This is an enum. Please refer to Phidgets Constants - Frequency Filter Mode

index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
**FreqSetTimeout.vi**

Sets the timeout value in microsseconds.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
timeout
The timeout value.

index
The input index

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

5.7 GPS

5.7.1 GPSCreate.vi

GPSCreate.vi
Create a Phidget GPS device

error in
der error in passes error or warning information out of a VI to be used by other VIs.
**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* string describes the origin of the error or warning.

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* string describes the origin of the error or warning.

### 5.7.2 GPSEventCloseFixStatus.vi

**GPSEventCloseFixStatus.vi**

Close a Phidget GPS Fix Status event handler

![Diagram of GPSEventCloseFixStatus.vi](image)
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.
5.7.3  GPSEventClosePosition.vi

GPSEventClosePosition.vi

Close a Phidget GPS Position event handler

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.7.4 GPSEventCreateFixStatus.vi

**GPSEventCreateFixStatus.vi**

Create a Phidget GPS Fix Status event handler

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.7.5 GPSEventCreatePosition.vi

GPSEventCreatePosition.vi

Create a Phidget GPS Position event handler

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.7.6  GPSEventExeFixStatus.vi

GPSEventExeFixStatus.vi

Executes when a position fix is obtained or lost.
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Same as Event Registration Refnum In

Status

The fix status. Possible values are 0 for False, 1 for True and others for undefined

Status (T?F)

The fix status. Possible values are True for Fixed and False for Not Fixed

5.7.7 GPSEventExePosition.vi

GPSEventExePosition.vi

Executes when position changes.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

*error out* passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* string describes the origin of the error or warning.

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Event Registration Refnum Out**
5.7.8  **GPSGetAltitude.vi**

**GPSGetAltitude.vi**

Gets the altitude.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

Altitude
The altitude

5.7.9 GPSGetDate.vi

GPSGetDate.vi
Gets the date of the last received position.

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Date**

The date in UTC.

**tm_mday**
The day of the month
\texttt{tm\_mon}

The month
\texttt{tm\_year}

The year

\section*{5.7.10 GPSGetFixStatus.vi}

\textbf{GPSGetFixStatus.vi}

Gets the GPS fix status.

\begin{itemize}
\item \textbf{Device In}
\item \texttt{Device \# Identification.}
\item \texttt{error in (no error)}
\item \textbf{error in} can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
\item \textbf{status}
\item \texttt{status} is \texttt{TRUE} (\texttt{X}) if an error occurred or \texttt{FALSE} (checkmark) to indicate a warning or that no error occurred.
\item \texttt{code}
\item \texttt{code} is the error or warning code.
\item \texttt{source}
\item \texttt{source} describes the origin of the error or warning.
\end{itemize}

\textbf{Device Out}
Same as Device In

$error out$

$error in$ can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

$status$

$status$ is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

$code$

$code$ is the error or warning code.

source

$sources$ describe the origin of the error or warning.

FixStatus

The fix status. Possible values are 0 for False, 1 for True and others for undefined

FixStatus (T?F)

The fix status. Possible values are True for Fixed and False for Not Fixed

5.7.11 GPSGetHeading.vi

GPSGetHeading.vi

Returns the current true course over ground of the active antenna in degrees (000.0 - 359.9).

Device In

Device # Identification.
**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Heading**
5.7.12 GPSGetLatitude.vi

GPSGetLatitude.vi

Gets the latitude of the antenna.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Latitude**

The latitude

5.7.13 GPSGetLongitude.vi

**GPSGetLongitude.vi**

Gets the longitude of the antenna.

- **Device In**
  - **Device In** Identification.
  - **error in (no error)**
  - **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Longitude

The longitude

5.7.14 GPSGetTime.vi

GPSGetTime.vi

Gets the current time as transmitted by the GPS receiver.
Device In

Device # Identification.

**error in** *(no error)*

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or
warning.

Time
Current time in UTC format.

`tm_ms`
Current ms

`tm_sec`
Current second

`tm_min`
Current minute

`tm_hour`
Current hour

5.7.15 GPSGetVelocity.vi

**GPSGetVelocity.vi**

Get the velocity of the antenna.

- **Device In**
  Device # Identification.

  - `error in (no error)`

- **Device Out**

- **Velocity**

  - `Device Out Velocity`

  - `Velocity`

  - `Device Out velocity`

- **Device In**
  Device # Identification.

  - `error in (no error)`

  - `error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

  - `status`

    - `status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

    - `code`
code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Velocity

The velocity.

5.7.16 GPSOpen.vi

GPSOpen.vi

Open a PhidgetGPS device.
Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in

derror out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # Identification.

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

derror out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.8 InterfaceKit

5.8.1 IFCreate.vi

IFCreate.vi

Create a Phidget InterfaceKit handle

**error in**

**error in** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Device # identification

5.8.2 IFEventClose.vi

IFEventClose.vi

Close an IF Kit event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.8.3 IFEventCreateInput.vi

IFEventCreateInput.vi

Create an IF Kit Input event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

erro r out

erro r out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.8.4 IFE ventCreateOutput.vi

IFEventCreateOutput.vi
Create an IF kit Output event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
5.8.5 IFEventCreateSensor.vi

**IFEventCreateSensor.vi**

Create an IF kit sensor event.

- **Device In**
  - Device # Identification.
  - **error in (no error)**
    - *error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**
  - *status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
  - *code* is the error or warning code.
- **source**
  - *source* describes the origin of the error or warning.

- **Device Out**
  - Same as Device In
error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.8.6 IFEventExe.vi

IFEventExe.vi

Executes when an IF kit event occurs.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device In

Device # Identification.

index

The input index

Value

The value returned by the event

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

Device Out

Same as Device In

Event?
Returns TRUE if the event has executed, or FALSE otherwise.

5.8.7 IFGetDataRate.vi

IFGetDataRate.vi

Get the data rate for an analog input

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DataRate

The data rate

5.8.8 IFGetDataRateMax.vi

IFGetDataRateMax.vi

Get the maximum supported data rate for an analog input

Device In

Device Out

error in (no error)

error out

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**DataRateMax**

The maximum data rate
5.8.9 **IFGetDataRateMin.vi**

**IFGetDataRateMin.vi**

Get the minimum supported data rate for an analog input.

- **Device In**
  - Device # Identification.
  - *error in (no error)*

- **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  - *status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  - *code* is the error or warning code.

- **source**
  - *source* describes the origin of the error or warning.

- **index**
  - The input index

- **Device Out**
  - Same as Device In

- **error out**
  - *error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DataRateMin

The minimum data rate.

5.8.10 IFGetInputCount.vi

IFGetInputCount.vi

Get the number of digital inputs supported by this board

Device In

Device # Identification.

error in (no error)

error in (no error) can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
describes the origin of the error or warning.

Device Out
Same as Device In

error out
can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
describes the origin of the error or warning.

Input Count
the number of digital inputs

5.8.11 IFGetInputState.vi

IFGetInputState.vi

The state of the indexed digital input.
**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.
source
source describes the origin of the error or warning.

Input State (0,1)
The input state. Possible values are 0 for False, 1 for True and others for undefined

Input State
The input state (Boolean type)

5.8.12 IFGetOutputCount.vi

IFGetOutputCount.vi
Get the number of digital outputs supported by this board

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.
Device Out

Same as Device In

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Output count

The digital output count

5.8.13 IFGetOutputState.vi

IFGetOutputState.vi

Get the state of a digital output

Device In

Device # Identification.

code

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

source

source describes the origin of the error or warning.
from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.
### Output State (0,1)

The output state. Possible values are 0 for False, 1 for True and others for undefined.

### Output State

The output state (Boolean type).

## 5.8.14 IFGetRatio.vi

IFGetRatio.vi

Get the ratiometric state for this board.

- **Device In**
  - Device # Identification.
- **error in (no error)**
  - error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**
  - code is the error or warning code.
- **source**
  - source describes the origin of the error or warning.

- **Device Out**
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Ratiometric**

The ratio metric state. Possible values are 0 for False, 1 for True and others for undefined

**Ratiometric State**

The ratio metric state (Boolean type).

### 5.8.15 IFGetSensorCount.vi

**IFGetSensorCount.vi**

Get the number of sensor (analog) inputs supported by this board

**Device In**

Device # Identification.

**error in (no error)**
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Sensor Count

The sensor input count
5.8.16  IFGetSensorValue.vi

**IFGetSensorValue.vi**

Get a sensor value (0-1000).

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

_**status**_ is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

_**code**_ is the error or warning code.

**source**

_**source**_ describes the origin of the error or warning.

**Sensor Value**

The sensor value

### 5.8.17 IFGetSensorValueRaw.vi

**IFGetSensorValueRaw.vi**

Get a sensor raw value (12-bit).

**Device In**

Device # Identification.

**error in (no error)**

_**error in**_ can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

_**status**_ is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Sensor Value (Raw)

The sensor value

5.8.18 IFGetTrig.vi

Get a sensor change trigger
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
Specific Modules

that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Trig

The change trigger

5.8.19 IFOpen.vi

IFOpen.vi

Open a PhidgetInterfaceKit

Device In

Serial Number

milliseconds

error in (no error)

Device Out

Serial Number Return

Attached?

error out

Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

code

code is the error or warning code.

source

source describes the origin of the error or warning.
source string describes the origin of the error or warning.

Device In

Device # Identification. This function will create a new device identification if it's 0

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.8.20 IFSetDataRate.vi

IFSetDataRate.vi

Set the data rate for an analog input
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**DataRate**

The data rate.

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

### 5.8.21 IFSetOutputState.vi

**IFSetOutputState.vi**

The Set the state of a digital output

#### Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.
warning.

**index**

The input index

**OutputState**

The output state. \((0 = \text{False} \ 1 = \text{True})\)

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

### 5.8.22 IFSetRatio.vi

**IFSetRatio.vi**

Set the ratio metric state for this board.

**Device In**

Device # Identification.
error in (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Ratiometric

The ratiometric state. ( 0 = False 1 = True )

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or
5.8.23 IFSetTrig.vi

IFSetTrig.vi

Set a sensor change trigger

Device In
Device # Identification.
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

index
The input index

Trig
The change trigger

Device Out
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

5.9  IR

5.9.1  IRCreate.vi

**IRCreate.vi**

Create a Phidget IR handle

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.
source
source string describes the origin of the error or warning.

error out
error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.
source
source string describes the origin of the error or warning.

Device Out
Device # identification

5.9.2 IREventCloseOnCode.vi

IREventCloseOnCode.vi

Close the Phidget IR OnCode event handle

Device In
Device # Identification.
error in (no error)

error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.9.3 **IREventCloseOnLearn.vi**

**IREventCloseOnLearn.vi**
Close the Phidget IR OnLearn event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
**5.9.4 IREventCloseOnRawData.vi**

Close the Phidget IR OnRawData event handle

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.
Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.9.5 IREventCreateOnCode.vi

IREventCreateOnCode.vi

Set up a Phidget IR OnCode event handle

Device In

error in (no error)

Device Out

Event Registration Refnum Out

error out

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

**code** is the error or warning code.

source

**source** describes the origin of the error or warning.

Device Out

Same as Device In error out

**error out** passes error or warning information out of a VI to be used by other VIs.

status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

**code** is the error or warning code.

source

**source** string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.9.6 IREventCreateOnLearn.vi

**IREventCreateOnLearn.vi**

Set up a Phidget IR OnLearn event handle.

- **Device In**
  - Device # Identification.

- **error in** (no error)

- **Device Out**
  - Event Registration Refnum Out error out
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification
5.9.7 IREventCreateOnRawData.vi

IREventCreateOnRawData.vi

Set up a Phidget IR OnRawData event handle.

Device In

Device # Identification.

error in (no error)

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
IReEventExeOnCode.vi

This is called when a code has been received that could be automatically decoded. Data is returned as an array with MSB in index 0. Bit count and a repeat flag are also returned. Repeats are detected as either the same code repeated in < 100ms or as a special repeat code.

- **Device In**: Device # Identification.
- **error in (no error)**: error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**: status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**: code is the error or warning code.
source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

text error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

text code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

BitCount

The bit count of the code

Repeat

Returns the repeats

Data

A user array to store the code data in

Data

Event Registration Refnum Out
5.9.9 **IREventExeOnLearn.vi**

**IREventExeOnLearn.vi**

This is called when a code has been received for long enough to be learned. The returned CodeInfo structure can be used to retransmit the same code.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**
Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**CodeInfo**

This contains all information needed to transmit a code. Please refer to [Phidgets Constants - CodeInfo](#).

**BitCount**
**Encoding**
**Length**
**Gap**
**Trail**
**Header 1**
**Header 2**
**One 1**
**One 2**
**Zero 1**
**Zero 2**
**Repeat 1**
Repeat 2
Repeat 3
Repeat 4
Repeat 5
Repeat 6
Repeat 7
Repeat 8
Repeat 9
Repeat 10
Repeat 11
Repeat 12
Repeat 13
Repeat 14
Repeat 15
Repeat 16
Repeat 17
Repeat 18
Repeat 19
Repeat 20
Repeat 21
Repeat 22
Repeat 23
Repeat 24
Repeat 25
Repeat 26

min_repeat
toggle_mask 1
toggle_mask 2
toggle_mask 3
toggle_mask 4
toggle_mask 5
toggle_mask 6
toggle_mask 7
toggle_mask 8
5.9.10 IREventExeOnRawData.vi

IREventExeOnRawData.vi

This is called when raw data has been read from the device. Raw data always starts with a space and ends with a pulse.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Data**

A user array for raw data to be written into
5.9.11 IRGetLastCode.vi

IRGetLastCode.vi

Get the last code that was received.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DataLength In (16)

Length of the user array - should be at least 16. This is set to the amount of data actually received.
written to the array

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

dataLength

Length of the user array.

bitCount

Set to the bit count of the code

data

A user array to store the code

5.9.12 IRGetLastLearnedCode.vi

IRGetLastLearnedCode.vi

Get the last code that was learned.
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**DataLength In (16)**

Length of the user array - should be at least 16. This is set to the amount of data actually written to the array.

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

dataLength

Length of the user array

CodeInfo

The CodeInfo structure for the learned code. Please refer to Phidgets Constants -> CodeInfo

BitCount

Encoding

Length

Gap

Trail

Header 1

Header 2

One 1

One 2

Zero 1

Zero 2

Repeat 1

Repeat 2

Repeat 3

Repeat 4

Repeat 5

Repeat 6

Repeat 7
Repeat 8
Repeat 9
Repeat 10
Repeat 11
Repeat 12
Repeat 13
Repeat 14
Repeat 15
Repeat 16
Repeat 17
Repeat 18
Repeat 19
Repeat 20
Repeat 21
Repeat 22
Repeat 23
Repeat 24
Repeat 25
Repeat 26
min_repeat
toggle_mask 1
toggle_mask 2
toggle_mask 3
toggle_mask 4
toggle_mask 5
toggle_mask 6
toggle_mask 7
toggle_mask 8
toggle_mask 9
toggle_mask 10
toggle_mask 11
toggle_mask 12
toggle_mask 13
toggle_mask 14
5.9.13 IRGetRawData.vi

IRGetRawData.vi

Read any available raw data. This should be polled continuously (every 20ms) to avoid missing data. Read data always starts with a space and ends with a pulse.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
DataLength In

The maximum amount of data to read. This is set to the actual amount of data read.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

dataLength

The amount of data to read

data

A user array for raw data to be written into

5.9.14 IROpen.vi

IROpen.vi

Open a PhidgetIR.
**Serial Number**

Serial Number. Specify -1 to open any.

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**error in (no error)**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device In**

Device # Identification. This function will create a new device identification if it's 0

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.9.15 IRTransmit.vi

IRTransmit.vi

Transmit a code according to the settings in a CodeInfo structure.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

CodeInfo

The CodeInfo structure specifying to send the code. Anything left as null to select default is filled in for the user. Please refer to Phidgets Constants -> CodeInfo

BitCount
Encoding
Length
Gap
Trail
Header 1
Header 2
One 1
One 2
Zero 1
Zero 2
Repeat 1
Repeat 2
Repeat 3
Repeat 4
Repeat 5
Repeat 6
Repeat 7
Repeat 8
Repeat 9
Repeat 10
Repeat 11
Repeat 12
Repeat 13
Repeat 14
Repeat 15
Repeat 16
Repeat 17
Repeat 18
Repeat 19
Repeat 20
Repeat 21
Repeat 22
Repeat 23
Repeat 24
Repeat 25
Repeat 26
min_repeat
toggle_mask 1
toggle_mask 2
toggle_mask 3
toggle_mask 4
toggle_mask 5
toggle_mask 6
toggle_mask 7
toggle_mask 8
toggle_mask 9
toggle_mask 10
toggle_mask 11
toggle_mask 12
toggle_mask 13
toggle_mask 14
toggle_mask 15
toggle_mask 16
CarrierFrequency
DutyCycle
Data
The code to send. Data is transmitted MSBit first. MSByte is in array index 0. LSBit is right justified, so MSBit may be in bit positions 0-7 in array index 0 depending on the bit count.

**Data**

**Device Out**

Same as Device In

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

### 5.9.16 IRTransmitRaw.vi

**IRTransmitRaw.vi**

Transmit RAW data as a series of pulses and spaces.
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

length

The length of the data array. Maximum length is 1024, but streams should be kept much shorter, i.e. < 100ms between gaps.

carrierFrequency

The Carrier Frequency in Hz. Leave as 0 for default.

dutyCycle

The Duty Cycle (10-50). Leave as 0 for default.

gap

The gap time in us. This guarantees a gap time (no transmitting) after the data is sent, but can be set to 0.

Data

The data to send. The array must start and end
with a pulse and each element is a positive time in us.

data

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.9.17  IRTransmitRepeat.vi

IRTransmitRepeat.vi

Transmits a repeat of the last transmitted code. Depending of the CodeInfo structure, this may be a retransmission of the code itself, or there may be a special repeat code

Device In

Device # Identification.

error in (no error)

Device Out

error out

5.9.17  IRTransmitRepeat.vi
from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.10 LED

5.10.1 LEDCount.vi

LEDCount.vi

Get the number of LEDs supported by this board.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count

The LED count

5.10.2 LEDCreate.vi

**LEDCreate.vi**

Create a Phidget LED handle

| error in | Device Out | error out |

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.
error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Device # identification.

5.10.3 LEDGetBrightness.vi

LEDGetBrightness.vi

Get the brightness of an LED

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

brightness

The LED brightness (0-100).

5.10.4 LEDGetCurrentLimit.vi

LEDGetCurrentLimit.vi
Get the current limit. This is for all outputs

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

CurrentLimit

The current limit

5.10.5 LEDGetCurrentLimitIndexed.vi

LEDGetCurrentLimitIndexed.vi

Gets an indexed current limit.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
Specific Modules

index

The LED index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Current Limit

The current limit

5.10.6 LEDGetVoltage.vi

LEDGetVoltage.vi

Get the output voltage. This is for all outputs.

Device In

Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Voltage

The output voltage
5.10.7 **LEDOpen.vi**

**LEDOpen.vi**

Open a Phidget LED.

**Serial Number**

Serial Number. Specify -1 to open any.

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**error in (no error)**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device In**

Device # identification. This function will create a new device identification if it's 0 or invalid

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully
attached, or FALSE otherwise.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

5.10.8 **LEDSetBrightness.vi**

**LEDSetBrightness.vi**

Set the brightness of an LED

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The LED index

**brightness**

The LED brightness (0-100).

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.10.9 LEDSetCurrentLimit.vi

**LEDSetCurrentLimit.vi**
Set the current limit. This is for all outputs.

Device In

Device # Identification.

*error in* (no error)

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

CurrentLimit

The current limit

Device Out

Same as Device In

*error out*

error out can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.10.10 LEDSetCurrentLimitIndexed.vi

**LEDSetCurrentLimitIndexed.vi**

Sets the indexed current limit.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.10.11 LEDSetVoltage.vi

**LEDSetVoltage.vi**

Set the output voltage. This is for all outputs.

*Device In*

Device # Identification.

*error in (no error)*

*Device Out*

Set output voltage.

*error out*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

*status*

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

*code*

*code* is the error or warning code.

*source*

*source* describes the origin of the error or warning.
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Voltage**

The output voltage

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.11 MotorControl

5.11.1 MCCreate.vi

**MCCreate.vi**

Create a Phidget Motor Control handle

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.
5.11.2 MCEventCloseCurrent.vi

MCEventCloseCurrent.vi

Close the Phidget Motor Control current change event handle

Device In

Device # Identification.

error in (no error)

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out
error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.11.3 MCEventCloseEMF.vi

MCEventCloseEMF.vi

Close the Phidget Motor Control EMF change event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.11.4 MCEventCloseInput.vi

MCEventCloseInput.vi

Close the Phidget Motor Control input change event handle

Device In

error in (no error)

Device Out

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Event Registration Refnum In**

**Device Out**

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

5.11.5 MCEventClosePositionChange.vi

**MCEventClosePositionChange.vi**

Close the Phidget motor controller position change event handler
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error
or warning.

5.11.6 **MCEventClosePositionUpdate.vi**

MCEventClosePositionUpdate.vi

Close the phidget motor controller position update event handle

![Diagram of MCEventClosePositionUpdate.vi](image)

**Device In**

- **error in (no error)**

  *error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

  - **status**
    - *status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**Device Out**

- **code**
  - *code* is the error or warning code.

  - **source**
    - *source* describes the origin of the error or warning.

**Event Registration Refnum In**

**Device Out**

- **error out**
  - *error out* passes error or warning information out of a VI to be used by other VIs.

  - **status**
    - *status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
5.11.7 MCEventCloseSensorUpdate.vi

MCEventCloseSensorUpdate.vi

Close the Phidget Motor controller Sensor update event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In
Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.11.8 MCEventCloseVelocity.vi

MCEventCloseVelocity.vi

Close the Phidget Motor Control velocity change event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

code

code passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.11.9 MCEventCreateCurrentChange.vi

MCEventCreateCurrentChange.vi

Device In —> Event —> Device Out

error in (no error) —> Event Registration Refnum Out —> error out

Device In
**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**

5.11.10 MCEventCreateCurrentUpdate.vi

**MCEventCreateCurrentUpdate.vi**
Device In

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

Event Registration Refnum Out
5.11.11 MCEventCreateEMF.vi

**MCEventCreateEMF.vi**

Create the Phidget Motor Control EMF change event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
Create the Phidget Motor Control input change event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.11.13 MCEventCreatePositionChange.vi

MCEventCreatePositionChange.vi

Create the Phidget motor controller position change event handler

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.11.14 MCEventCreatePositionUpdate.vi

MCEventCreatePositionUpdate.vi

Create the phidget motor controller position update event handle
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.11.15 MCEventCreateSensorUpdate.vi

MCEventCreateSensorUpdate.vi

Create the Phidget Motor controller Sensor update event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

### 5.11.16 MCEventCreateVelocity.vi

**MCEventCreateVelocity.vi**

Create the Phidget Motor Control velocity change event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

*code* is the error or warning code.

source

*source* describes the origin of the error or warning.

Device Out

Same as Device In

error out

*error out* passes error or warning information out of a VI to be used by other VIs.

status

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

*code* is the error or warning code.

source

*source* string describes the origin of the error or warning.

5.11.17 MCEventExeCurrent.vi

**MCEventExeCurrent.vi**

Executes on an MC Current change event
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Current

The current

index

The motor index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

© 2014 Phidgets Inc.
code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Event # Identification

5.11.18 MCEventExeEMF.vi

MCEventExeEMF.vi

Executes on an EMF event

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Voltage

The voltage

index

The motor index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Event # Identification
5.11.19 MCEventExeInput.vi

MCEventExeInput.vi

Executes when the state of a digital input changes.

- **error in (no error)**

  `error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**

  `status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**

  `code` is the error or warning code.

- **source**

  `source` describes the origin of the error or warning.

- **Device In**

  Device # Identification.

- **Event Registration Refnum In**

  Event # Identification

- **index**

  The input index

- **InputState**

  The state of the input
error out

**error out** passes error or warning information out of a VI to be used by other VIs.

status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

**source** string describes the origin of the error or warning.

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Event # Identification

### 5.11.20 MCEventExePositionChange.vi

**MCEventExePositionChange.vi**

Executes if the motor's position changes.

error in (no error)

can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device In

Device # Identification.

Event Registration Refnum In

Event # Identification

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

Event?
Returns TRUE if the event has executed, or FALSE otherwise.

**Event Registration Refnum Out**

**Event # Identification**

**index**

The encoder index

**time**

The amount of time in which PositionChange counts occurred in 1/3 ms

**position**

The change in encoder position

### 5.11.21 MCEventExePositionUpdate.vi

**MCEventExePositionUpdate.vi**

Executes every 8ms regardless of if the position changed or not.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source describes the origin of the error or warning.

Device In

Device # Identification.

Event Registration Refnum In

Event # Identification

index

The motor index

position

The change in position

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Event # Identification
5.11.22 MCEventExeSensorUpdate.vi

MCEventExeSensorUpdate.vi

An event containing sensor value information for sensors plugged into the Analog Inputs.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device In

Device # Identification.

Event Registration Refnum In

Event # Identification

index

The motor index

value

The sensor's value
**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Event Registration Refnum Out**

Event # Identification

---

5.11.23 **MCEventExeVelocity.vi**

**MCEventExeVelocity.vi**

Executes when the velocity a motor is being driven at changes

**Device In**

Device # Identification.  

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

---

**Event Registration Refnum In**

**Event # Identification**

**Velocity**

The velocity

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**
source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Event # Identification

5.11.24 MCGetAcceleration.vi

MCGetAcceleration.vi

Gets the motor's acceleration

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Acceleration

The acceleration

5.11.25 MCGetAccelerationMax.vi

MCGetAccelerationMax.vi

Gets the maximum settable acceleration.

Device In

Device # Identification.

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Acceleration Max**
5.11.26 MCGetAccelerationMin.vi

MCGetAccelerationMin.vi

Gets the minimum settable acceleration.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

**Device Out**

Same as Device In

**error out**
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Acceleration Min

Minimum acceleration.

5.11.27 MCGetBackEMFValue.vi

MCGetBackEMFValue.vi

Gets the back EMF voltage

Device In

Device # Identification.

error in (no error)

event in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

MCGetBackEMFValue.vi

© 2014 Phidgets Inc.
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Voltage

The back EMF voltage
5.11.28 MCGetBraking.vi

MCGetBraking.vi

Gets the braking amount for the motor at rest.

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

index
The motor index

Device Out
Same as Device In

error out
error out can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Braking

The braking percentage (0-100%)

5.11.29 MCGetCurrent.vi

MCGetCurrent.vi

Get the current flowing through the motor.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Current

The current

5.11.30 MCGetEMFState.vi

MCGetEMFState.vi

Get the EMF sensing state. When back-EMF sensing is enabled, the motor will coast (freewheel) 5% of the time while the back EMF measurement is taken.
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

EMFState (0,1)

The EMF sensing state. Possible values are 0 for False, 1 for True and others for undefined.

EMFState

The EMF sensing state. Possible values are True for Enabled and False for Not Enabled.

5.11.31 MCGetEncoderCount.vi

MCGetEncoderCount.vi

Returns the number of encoder inputs.

Device In Device Out

Encoder # Identification.

error in (no error)

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.11.32 MCGetEncoderPosition.vi

MCGetEncoderPosition.vi

Gets the encoder position.

Device In

Device In

Encoder Count

Device Out

Device Out

Encoder Count

error in (no error)

error out

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The encoder index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source
describes the origin of the error or warning.

Position
Encoder position

5.11.33 MCGetInputCount.vi

MCGetInputCount.vi
Get the number of digital inputs on the motor controller.

Device In
Device # Identification.

error in (no error)
can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
is the error or warning code.

Device Out
Same as Device In
Specific Modules

error out

text

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
	source

text

source describes the origin of the error or warning.

InputCount

The number of digital inputs.

5.11.34 MCGetInputState.vi

MCGetInputState.vi

Gets the state of the digital input.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

InputState (0,1)

The input state. Possible values are 0 for False, 1 for True and others for undefined

InputState
5.11.35 MCGetMotorCount.vi

MCGetMotorCount.vi

Gets the number of motors the controller can handle.

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Motor Count

The number of motors

5.11.36 MCGetRatiometricState.vi

MCGetRatiometricState.vi

Gets the ratiometric state for the analog inputs.

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Ratiometric (0,1)

The ratiometric state. Possible values are 0 for False, 1 for True and others for undefined

Ratiometric

The ratiometric state of the input (boolean)

5.11.37 MCGetSensorCount.vi

MCGetSensorCount.vi

The number of analog sensors the MC can accommodate.
Device In

Device # Identification.

**error in** (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source describes the origin of the error or warning.

SensorCount

The number of sensors

5.11.38 MCGetSensorValue.vi

MCGetSensorValue.vi

Gets the current value for a sensor input. Range is 0-1000

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index
The sensor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Value**

The sensor value

### 5.11.39 MCGetSensorValueRAW.vi

**MCGetSensorValueRAW.vi**

Gets the raw 12-bit value for a sensor input. Range is 0-4096

**Device In**

Device # Identification.

**error in (no error)**
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The sensor index

**Device Out**

Same as Device In

**error out**

e**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.11.40 MCGetSupplyVoltage.vi

MCGetSupplyVoltage.vi

Gets the voltage of the poser supply.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Voltage**

The voltage

### 5.11.41 MCGetVelocity.vi

**MCGetVelocity.vi**

Velocity is the percentage of time the motor is being powered for, from -100% to 100%

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Velocity

The velcoity

5.11.42 MCOpen.vi

MCOpen.vi
Open a Phidget Motor controller

- **Serial Number**
  - Specify -1 to open any.

- **milliseconds**
  - Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

- **error in (no error)**
  - error out passes error or warning information out of a VI to be used by other VIs.

- **status**
  - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  - code is the error or warning code.

- **source**
  - source string describes the origin of the error or warning.

- **Device In**
  - Device # identification. This function will create a new device identification if it's 0 or invalid

- **Serial Number Return**
  - Serial Number of the opened phidget

- **Attached?**
  - Returns TRUE is the device successfully attached, or FALSE otherwise.

- **error out**
  - error out passes error or warning information
out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

### 5.11.43 MCSetAcceleration.vi

**MCSetAcceleration.vi**

Set the motor's acceleration (in % duty cycle/second)

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
**5.11.44 MCSetBraking.vi**

**MCSetBraking.vi**

Set the braking amount for a motor at rest.
Device In

Device # Identification.

**error in** *(no error)*

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Braking**

The breaking duty cycle

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.11.45 MCSetEMFState.vi

MCSetEMFState.vi

Set the EMF Sensing State

Device In

Device In # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
warning.

Set EMF

The EMF Sensing State

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.11.46 MCSetEncoderPosition.vi

MCSetEncoderPosition.vi

Set the encoder position

Device In

index

Device Out

Encoder Set Position

error in (no error)

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The encoder index

**Position**

The position

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.11.47 MCSetRatiometricState.vi

MCSetRatiometricState.vi

Set the ratiometric state for the device.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Ratiometric

The ratiometric state

Device Out
Same as Device In

**error out**

er**ror in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

c**ode** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.11.48 MCSetVelocity.vi

**MCSetVelocity.vi**

Set the Motor controller velocity in % duty cycle from -100% to 100%

**Device In**

er**ror in** (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

Velocity

Device Out

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.12 PHSensor

5.12.1 PHCreate.vi

PHCreate.vi

Create a Phidget PH Sensor handle.
error in
error out passes error or warning information out of a VI to be used by other VIs.
status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code
code is the error or warning code.
source
source string describes the origin of the error or warning.
error out
error out passes error or warning information out of a VI to be used by other VIs.
status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code
code is the error or warning code.
source
source string describes the origin of the error or warning.

5.12.2 PHEventClose.vi

PHEventClose.vi

Close the Phidget PH Sensor change event handle
Device In

Device # Identification.

error in (no error)

text

text

text

text

text

text

Device Out

Same as Device In

text

text

text

text

error out

text

text

text

text

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

text

text

text

text

text

text

text

text

text

text

text

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

text

text

text

text

text

text

text

text

text

text

text

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

text

text

text

text

text

text

text

text

text

text

text

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

text

text

text

text

text

text

text

text

text

text

text

code
code is the error or warning code.

source

source string describes the origin of the error or warning.

5.12.3 PHEventCreate.vi

PHEventCreate.vi

Set up a PH change event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

### 5.12.4 PHEventExe.vi

**PHEventExe.vi**

This is called when the PH changes by more than the change trigger

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

<table>
<thead>
<tr>
<th>code</th>
</tr>
</thead>
</table>

**Code** is the error or warning code.

**Source**

**Source** describes the origin of the error or warning.

---

**Event Registration Refnum In**

Event # Identification

**PH**

The PH

**Device Out**

Same as Device In

**Error Out**

**Error out** passes error or warning information out of a VI to be used by other VIs.

**Status**

**Status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

<table>
<thead>
<tr>
<th>code</th>
</tr>
</thead>
</table>

**Code** is the error or warning code.

**Source**

**Source** string describes the origin of the error or warning.

---

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Event Registration Refnum Out**

Same as the Event Registration Refnum In
5.12.5 PHGetPH.vi

PHGetPH.vi

Get the measured PH

Device In

Device In

error in (no error)

Device Out

PH

code

source

status

Device In

Device # Identification.

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PH

The PH

5.12.6 PHGetPHMax.vi

PHGetPHMax.vi

Get the maximum PH that the sensor could report

Device In

dervice # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

Device Out
Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PH Max
The maximum PH

5.12.7 PHGetPHMin.vi

PHGetPHMin.vi
Get the minimum PH that the sensor could report

Device In

Device # Identification.

error in (no error)

Device Out

error out

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**PH Min**

The minimum PH
5.12.8 PHGetPHTrigger.vi

**PHGetPHTrigger.vi**

Get the PH change trigger

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

**PH Trigger**

The change trigger

### 5.12.9 PHGetPotential.vi

**PHGetPotential.vi**

Get the sensed potential

- **Device In**
  - Device # Identification.
  - **error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**
source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Potential

The potential

5.12.10 PHGetPotentialMax.vi

PHGetPotentialMax.vi

Get the maximum potential that can be sensed

Device In

Device # Identification.

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Potential Max**

The maximum potential
5.12.11 PHGetPotentialMin.vi

PHGetPotentialMin.vi

Get the minimum potential that can be sensed

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Potential Min

The minimum potential

5.12.12 PHOpen.vi

PHOpen.vi

Open a Phidget PH Sensor

Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

code out

code out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.12.13 PHSetTemperature.vi

PHSetTemperature.vi

Set the temperature to be used for PH calculations
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Temperature

The temperature (degrees ceclius). By default this is 20.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.12.14 PHSetTrig.vi

**PHSetTrig.vi**

Set the PH change trigger.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Trigger**
5.13 RFID

5.13.1 RFIDCreate.vi

RFIDCreate.vi

Create a Phidget RFID handle

The change trigger

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Device Out**

Device # identification

### 5.13.2 RFIDEventClose.vi

**RFIDEventClose.vi**

Close the Phidget RFID Tag or TagLost event handle

- **Device In**
  - Device # Identification.
  - error in (no error)

- **Device Out**
  - error out
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** string describes the origin of the error or warning.
RFIDEVENTCLOSE2.VI

5.13.3 RFIDEVENTCLOSE2.VI

Close the Phidget RFID Tag or TagLost event handle (2)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
**5.13.4 RFIDEVENTCLOSEOUTPUT.VI**

**RFIDEVENTCLOSEOUTPUT.VI**

Close the Phidget RFID output change event handle

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.13.5 RFIDEventCreateOutput.vi

RFIDEventCreateOutput.vi

Set up an output change event handle.

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.13.6 RFIDEeventCreateTag.vi

RFIDEeventCreateTag.vi

Set up a tag change event handle
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out
5.13.7 RFIDEventCreateTag2.vi

RFIDEventCreateTag2.vi

Set up a tag change event handle (2)

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.13.8 RFIDEventCreateTagLost.vi

RFIDEventCreateTagLost.vi

Set up a tag lost event handle

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

### 5.13.9 RFIDEEventCreateTagLost2.vi

**RFIDEEventCreateTagLost2.vi**

Set up a tag lost event handle (2)

Device In

type: [Device # Identification]

type: [error in (no error)]

description: [error in] can accept error information wired from VIs previously called. Use this

Device Out

Event Registration Refnum Out

type: [error out]

description: 

**Device In**

Device # Identification.

**error in (no error)**

description: 

**error in** can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

5.13.10 RFIDEventExe.vi

RFIDEventExe.vi
This is called when the Phidget RFID Tag or TagLost event changes. Tag Event is called when a tag is first detected by the reader. TagLost Event is called when a tag is no longer detected by the reader.

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

*error out* passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Tag

The tag.

Event Registration Refnum Out

Same as the Event Registration Refnum In

5.13.11 RFIDEventExe2.vi

RFIDEventExe2.vi

This is called when the Phidget RFID Tag or TagLost event changes. (2)

Device In

Device # Identification.

error in (no error)

error in (no error) can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Tag**
5.13.12 RFIDEventExeOutput.vi

RFIDEventExeOutput.vi

This is called when an output changes

- **error in (no error)**
- **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
- **code** is the error or warning code.

- **source**
- **source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification
Device In

Device # Identification.

index

The output index

OutputState

The output state. Possible values are 0 for False, 1 for True and others for undefined

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.13.13 RFIDGetAntennaState.vi

RFIDGetAntennaState.vi

Get the state of the antenna
Device In

Device # Identification.

*error in (no error)*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

*status*

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

*code*

*code* is the error or warning code.

*source*

*source* describes the origin of the error or warning.

Device Out

Same as Device In

*error out*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

*status*

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

*code*
5.13.14 RFIDGetLastTag.vi

**RFIDGetLastTag.vi**

Get the last tag read by the reader. This tag may or may not still be on the reader.

- **Device In**
  - Device # Identification.
  - **error in (no error)**
  - **error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  - **code** is the error or warning code.

- **source**
  - **source** describes the origin of the error or warning.
source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Tag

The tag. This must be an unsigned char array of size 5.

5.13.15 RFIDGetLastTag2.vi

RFIDGetLastTag2.vi

Get the last tag read by the reader. This tag may or may not still be on the reader. (2)
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device In**

Device # Identification.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

**Tag String**

The tag string

**Tag Protocol**
5.13.16 RFIDGetLEDState.vi

RFIDGetLEDState.vi

Get the state of the onboard LED

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**LED State**

The LED state. Possible values are 0 for False, 1 for True and others for undefined

**LED On?**

The LED state

5.13.17 **RFIDGetOutputCount.vi**

**RFIDGetOutputCount.vi**

Get the number of outputs supported by this board.

**Device In**

Device # Identification.

**error in (no error)**

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

count

The number of outputs.

5.13.18 RFIDGetOutputState.vi

RFIDGetOutputState.vi

Get the state of an output
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The output index

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**OutputState**

The output state. Possible values are 0 for False, 1 for True and others for undefined

**Output?**

The output state.

5.13.19 RFIDGetTagState.vi

**RFIDGetTagState.vi**

Get the tag present status. This is whether or not a tag is being read by the reader.

**Device In**

Device # Identification.

**error in (no error)**

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Tag State

The tag state. Possible values are 0 for False, 1 for True and others for undefined.

Tag On?

The tag state

5.13.20 RFIDOpen.vi

RFIDOpen.vi

Open a Phidget RFID
Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE if the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.13.21 RFIDSetAntennaState.vi

RFIDSetAntennaState.vi

Set the state of the antenna. Note that the antenna must be enabled before tags will be read.

Device In

Device # Identification.

erro in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source

source describes the origin of the error or warning.

AntennaState

The antenna state

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.13.22 RFIDSetLEDState.vi

RFIDSetLEDState.vi

Set the state of the onboard LED

Device In

Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

LEDState

Set the LED state

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.13.23 RFIDSetOutputState.vi

RFIDSetOutputState.vi

Set the state of an output

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

OutputState

Set the output state

index

The output index

Device Out

Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.13.24 RFIDWriteTag.vi

RFIDWriteTag.vi

Write an RFID Tag

Device In

Device Out

string

protocol

Lock

error in (no error)

device in (no error)

Device In

Device # Identification.

code

code can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

string

The data to send

protocol

The protocol to use

Lock

Locks the tag from further writes.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.14 Servo
5.14.1 ServoCount.vi

ServoCount.vi

Gets the number of motors supported by this controller.

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Count

The motor count.

5.14.2 ServoCreate.vi

ServoCreate.vi

Create a Phidget Servo handle.

error in

error in passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out
**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Device # identification

5.14.3 ServoEventClose.vi

**ServoEventClose.vi**

Close the Phidget Servo event handle.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.14.4 ServoEventCreate.vi

ServoEventCreate.vi

Set up a position change event handle

Device In  Device Out

error in (no error)  error out

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**
### 5.14.5 ServoEventExe.vi

**ServoEventExe.vi**

This is called when the Phidget Advanced Servo event occurs.

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Value**

The return value of the position.
index

The servo index

Device Out

Same as Device In

error out

code

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.14.6 ServoGetEngaged.vi

ServoGetEngaged.vi

Get the engaged state of a motor. This is whether the motor is powered or not

Device In

Device # Identification.
error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The servo index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

**EngagedState_out**

The engaged state. Possible values are 0 for False, 1 for True and others for undefined

**Engaged?**

The engaged state (Boolean type). Possible values are True for Engaged and False for Not Engaged.

---

### 5.14.7 ServoGetPos.vi

**ServoGetPos.vi**

Get the current position of a motor

#### Device In

Device # Identification.

#### error in (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

#### status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

#### code

code is the error or warning code.

#### source

**source** describes the origin of the error or warning.
index

The servo index

Device Out

Same as Device In

error out

device in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

pos_out

The position

5.14.8 ServoGetPosMax.vi

ServoGetPosMax.vi

Get the maximum position that a motor can go to

Device In

Device # Identification.

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The servo index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.14.9  ServoGetPosMin.vi

ServoGetPosMin.vi

Get the minimum position that a motor can go to

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**index**

The servo index

**Device Out**

Same as Device In

**error out**

The maximum position
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

posmin_out

The minimum position

5.14.10 ServoGetServoType.vi

ServoGetServoType.vi

Get the servo type of a motor

Device In

GetType

Device Out

error in (no error)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

GetServoType

Returns the servo type. This is an enum. Please refer to Phidgets Constants -> ServoType

5.14.11 ServoOpen.vi

ServoOpen.vi
Open a PhidgetServo.

- **Serial Number**
  - Serial Number. Specify -1 to open any.

- **milliseconds**
  - Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

- **error in (no error)**
  - **error out** passes error or warning information out of a VI to be used by other VIs.

- **status**
  - **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

- **code**
  - **code** is the error or warning code.

- **source**
  - **source** string describes the origin of the error or warning.

- **Device In**
  - Device # Identification. This function will create a new device identification if it's 0

- **Serial Number Return**
  - Serial Number of the opened phidget

- **Attached?**
  - Returns TRUE is the device successfully attached, or FALSE otherwise.

- **error out**
  - **error out** passes error or warning information
out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.14.12 ServoSetEngaged.vi

ServoSetEngaged.vi

Set the engaged state of a motor. This is whether the motor is powered or not

Device In

Device # Identification.

to (no error)

error (no error)

error in (no error)

Device Out

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
Specific Modules

5.14.13 ServoSetPos.vi

ServoSetPos.vi

Set the position of a motor.
### Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**pos_in**

The servo position

**index**

The servo index

### Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.14.14 ServoSetServoParameters.vi

ServoSetServoParameters.vi

Set the servo parameters of a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

index

The servo index

min_us

The minimum supported PCM in microseconds

max_us

The maximum supported PCM in microseconds

degrees

The degrees of rotation defined by the given PCM range

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.14.15 ServoSetServoType.vi

ServoSetServoType.vi
Set the servo type of a motor

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The servo index

**setServoType**

The servo type. This is an enum. Please refer to Phidgets Constants -> ServoType

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.15 Spatial

#### 5.15.1 SpatialCreate.vi

**SpatialCreate.vi**

Create a Phidget Spatial handle

- **error in**
- **Device Out**
- **error out**

**error in**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** string describes the origin of the error or warning.
error out

**error out** passes error or warning information out of a VI to be used by other VIs.

status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

**source** string describes the origin of the error or warning.

Device Out

Device # identification

5.15.2 SpatialEventClose.vi

**SpatialEventClose.vi**

Close the Phidget Spatial event handle

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

**status** is TRUE (X) if an error occurred or
FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

### 5.15.3 SpatialEventCreate.vi

**SpatialEventCreate.vi**

Set up a data change event handle

Device In

error in (no error)

Device Out

Event Registration Refnum Out

error out

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**
5.15.4 SpatialEventExe.vi

SpatialEventExe.vi

This is called when data come. It is called at SpatialGetDataRate, up to 8ms. For the rate faster than 8ms, multiple sets of data are supplied in a single event.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device In

Device # Identification.

Event Registration Refnum In

Event # Identification

error out

error out passes error or warning information
out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Same as Device In

**Event?**

Returns TRUE if the event has executed, or FALSE otherwise.

**Event Registration Refnum Out**

Event # Identification

**Output**

The Output Data

**acc0**

Acceleration X.

**acc1**

Acceleration Y.

**acc2**

Acceleration Z

**ang0**

Angular rate X

**ang1**

Angular rate Y

**ang2**
5.15.5 SpatialGetAcce.vi

**SpatialGetAcce.vi**

Get the current acceleration data of an axis

Device In

Device # Identification.

*error in (no error)*

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code.

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The acceleration index. (x, y, z)

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other
VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Acce

The acceleration in gs

5.15.6 SpatialGetAcceAxisCount.vi

SpatialGetAcceAxisCount.vi

Get the number of acceleration axes supplied by this board
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
5.15.7 SpatialGetAcceMax.vi

SpatialGetAcceMax.vi

Get the maximum acceleration supported by an axis

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The axis count
The acceleration index. \((x, y, z)\)

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**AcceMax**

The maximum acceleration

### 5.15.8 SpatialGetAcceMin.vi

**SpatialGetAcceMin.vi**

Get the minimum acceleration supported by an axis

**Device In**

Device # Identification.

**error in (no error)**
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The axis index (x, y, z)

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.15.9 SpatialGetAngRate.vi

SpatialGetAngRate.vi

Get the current angular rate of an axis

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The axis index (x, y, z)

Device Out

Same as Device In

error out
5.15.10 SpatialGetAngRateMax.vi

SpatialGetAngRateMax.vi

Get the maximum angular rate supported by an axis

Device In

Device Out

error in (no error)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

AngRate

The angular rate in degrees/second
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The axis index (x, y, z)

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

AngRateMax

The maximum angular rate
5.15.11 SpatialGetAngRateMin.vi

SpatialGetAngRateMin.vi

Get the minimum angular rate supported by an axis

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The axis index (x, y, z)

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**AngRateMin**

The minimum angular rate

### 5.15.12 SpatialGetCompassAxisCount.vi

**SpatialGetCompassAxisCount.vi**

Get the number of compass axes supplied by this board

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
code is the error or warning code. 

source 

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status 

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code. 

source 

source describes the origin of the error or warning.

CompassAxisCount

The number of compass axes

5.15.13 SpatialGetDataRate.vi

SpatialGetDataRate.vi

Get the data rate

Device In 

error in (no error)
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.15.14 SpatialGetDataRateMax.vi

SpatialGetDataRateMax.vi

Get the maximum supported data rate

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DataRateMax

The data rate in milliseconds

5.15.15 SpatialGetDataRateMin.vi

SpatialGetDataRateMin.vi

Get the minimum supported data rate

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**DataRateMin**

The data rate in milliseconds

5.15.16 SpatialGetGyroAxisCount.vi

**SpatialGetGyroAxisCount.vi**

Get the number of gyroscope axes supplied by this board
Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source

source describes the origin of the error or warning.

GyroAxisCount

The number of gyro axes.

5.15.17 SpatialGetMagField.vi

SpatialGetMagField.vi

Get the current magnetic field strength of an axis

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index
The axis index (x, y, z)

Device Out

Same as Device In

error out

device in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

MagField

The magnetic field strength in Gauss

5.15.18 SpatialGetMagFieldMax.vi

SpatialGetMagFieldMax.vi

Get the maximum magnetic field strength supported by an axis.

Device In

Device # Identification.

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The axis index (x, y, z)

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.15.19 SpatialGetMagFieldMin.vi

SpatialGetMagFieldMin.vi

Get the minimum magnetic field strength supported by an axis.

Device In
- Device # Identification.

error in (no error)
- error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
- code is the error or warning code.

source
- source describes the origin of the error or warning.

index
- The axis index (x, y, z)

Device Out
- Same as Device In

error out

MagFieldMax
- The maximum magnetic field strength in Gauss
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**MagFieldMin**

The minimum magnetic field strength in Gauss.

### 5.15.20 SpatialOpen.vi

**SpatialOpen.vi**

Open a PhidgetSpatial

**Serial Number**

Serial Number. Specify -1 to open any.

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**error in (no error)**

**error out** passes error or warning information out of a VI to be used by other VIs.
**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Device In**

Device # identification. This function will create a new device identification if it's 0 or invalid.

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Device Out**

Same as Device In
5.15.21 SpatialResetCompassCorrectionParameters.vi

**SpatialResetCompassCorrectionParameters.vi**

Reset the compass correction factors. Magnetic field data will be presented directly as reported by the sensor.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.15.22 SpatialSetCompassCorrectionParameters.vi

**SpatialSetCompassCorrectionParameters.vi**

Set the compass correction factors. This can be used to correcting any sensor errors, including hard and soft iron offsets and sensor error factors.

#### Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

#### status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

#### code

**code** is the error or warning code.

#### source

**source** describes the origin of the error or warning.
Settings

The correction parameters.

* magField
* offset0
* offset1
* offset2
* gain0
* gain1
* gain2
* T0
* T1
* T2
* T3
* T4
* T5

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.15.23 SpatialSetDataRate.vi

SpatialSetDataRate.vi

Set the data rate. Note that data at rates faster than 8ms will be delivered to events as an array of data.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

DateRate

The data rate.

Device Out

Same as Device In.

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
describes the origin of the error or warning.

5.15.24 SpatialZeroGyro.vi

SpatialZeroGyro.vi

Zero the gyroscope. This takes about two seconds and the gyro axes will report 0 during the process. This should only be called when the board is not moving

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source
describes the origin of the error or warning.

Device Out
Same as Device In

error out
can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
describes the origin of the error or warning.

5.16 Stepper
5.16.1 StepperCreate.vi

StepperCreate.vi
Create a Phidget Stepper handle.

error in

error out passes error or warning information out of a VI to be used by other VIs.

status
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Device Out**

Device # identification

### 5.16.2 StepperEventCloseCurrentVelocity.vi

**StepperEventCloseCurrentVelocity.vi**

Close the Phidget Stepper current change event handle

Device In
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.
5.16.3 StepperEventCloseInput.vi

StepperEventCloseInput.vi

Close the Phidget Stepper input change event handle

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

Event Registration Refnum In
Event # Identification

Device Out
Same as Device In

error out
error out passes error or warning information out of a VI to be used by other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.16.4 StepperEventClosePosition.vi

StepperEventClosePosition.vi

Close the Phidget Stepper position change event handle

Device In

Device # Identification.

error in (no error)

effect in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
Event Registration Refnum In
Event # Identification

Device Out
Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.16.5 StepperEventClosePosition71.vi

StepperEventClosePosition71.vi

Close the Phidget Stepper position change event handle for Labview version 7.1 only

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

code

code out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.16.6 StepperEventCreateCurrent.vi

StepperEventCreateCurrent.vi

Set up a current change event handle
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

Event Registration Refnum Out
5.16.7 StepperEventCreateInput.vi

StepperEventCreateInput.vi

Set up an input change event handle

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

### 5.16.8 StepperEventCreatePosition.vi

**StepperEventCreatePosition.vi**

Set up a position change event handle

 Device In  
 error in (no error)  
 Device Out  
 Event Registration Refnum Out

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**
source describes the origin of the error or warning.

Device Out

Same as Device In

error out

epsilon out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.16.9 StepperEventCreatePosition71.vi

StepperEventCreatePosition71.vi

Set up a position change event handle for Labview version 7.1 only

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Event Registration Refnum Out**

Event # Identification

5.16.10 StepperEventCreateVelocity.vi

StepperEventCreateVelocity.vi
Set up a velocity change event handle

Device In

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Event # Identification

5.16.11 StepperEventExeCurrentVelocity.vi

StepperEventExeCurrentVelocity.vi

This is called when the Phidget Stepper Current/Velocity event changes

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In
Event # Identification

Value

The return value of related event. (E.g. for velocity change event, this value is velocity.)

index

The motor index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.16.12 StepperEventExeInput.vi

**StepperEventExeInput.vi**

This is called when the Phidget Stepper Input event changes.
**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification

**Device In**

Device # Identification.

**index**

The motor index

**Value**

The input

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In

Device Out

Same as Device In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.16.13 StepperEventExePosition.vi

StepperEventExePosition.vi

This is called when the Phidget Stepper Position event changes

Device In

Device # Identification.

code in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

index

The motor index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Value

The position

Event?

Returns TRUE if the event has executed, or
FALSE otherwise.

**Event Registration Refnum Out**

Same as the Event Registration Refnum In

### 5.16.14 StepperEventExePosition71.vi

**StepperEventExePosition71.vi**

This is called when the Phidget Stepper Position event changes. This function is for Labview version 7.1 only

![Diagram](image-url)

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Event Registration Refnum In**

Event # Identification
Specific Modules

5.16.15 StepperGetAcceleration.vi

StepperGetAcceleration.vi

Get the last set acceleration for a motor

Device In

Device Out

index

The motor index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

Event Registration Refnum Out

Same as the Event Registration Refnum In

value

The position

© 2014 Phidgets Inc.
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source describes the origin of the error or warning.

Acce
The acceleration

5.16.16 StepperGetAccelerationMax.vi

StepperGetAccelerationMax.vi
Get the maximum acceleration supported by a motor

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

index
The motor index
Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

AcceMax

The maximum acceleration

5.16.17 StepperGetAccelerationMin.vi

StepperGetAccelerationMin.vi

Get the minimum acceleration supported by a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**AcceMin**

The minimum acceleration
5.16.18 StepperGetCurrent.vi

StepperGetCurrent.vi

Get the current current draw for a motor

Device In

index

error in (no error)

Device Out

Current

error out

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Current

The current

5.16.19 StepperGetCurrentLimit.vi

StepperGetCurrentLimit.vi

Get the current limit for a motor

Device In

Device In

Device # Identification.

error in (no error)

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
describes the origin of the error or warning.

index
The motor index

Device Out
Same as Device In

error out
can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
describes the origin of the error or warning.

5.16.20 StepperGetCurrentMax.vi

StepperGetCurrentMax.vi

Get the maximum current limit
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

CurrentMax

The maximum current limit

5.16.21 StepperGetCurrentMin.vi

StepperGetCurrentMin.vi

Get the minimum current limit

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

done can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

done is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**CurrentMin**

The minimum current limit.

5.16.22 StepperGetCurrentPosition[vi](5.16.22 StepperGetCurrentPosition.vi)

Get the current position of a motor

Get the current position of a motor
Device # Identification.

**error in** (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
source

source describes the origin of the error or warning.

Position
The position

5.16.23 StepperGetCurrentPosition71.vi

StepperGetCurrentPosition71.vi
Get the current position of a motor. This function is for Labview version 7.1 only

Device In
Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

index
The motor index
Device Out

Same as Device In

table

table can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Position

The position

5.16.24 StepperGetEngaged.vi

StepperGetEngaged.vi

Get the engaged state of a motor. This is whether the motor is powered or not

Device In

Device # Identification.

table

table can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

EngagedState
The engaged state. Possible values are 0 for False, 1 for True and others for undefined

**Engaged?**

The engaged state (Boolean type). Possible values are True for Engaged and False for Not Engaged

### 5.16.25 StepperGetPositionMax.vi

**StepperGetPositionMax.vi**

Get the maximum position that a motor can go to

- **Device In**: Device # Identification.
- **error in (no error)**: `error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
- **status**: `status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**: `code` is the error or warning code.
- **source**: `source` describes the origin of the error or warning.
- **index**
The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**PositionMax**

The maximum position

5.16.26 StepperGetPositionMax71.vi

**StepperGetPositionMax71.vi**

Get the maximum position that a motor can go to. This function is for Labview version 7.1 only.

**Device In**

Device # Identification.

**error in (no error)**
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
PositionMax
Maximum position

5.16.27 StepperGetPositionMin.vi

StepperGetPositionMin.vi

Get the minimum position that a motor can go to.

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

index
The motor index

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PositionMin

The minimum position

5.16.28 StepperGetPositionMin71.vi

StepperGetPositionMin71.vi

Get the minimum position that a motor can go to. This function is for Labview version 7.1 only

Device In

Device In is used to identify the device.

error in (no error)
	error in (no error) can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PositionMin

The minimum position
5.16.29 StepperGetTargetPosition.vi

StepperGetTargetPosition.vi

Get the last set target position of a motor

Device In

Device In

Device Out

Device Out

error in (no error)

error in (no error)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index

Device Out

Device Out

Same as Device In

error out

error out can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Position

The position

5.16.30 StepperGetTargetPosition71.vi

StepperGetTargetPosition71.vi

Get the last set target position of a motor. This function is for Labview version 7.1 only

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

describes the origin of the error or warning.

index

The motor index

Device Out

Same as Device In

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

describes the origin of the error or warning.

Position

The position

5.16.31 StepperGetVelocity.vi

StepperGetVelocity.vi

Get the current velocity of a motor
Device In

Device # Identification.

`error in (no error)`

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

`status`

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

`code`

`code` is the error or warning code.

`source`

`source` describes the origin of the error or warning.

`index`

The motor index

Device Out

Same as Device In

`error out`

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

`status`

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or...
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Velocity

The velocity

5.16.32 StepperGetVelocityLimit.vi

StepperGetVelocityLimit.vi

Get the last set velocity limit for a motor

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or
warning.

index

The motor index

Device Out

Same as Device In

error out

text

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

VelocityLimit

The velocity limit

5.16.33 StepperGetVelocityMax.vi

StepperGetVelocityMax.vi

Get the maximum velocity that can be set for a motor
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
5.16.34 StepperGetVelocityMin.vi

StepperGetVelocityMin.vi

Get the minimum velocity that can be set for a motor

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index
Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

VelocityMin

The minimum velocity

5.16.35 StepperInputCount.vi

StepperInputCount.vi

Get the number of digital inputs supported by this board

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

The **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

er**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

The **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**count**

The digital input count
5.16.36 StepperInputState.vi

**StepperInputState.vi**

Get the state of a digital input

![Diagram](Image)

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### Input State (0,1)

The input state. Possible values are 0 for False, 1 for True and others for undefined.

**Input State**

The input state (Boolean type).

---

### 5.16.37 StepperMotorCount.vi

**StepperMotorCount.vi**

Get the number of motors supported by this controller.

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

count

The motor count

5.16.38 StepperOpen.vi

StepperOpen.vi
Serial Number

Serial Number. Specify -1 to open any.

milliseconds

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

error in (no error)

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device In

Device # identification. This function will create a new device identification if it's 0 or invalid

Serial Number Return

Serial Number of the opened phidget

Attached?

Returns TRUE is the device successfully attached, or FALSE otherwise.

error out

error out passes error or warning information out of a VI to be used by other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Same as Device In

5.16.39 StepperSetAcceleration.vi

StepperSetAcceleration.vi

Set the acceleration for a motor.

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
**code** is the error or warning code.

**source** describes the origin of the error or warning.

**acce_in**
The acceleration

**index**
The motor index

**Device Out**
Same as Device In

**error out**
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**
**code** is the error or warning code.

**source** describes the origin of the error or warning.

### 5.16.40 StepperSetCurrentLimit.vi

**StepperSetCurrentLimit.vi**

Set the current limit for a motor
Device In

Device # Identification.

error in (no error)

derror in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Current

The current limit

index

The motor index

Device Out

Same as Device In

error out

derror in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.16.41 StepperSetCurrentPosition.vi

StepperSetCurrentPosition.vi

Set the current position of a motor. It will not move the motor, just update the position value

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
warning.

**position**

The position

**index**

The motor index

**Device Out**

Same as Device In

**error out**

text can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

describes the origin of the error or warning.

5.16.42 StepperSetCurrentPosition71.vi

**StepperSetCurrentPosition71.vi**

Set the current position of a motor. It will not move the motor, just update the position value. This function is for Labview version 7.1 only.
Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The motor index

**position**

The position

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
5.16.43 StepperSetEngaged.vi

StepperSetEngaged.vi

Set the engaged state of a motor. This is whether the motor is powered or not.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The motor index
SetEngaged

Set the engage state

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.16.44 StepperSetTargetPosition.vi

StepperSetTargetPosition.vi

Set the target position of a motor

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should
be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

position

The position

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.16.45 StepperSetTargetPosition71.vi

StepperSetTargetPosition71.vi

Set the target position of a motor. This function is for Labview version 7.1 only.

**Device In**

Device # Identification.

**error in (no error)**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**index**

The motor index

**position**

The position

**Device Out**

Same as Device In

**error out**
5.16.46 StepperSetVelocityLimit.vi

StepperSetVelocityLimit.vi

Set the velocity limit for a motor

Device In
- Device # Identification.
- error in (no error)

Device Out
- Velocity Limit
- Device Out
- error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

velocity

The velocity limit

index

The motor index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.16.47 StepperStoppedState.vi

StepperStoppedState.vi

Get the stopped state of a motor. This is true when the motor is not moving and there are no outstanding commands
**Device In**

Device # Identification.

**error in (no error)**

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

`code` is the error or warning code.

**source**

`source` describes the origin of the error or warning.

**index**

The motor index

**Device Out**

Same as Device In

**error out**

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**StoppedState**

The stopped state. Possible values are 0 for False, 1 for True and others for undefined.

**Stopped?**

The stopped state (Boolean type). Possible values are True for Stopped and False for Not Stopped.

---

### 5.17 TemperatureSensor

#### 5.17.1 TempCreate.vi

**TempCreate.vi**

Create a Phidget Temperature Sensor handle

![Block diagram of TempCreate.vi](image)

**error in**

error in passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**
source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Device # identification

5.17.2 TempEventClose.vi

TempEventClose.vi

Close the Phidget Temperature Sensor event handle.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

5.17.3 TempEventCreate.vi

TempEventCreate.vi

Set up a temperature change event handle
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Device Out

Same as Device In

**error out**

**error out** passes error or warning information out of a VI to be used by other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** string describes the origin of the error or warning.

Event Registration Refnum Out
5.17.4 TempEventExe.vi

TempEventExe.vi

This is called when the temperature changes by more than the change trigger.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Event Registration Refnum In

Event # Identification

Value

The temperature...
index

The input index

Device Out

Same as Device In

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Event Registration Refnum Out

Same as the Event Registration Refnum In

Event?

Returns TRUE if the event has executed, or FALSE otherwise.

5.17.5 TempGetAmbient.vi

TempGetAmbient.vi

Get the ambient (board) temperature

Device In

Device # Identification.

error in (no error)
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Ambient**

The ambient (board) temperature
5.17.6 TempGetAmbientMax.vi

TempGetAmbientMax.vi

Get the maximum temperature that the ambient onboard temperature sensor can measure

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
5.17.7 TempGetAmbientMin.vi

TempGetAmbientMin.vi

Get the minimum temperature that the ambient onboard temperature sensor can measure.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Ambient Max

The maximum temperature
source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Ambient Min**

The minimum temperature

5.17.8 **TempGetPotential.vi**

**TempGetPotential.vi**

Get the currently sensed potential for a thermocouple input

**Device In**

Device # Identification.

**error in (no error)**
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
potential

The potential

5.17.9 TempGetPotentialMax.vi

TempGetPotentialMax.vi

Get the maximum potential that a thermocouple input can measure

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PotentialMax

The maximum potential

5.17.10 TempGetPotentialMin.vi

TempGetPotentialMin.vi

Get the minimum potential that a thermocouple input can measure

Device In

Device Out

error in (no error)

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

PotentialMin

The minimum potential
5.17.11 TempGetTemperature.vi

TempGetTemperature.vi

Get the temperature measured by a thermocouple input

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**temperature**

The temperature

5.17.12 TempGetTemperatureMax.vi

**TempGetTemperatureMax.vi**

Get the maximum temperature that can be measured by a thermocouple input. This depends on the type of thermocouple attached, as well as the ambient temperature

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**index**

The input index

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Temp Max**

The maximum temperature

### 5.17.13 TempGetTemperatureMin.vi

**TempGetTemperatureMin.vi**

Get the minimum temperature that can be measured by a thermocouple input. This depends on the type of thermocouple attached, as well as the ambient temperature
Device In

Device # Identification.

error in (no error)

to 

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
5.17.14 TempGetThermocoupleType.vi

TempGetThermocoupleType.vi

Get the type of thermocouple set to be at a thermocouple input. By default this is K-Type (1).

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Temp Min

The minimum temperature
warning.

index

The input index

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

ThermocoupleType

The thermocouple type. This is an enum. Please refer to Phidgets Constants -> ThermocoupleType

5.17.15 TempGetTrigger.vi

TempGetTrigger.vi

Get the change trigger for a thermocouple input
Device In

Device # Identification.

`error in` (no error)

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The input index

Device Out

Same as Device In

error out

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.
source
describes the origin of the error or warning.

trigger
The change trigger

5.17.16 TempInputCount.vi

TempInputCount.vi
Get the number of thermocouple inputs supported by this board

Device In
Device # Identification.

error in (no error)
can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

Device Out
Same as Device In
**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**count**

The thermocouple input count

---

### 5.17.17 TempOpen.vi

**TempOpen.vi**

Open a Phidget Temperature Sensor

![Diagram of TempOpen.vi]

**Serial Number**

Serial Number. Specify -1 to open any.

**milliseconds**

Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)

**error in**

**error out** passes error or warning information
out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* string describes the origin of the error or warning.

---

**Device In**

Device # Identification.

**Serial Number Return**

Serial Number of the opened phidget

**Attached?**

Returns TRUE is the device successfully attached, or FALSE otherwise.

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* string describes the origin of the error or warning.

---

**Device Out**

Same as Device In
5.17.18 TempSetThermocoupleType.vi

**TempSetThermocoupleType.vi**

Set the type of thermocouple plugged into a thermocouple input. By default this is K-Type

**Device In**

Device # Identification.

**error in (no error)**

`error in` can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

`status` is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

`code` is the error or warning code.

**source**

`source` describes the origin of the error or warning.

**index**

The input index

**ThermocoupleType**

The thermocouple type. This is an enum. Please refer to Phidgets Constants -> ThermocoupleType

**Device Out**

Same as Device In
error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.17.19 TempSetTrigger.vi

TempSetTrigger.vi

Set the change trigger for a thermocouple input

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

trigger

The change trigger

index

The input index

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.18 TextLCD

5.18.1 TextCreate.vi

TextCreate.vi
Create a Phidget TextLCD handle

error in

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

error out

error out passes error or warning information out of a VI to be used by other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source string describes the origin of the error or warning.

Device Out

Device # identification
5.18.2 TextGetBacklightState.vi

TextGetBacklightState.vi

Get the state of the backlight

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Backlight State

The backlight state. Possible values are 0 for False, 1 for True and others for undefined

Backlight?

The backlight state (Boolean type).

5.18.3 TextGetBrightness.vi

TextGetBrightness.vi

Get the brightness of the backlight. Not supported on all TextLCDs

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Brightness

The backlight brightness (0-255)

5.18.4 TextGetColumnCount.vi

TextGetColumnCount.vi

Get the number of columns per supported by this display
Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
**5.18.5 TextGetContrast.vi**

**TextGetContrast.vi**

Get the last set contrast value

- **Device In**
  - Device # Identification.
  - **error in (no error)**

- **Device Out**
  - **Contrast**
  - **error out**

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**
Same as Device In

error out

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Contrast

The contrast (0-255)

5.18.6 TextGetCursorBlinkState.vi

TextGetCursorBlinkState.vi

Get the cursor blink state

Device In

Device Out

Device In

(error in (no error)

Device # Identification.

(error in (no error)

can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

**CursorBlink State**

The cursor blink state. Possible values are 0 for False, 1 for True and others for undefined
5.18.7 TextGetCursorState.vi

TextGetCursorState.vi

Get the cursor visible state

Device In
Device # Identification.

error in (no error)
error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code
code is the error or warning code.

source
source describes the origin of the error or warning.

Device Out
Same as Device In

error out
error in can accept error information wired from VIs previously called. Use this
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Cursor State**

The state of the cursor. Possible values are 0 for False, 1 for True and others for undefined.

**Cursor?**

The state of the cursor (Boolean type)

### 5.18.8 TextGetRowCount.vi

**TextGetRowCount.vi**

Get the number of rows supported by this display

**Device In**

Device # Identification.

**error in (no error)**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

count

The row count

5.18.9 TextGetScreenCount.vi

TextGetScreenCount.vi
Gets the number of screens supported by the TextLCD

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
code

code is the error or warning code.

source

source describes the origin of the error or warning.

ScreenCount

The number of screens

5.18.10 TextGetScreenIndex.vi

TextGetScreenIndex.vi

Gets the screen index

Device In

Device In

error in (no error)

device identification.

error in

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.18.11 TextGetScreenSize.vi

TextGetScreenSize.vi

Gets the screen size for the active TextLCD display.

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

ScreenIndex

The screen index
information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Device Out**

Same as Device In

**error out**

*error in* can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

*status* is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

*code* is the error or warning code.

**source**

*source* describes the origin of the error or warning.

**Screen Size**

The screen size
5.18.12 TextOpen.vi

TextOpen.vi

Open a Phidget TextLCD

- **Serial Number**: Serial Number. Specify -1 to open any.
- **milliseconds**: Time to wait for the attachment. Specify 0 to wait forever. (Default is 5000)
- **error in (no error)**: error out passes error or warning information out of a VI to be used by other VIs.
- **status**: status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- **code**: code is the error or warning code.
- **source**: source string describes the origin of the error or warning.

- **Device In**: Device # identification. This function will create a new device identification if it's 0 or invalid
- **Serial Number Return**: Serial Number of the opened phidget
- **Attached?**: Returns TRUE is the device successfully
attached, or FALSE otherwise.

**error out**

error out passes error or warning information out of a VI to be used by other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source string describes the origin of the error or warning.

**Device Out**

Same as Device In

### 5.18.13 TextReset.vi

**TextReset.vi**

Re-initializes the LCD Display, clearing it, etc.

**Device In**

Device # Identification.

e**r** **o** **r** **o** **u** **n** (no error)

**Device Out**

**status**

Device In (no error)

e**r** **o** **r** **o** **u** **n** (no error)

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**source**

source string describes the origin of the error or warning.
status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.18.14 TextSetBacklightState.vi

TextSetBacklightState.vi

Set the state of the backlight

© 2014 Phidgets Inc.
Device In

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

Turn Backlight

Set the backlight state

Device Out

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.
5.18.15 TextSetBrightness.vi

TextSetBrightness.vi

Set the brightness of the backlight. Not supported on all TextLCDs

Device In

Device # Identification.

error in (no error)

can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

Brightness

The backlight brightness (0-255).

Device Out

Same as Device In
**5.18.16 TextSetCharacter.vi**

**TextSetCharacter.vi**

Set a custom character. See the product manual for more information.

---

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

index

The custom character index (8-15)

var1

The first part of the custom character

var2

The second part of the custom character

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.
5.18.17 TextSetContrast.vi

TextSetContrast.vi

Set the last set contrast value

---

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

---

**Contrast**

The contrast (0-255).

---

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.
status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

5.18.18 TextSetCursorBlinkState.vi

TextSetCursorBlinkState.vi

Set the cursor blink state

Device In

Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source
source describes the origin of the error or warning.

**Cursor Blink**

Set the cursor blink state

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.

---

5.18.19 TextSetCursorState.vi

**TextSetCursorState.vi**

Set the cursor visible state

**Device In**

Device # Identification.

**error in (no error)**
**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**Turn Cursor ON**

Set the state of the cursor

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.18.20 TextSetDisplayChar.vi

TextSetDisplayChar.vi

Set a single character on the display

Device In
Device # Identification.

error in (no error)

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

row

The row index

column

The column index

character

The character to display

Device Out
Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

### 5.18.21 TextSetDisplayString.vi

**TextSetDisplayString.vi**

Set a row on the display

**Device In**

Device # Identification.

**error in (no error)**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**
**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.

**row**

The row index

**string**

The string to display. Make sure this is not longer then `TextGetColumnCount`

**Device Out**

Same as Device In

**error out**

**error in** can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

**status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

**code** is the error or warning code.

**source**

**source** describes the origin of the error or warning.
5.18.22 TextSetScreenIndex.vi

TextSetScreenIndex.vi

Choose the screen to modify

Device In

ScreenIndex

Device Out

error in (no error)

Device In

Device # Identification.

error in (no error)

Device In can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

status

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code

code is the error or warning code.

source

source describes the origin of the error or warning.

ScreenIndex

The index of the screen being selected

Device Out

Same as Device In

error out

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other
5.18.23 TextSetScreenSize.vi

TextSetScreenSize.vi

Set the size of the screen

Device In

Device Out

error in (no error)

Set Size

status

code

source

Device In

Device # Identification.

texterror in (no error)
text

text

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

code is the error or warning code.

source describes the origin of the error or warning.
source describes the origin of the error or warning.

**Screen Size**

The screen size

**Device Out**

Same as Device In

**error out**

error in can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

**status**

status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.

**code**

code is the error or warning code.

**source**

source describes the origin of the error or warning.
6 Phidgets Constants

This section describes each of the Phidgets constants used by different Phidgets.

6.1 CodeInfo

The PhidgetIR CodeInfo structure contains all information needed to transmit a code, apart from the actual code data. Some values can be set to null to select defaults. See the product manual for more information.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Item</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitCount</td>
<td>encoding</td>
<td>Number of bits in the code. This is an enum. Please refer to Phigets Constant -&gt; IREncoding</td>
</tr>
<tr>
<td>length</td>
<td>gap</td>
<td>Gap time in us.</td>
</tr>
<tr>
<td>gap</td>
<td>header [2]</td>
<td>Trail time in us - can be 0 for none.</td>
</tr>
<tr>
<td>trail</td>
<td>one [2]</td>
<td>Header pulse and space - can be 0 for none.</td>
</tr>
<tr>
<td>zero [2]</td>
<td>repeat [26]</td>
<td>Pulse and Space times to represent a '0' bit, in us.</td>
</tr>
<tr>
<td>repeat [26]</td>
<td>min_repeat</td>
<td>A series or pulse and space times to represent the repeat code. Start and end with pulses and null terminate. Set to 0 for none.</td>
</tr>
<tr>
<td>min_repeat</td>
<td>toggle_mask [16]</td>
<td>Minimum number of times to repeat a code on transmit.</td>
</tr>
<tr>
<td>toggle_mask [16]</td>
<td>carrierFrequency</td>
<td>Carrier frequency in Hz - defaults to 38kHz.</td>
</tr>
<tr>
<td>carrierFrequency</td>
<td>dutyCycle</td>
<td>Duty Cycle in Hz - defaults to 38kHz.</td>
</tr>
</tbody>
</table>

6.2 Frequency Filter Mode

An enum value with the following definition:

<table>
<thead>
<tr>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZERO_CROSSING = 1</td>
<td>Phidget responds to Zero Crossing</td>
</tr>
<tr>
<td>LOGIC_LEVEL = 2</td>
<td>Phidget responds to logic levels</td>
</tr>
</tbody>
</table>

6.3 IREncoding

The PhidgetIR supports these data encodings:
<table>
<thead>
<tr>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIDGET_IR_ENCODING_UNKNOWN = 1,</td>
<td>Unknown - the default value</td>
</tr>
<tr>
<td>PHIDGET_IR_ENCODING_SPACE = 2,</td>
<td>Space encoding, or Pulse Distance Modulation</td>
</tr>
<tr>
<td>PHIDGET_IR_ENCODING_PULSE = 3,</td>
<td>Pulse encoding, or Pulse Width Modulation</td>
</tr>
<tr>
<td>PHIDGET_IR_ENCODING_BIPHASE = 4,</td>
<td>Bi-Phase, or Manchester encoding</td>
</tr>
<tr>
<td>PHIDGET_IR_ENCODING_RC5 = 5,</td>
<td>RC5 - a type of Bi-Phase encoding</td>
</tr>
<tr>
<td>PHIDGET_IR_ENCODING_RC6 = 6,</td>
<td>RC6 - a type of Bi-Phase encoding</td>
</tr>
<tr>
<td>Others</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

### 6.4 IRLength

The PhidgetIR supports these encoding lengths:

<table>
<thead>
<tr>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIDGET_IR_LENGTH_UNKNOWN = 1,</td>
<td>Unknown - the default value</td>
</tr>
<tr>
<td>PHIDGET_IR_LENGTH_CONSTANT = 2,</td>
<td>Constant - the bitstream + gap length is constant</td>
</tr>
<tr>
<td>PHIDGET_IR_LENGTH_VARIABLE = 3,</td>
<td>Variable - the bitstream has a variable length with a constant gap</td>
</tr>
<tr>
<td>Others</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

### 6.5 ServoType

An enum value with the following definition:

<table>
<thead>
<tr>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIDGET_SERVO_DEFAULT = 1,</td>
<td>Default - This is what the servo API been historically used, originally based on the Futaba FP-S148</td>
</tr>
<tr>
<td>PHIDGET_SERVO_RAW_us_MODE = 2,</td>
<td>Raw us mode - all position, velocity, acceleration functions are specified in microseconds rather then degrees</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS322HD = 3,</td>
<td>HiTec HS-322HD Standard Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS5245MG = 4,</td>
<td>HiTec HS-5245MG Digital Mini Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_805BB = 5,</td>
<td>HiTec HS-805BB Mega Quarter Scale Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS422 = 6,</td>
<td>HiTec HS-422 Standard Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_TOWERPRO_MG90 = 7,</td>
<td>Tower Pro MG90 Micro Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HSR1425CR = 8,</td>
<td>HiTec HSR-1425CR Continuous Rotation Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS785HB = 9,</td>
<td>HiTec HS-785HB Sail Winch Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS485HB = 10,</td>
<td>HiTec HS-485HB Deluxe Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_HS645MG = 11,</td>
<td>HiTec HS-645MG Ultra Torque Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_HITEC_815BB = 12,</td>
<td>HiTec HS-815BB Mega Sail Servo</td>
</tr>
<tr>
<td>PHIDGET_SERVO_FIRGELLI_L12_30_50_06_R = 13,</td>
<td>Firgelli L12 Linear Actuator 30mm 50:1</td>
</tr>
<tr>
<td>PHIDGET_SERVO_FIRGELLI_L12_50_10_06_R = 14,</td>
<td>Firgelli L12 Linear Actuator 50mm 100:1</td>
</tr>
<tr>
<td>PHIDGET_SERVO_FIRGELLI_L12_50_21_06_R = 15,</td>
<td>Firgelli L12 Linear Actuator 50mm 210:1</td>
</tr>
<tr>
<td>PHIDGET_SERVO_FIRGELLI_L12_100_5_06_R = 16,</td>
<td>Firgelli L12 Linear Actuator 100mm 50:1</td>
</tr>
<tr>
<td>PHIDGET_SERVO_FIRGELLI_L12_100_100_06_R = 17,</td>
<td>Firgelli L12 Linear Actuator 100mm 100:1</td>
</tr>
<tr>
<td>PHIDGET_SERVO_USER_DEFINED = others</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

### 6.6 ThermocoupleType

An enum value with the following definition:

<table>
<thead>
<tr>
<th>Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHIDGET_TEMPERATURE_SENSOR_K_TYPE = 1,</td>
<td>K-Type thermocouple</td>
</tr>
<tr>
<td>PHIDGET_TEMPERATURE_SENSOR_J_TYPE = 2,</td>
<td>J-Type thermocouple</td>
</tr>
<tr>
<td>PHIDGET_TEMPERATURE_SENSOR_E_TYPE = 3,</td>
<td>E-Type thermocouple</td>
</tr>
<tr>
<td>PHIDGET_TEMPERATURE_SENSOR_T_TYPE = 4,</td>
<td>T-Type thermocouple</td>
</tr>
<tr>
<td>TYPE = 4,</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>PHIDGET_SERVO_USER_DEFINED = others</td>
<td>Undefined</td>
</tr>
</tbody>
</table>