

App Inventor for FIRST® Tech Challenge

[FtcAccelerationSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcAnalogInput](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcAnalogOutput](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcColorSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcCompassSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcDcMotor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcDcMotorController](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcDeviceInterfaceModule](#)

[Events \(when-do blocks\)](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcDigitalChannel](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcElapsedTime](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcGamepad](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcGyroSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcI2cDevice](#)

[Events \(when-do blocks\)](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcI2cDeviceReader](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcI2cSeekerSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcLED](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcLegacyModule](#)

[Events \(when-do blocks\)](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcLightSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcLinearOpMode](#)

[Events \(when-do blocks\)](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcOpMode](#)

[Events \(when-do blocks\)](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcOpticalDistanceSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcPwmOutput](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcRobotController](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcServo](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcServoController](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcTouchSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcTouchSensorMultiplexer](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcUltrasonicSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

[FtcVoltageSensor](#)

[Methods \(call blocks\)](#)

[Properties](#)

FtcAccelerationSensor

A component for an acceleration sensor of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Status

The Status. (read only)

Version

The version. (read only)

XAccel

The X Acceleration, in g's. (read only)

YAccel

The Y Acceleration, in g's. (read only)

ZAccel

The Z Acceleration, in g's. (read only)

FtcAnalogInput

A component for an analog input device of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Value

The current ADC results from the A0-A7 channel input pins. (read only)

Version

The version. (read only)

FtcAnalogOutput

A component for an analog output device of an FTC robot.

Methods (call blocks)

SetAnalogOutputFrequency(number frequency)

Sets the channel output frequency in the range 1-5,000 Hz in mode 1, 2 or 3.

SetAnalogOutputMode(number mode)

Sets the channel operating mode.

Mode 0: Voltage output. Range: -4V - 4V.

Mode 1: Sine wave output. Range: 0 - 8V.

Mode 2: Square wave output. Range: 0 - 8V.

Mode 3: Triangle wave output. Range: 0 - 8V.

SetAnalogOutputVoltage(number voltage)

Sets the channel output voltage.

If mode == 0: takes input from -1023-1023, output in the range -4 to +4 volts.

If mode == 1, 2, or 3: takes input from 0-1023, output in the range 0 to 8 volts.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

FtcColorSensor

A component for a color sensor of an FTC robot.

Methods (call blocks)

HSV ConvertColorToHSV(number color)

Convert an integer ARGB (alpha, red, green, blue) color to an HSV (hue, saturation, value).

*Prior to the 11/05/15 release, this block was named **ColorToHSV**.*

number ConvertHSVToColor(HSV hsv)

Convert an HSV (hue, saturation, value) to an integer ARGB (alpha, red, green, blue) color.
*Prior to the 11/05/15 release, this block was named **HSVToColor**.*

number ConvertHSVToColorWithAlpha(number alpha, HSV hsv)

Convert a specified alpha value and an HSV (hue, saturation, value) to an integer ARGB (alpha, red, green, blue) color.

*Prior to the 11/05/15 release, this block was named **HSVToColorWithAlpha**.*

HSV ConvertRGBToHSV(number red, number green, number blue)

Convert red, green, and blue values to an HSV (hue, saturation, value).

*Prior to the 11/05/15 release, this block was named **RGBToHSV**.*

number CreateARGB(number alpha, number red, number green, number blue)

Create an integer ARGB (alpha, red, green, blue) color from alpha, red, green, blue values.

*Prior to the 11/05/15 release, this block was named **ARGB**.*

HSV CreateHSV(number hue, number saturation, number value)

Create an HSV (hue, saturation, value).

number CreateRGB(number red, number green, number blue)

Return an integer ARGB (alpha, red, green, blue) color from red, green, blue values.

*Prior to the 11/05/15 release, this block was named **RGB**.*

EnableLed(boolean enable)

Enable the LED light.

number ExtractAlpha(number color)

Extract the alpha component of an integer ARGB (alpha, red, green, blue) color.

*Prior to the 11/05/15 release, this block was named **Alpha**.*

number ExtractBlue(number color)

Extract the blue component of an integer ARGB (alpha, red, green, blue) color.

*Prior to the 11/05/15 release, this block was named **Blue**.*

number ExtractGreen(number color)

Extract the green component of an integer ARGB (alpha, red, green, blue) color.

*Prior to the 11/05/15 release, this block was named **Green**.*

number ExtractHue(HSV hsv)

Extract the hue from the given HSV (hue, saturation, value).

*Prior to the 11/05/15 release, this block was named **Hue**.*

number ExtractRed(number color)

Extract the red component of an integer ARGB (alpha, red, green, blue) color.
*Prior to the 11/05/15 release, this block was named **Red**.*

number ExtractSaturation(HSV hsv)

Extract the saturation from the given HSV (hue, saturation, value).
*Prior to the 11/05/15 release, this block was named **Saturation**.*

number ExtractValue(HSV hsv)

Extract the value from the given HSV (hue, saturation, value).
*Prior to the 11/05/15 release, this block was named **Value**.*

number ParseColor(text colorText)

Parse the color string, and return the corresponding ARGB (alpha, red, green, blue) color. Supported formats are: #RRGGBB #AARRGGBB or one of the following names: 'red', 'blue', 'green', 'black', 'white', 'gray', 'cyan', 'magenta', 'yellow', 'lightgray', 'darkgray', 'grey', 'lightgrey', 'darkgrey', 'aqua', 'fuchsia', 'lime', 'maroon', 'navy', 'olive', 'purple', 'silver', 'teal'.

text ToString()

Return text representing the state of the hardware device.

Properties

ARGB

The color detected by the sensor as an integer ARGB (alpha, red, green, blue) color. (read only)

Alpha

The amount of light detected by the sensor as an integer. (read only)

Blue

The Blue values detected by the sensor as an integer. (read only)

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Green

The Green values detected by the sensor as an integer. (read only)

I2cAddress

The I2C address of the color sensor. Not all color sensors support this feature.

This block was added in the 11/05/15 release.

MAX_NEW_I2C_ADDRESS

The constant for MAX_NEW_I2C_ADDRESS. (read only)

This block was added in the 11/05/15 release.

MIN_NEW_I2C_ADDRESS

The constant for MIN_NEW_I2C_ADDRESS. (read only)

This block was added in the 11/05/15 release.

Red

The Red values detected by the sensor as an integer. (read only)

Version

The version. (read only)

FtcCompassSensor

A component for a compass sensor of an FTC robot.

Methods (call blocks)

SetMode(CompassMode mode)

Change to calibration or measurement mode.

Valid values are CompassMode_CALIBRATION_MODE or CompassMode_MEASUREMENT_MODE.

text ToString()

Return text representing the state of the hardware device.

Properties

CalibrationFailed

Whether calibration failed. (read only)

CompassMode_CALIBRATION_MODE

The constant for CompassMode_CALIBRATION_MODE. (read only)

CompassMode_MEASUREMENT_MODE

The constant for CompassMode_MEASUREMENT_MODE. (read only)

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Direction

The Direction, in degrees. (read only)

Status

The Status. (read only)

Version

The version. (read only)

FtcDcMotor

A component for a DC motor of an FTC robot.

Methods (call blocks)

boolean GetPowerFloat()

Is motor power set to float?

boolean IsBusy()

Whether the motor is busy.

SetPowerFloat()

Allow the motor to float.

text ToString()

Return text representing the state of the hardware device.

Properties

ChannelModeMode

The run mode.

Valid values are RunMode_RUN_USING_ENCODERS, RunMode_RUN_WITHOUT_ENCODERS, RunMode_RUN_TO_POSITION, or RunMode_RESET_ENCODERS

ConnectionInfo

The connection information. (read only)

CurrentPosition

The current encoder value.

If this motor has been set to REVERSE, the value will be multiplied by -1. (read only)

Device

The manufacturer and name of the device. (read only)

Direction

Whether this motor should spin forward or reverse.

Valid values are Direction_FORWARD or Direction_REVERSE.

Direction_FORWARD

The constant for Direction_FORWARD. (read only)

Direction_REVERSE

The constant for Direction_REVERSE. (read only)

PortNumber

The port number. (read only)

Power

The current motor power, between -1 and 1.

RunMode_RESET_ENCODERS

The constant for RunMode_RESET_ENCODERS. (read only)

RunMode_RUN_TO_POSITION

The constant for RunMode_RUN_TO_POSITION. (read only)

RunMode_RUN_USING_ENCODERS

The constant for RunMode_RUN_USING_ENCODERS. (read only)

RunMode_RUN_WITHOUT_ENCODERS

The constant for RunMode_RUN_WITHOUT_ENCODERS. (read only)

TargetPosition

The motor target position.

If this motor has been set to REVERSE, the value will be multiplied by -1.

Version

The version. (read only)

FtcDcMotorController

A component for a DC motor controller of an FTC robot.

Methods (call blocks)

number GetDifferentialControlLoopCoefficientD(number motor)

Get the differential control loop coefficient d, if supported by the controller.

This block was added in the 11/05/15 release.

number GetDifferentialControlLoopCoefficientI(number motor)

Get the differential control loop coefficient i, if supported by the controller.

This block was added in the 11/05/15 release.

number GetDifferentialControlLoopCoefficientP(number motor)

Get the differential control loop coefficient p, if supported by the controller.

This block was added in the 11/05/15 release.

number GetGearRatio(number motor)

Get the gear ratio, if supported by the controller.

This block was added in the 11/05/15 release.

RunMode GetMotorChannelMode(number motor)

Get the current channel mode.

Valid values are RunMode_RUN_USING_ENCODERS, RunMode_RUN_WITHOUT_ENCODERS, RunMode_RUN_TO_POSITION, or RunMode_RESET_ENCODERS.

number GetMotorCurrentPosition(number motor)

Get the current motor position.

number GetMotorPower(number motor)

Get the current motor power.

boolean GetMotorPowerFloat(number motor)

Allow motor to float.

number GetMotorTargetPosition(number motor)

Get the current motor target position.

boolean IsBusy(number motor)

Is the motor busy?

SetDifferentialControlLoopCoefficients(number motor, number p, number i, number d)

Set the differential control loop coefficients, if supported by the controller.

This block was added in the 11/05/15 release.

SetGearRatio(number motor, number ratio)

Set the gear ratio (from -1.0 to 1.0), if supported by the controller.

This block was added in the 11/05/15 release.

SetMotorChannelMode(number motor, RunMode mode)

Set the current channel mode.

Valid values are RunMode_RUN_USING_ENCODERS, RunMode_RUN_WITHOUT_ENCODERS, RunMode_RUN_TO_POSITION, or RunMode_RESET_ENCODERS.

SetMotorPower(number motor, number power)

Set the current motor power (from -1.0 to 1.0).

SetMotorPowerFloat(number motor)

Allow motor to float.

SetMotorPowerForGroup(list listOfFtcDcMotors, number power)

Set the power for a group of motors, if supported by the controller..

This block was added in the 11/05/15 release.

SetMotorTargetPosition(number motor, number position)

Set the motor target position.

text ToString()

Return text representing the state of the hardware device.

Properties

BatteryVoltage

Get the battery voltage, if supported by the controller. (read only)

This block was added in the 11/05/15 release.

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

DeviceMode_READ_ONLY

The constant for DeviceMode_READ_ONLY. (read only)

DeviceMode_WRITE_ONLY

The constant for DeviceMode_WRITE_ONLY. (read only)

MotorControllerDeviceMode

Set the device into read-only or write-only mode.

Valid values are DeviceMode_READ_ONLY or DeviceMode_WRITE_ONLY.

RunMode_RESET_ENCODERS

The constant for RunMode_RESET_ENCODERS. (read only)

RunMode_RUN_TO_POSITION

The constant for RunMode_RUN_TO_POSITION. (read only)

RunMode_RUN_USING_ENCODERS

The constant for RunMode_RUN_USING_ENCODERS. (read only)

RunMode_RUN_WITHOUT_ENCODERS

The constant for RunMode_RUN_WITHOUT_ENCODERS. (read only)

Version

The version. (read only)

FtcDeviceInterfaceModule

A component for a device interface module of an FTC robot.

Events (when-do blocks)

I2cPortIsReady(number port)

This event is triggered when an I2C port is ready, after the latest data has been read from the I2C Controller.

This event is only enabled if EnableI2cReadMode or EnableI2cWriteMode is used.

Methods (call blocks)

CopyBufferIntoWriteBuffer(number physicalPort, ByteArray byteArray)

Copy a byte array into the buffer that is set to be written out to the device.

EnableI2cReadMode(number physicalPort, number i2cAddress, number memAddress, number length)

Enable read mode for a particular I2C device and enable the I2cPortIsReady event for the given port.

EnableI2cWriteMode(number physicalPort, number i2cAddress, number memAddress, number length)

Enable write mode for a particular I2C device and enable the I2cPortIsReady event for the given port.

number GetAnalogInputValue(number channel)

Return the current ADC results from the A0-A7 channel input pins.

ByteArray GetCopyOfReadBuffer(number physicalPort)

Get a copy of the most recent data read in from the device.

ByteArray GetCopyOfWriteBuffer(number physicalPort)

Get a copy of the data that is set to be written out to the device.

Mode GetDigitalChannelMode(number channel)

Get the mode of a digital channel.

Valid values are Mode_INPUT or Mode_OUTPUT.

boolean GetDigitalChannelState(number channel)

Get the state of a digital channel.

If it's in OUTPUT mode, this will return the output bit.

If the channel is in INPUT mode, this will return the input bit.

number GetDigitalIOControlByte()

Get the digital IO control byte.

number GetDigitalInputStateByte()

A byte containing the current logic levels present in the D7-D0 channel pins. If a particular pin is in output mode, the current output state will be reported.

number GetDigitalOutputStateByte()

The D7-D0 output set field is a byte containing the required I/O output of the D7-D0 channel pins.

If the corresponding Dy-D0 I/O control field bit is set to one, the channel pin will be in output mode and will reflect the value of the corresponding D7-D0 output set field bit.

boolean GetLEDState(number channel)

Indicates whether the LED on the given channel is on or not.

number GetPulseWidthOutputTime(number port)

Gets the pulse width for the channel output in units of 1 microsecond.

number GetPulseWidthPeriod(number port)

Gets the pulse repetition period for the channel output in units of 1 microsecond.

boolean IsI2cPortActionFlagSet(number port)

Get the port action flag; this flag is set if the particular port is busy.

boolean IsI2cPortInReadMode(number port)

Is the port in read mode?

boolean IsI2cPortInWriteMode(number port)

Is the port in write mode?

boolean IsI2cPortReady(number port)

Determine if a physical port is ready.

ReadI2cCacheFromController(number port)

Read the local cache in from the I2C Controller.

NOTE: unless this method is called the internal cache isn't updated.

SetAnalogOutputFrequency(number port, number frequency)

Sets the channel output frequency in the range 1-5,000 Hz in mode 1, 2 or 3.

SetAnalogOutputMode(number port, number mode)

Sets the channel operating mode.

Mode 0: Voltage output. Range: -4V - 4V.

Mode 1: Sine wave output. Range: 0 - 8V.

Mode 2: Square wave output. Range: 0 - 8V.

Mode 3: Triangle wave output. Range: 0 - 8V.

SetAnalogOutputVoltage(number port, number voltage)

Sets the channel output voltage.

If mode == 0: takes input from -1023-1023, output in the range -4 to +4 volts.

If mode == 1, 2, or 3: takes input from 0-1023, output in the range 0 to 8 volts.

SetDigitalChannelMode(number channel, Mode mode)

Set the mode of a digital channel.

Valid values are Mode_INPUT or Mode_OUTPUT.

SetDigitalChannelState(number channel, boolean state)

Set the state of a digital channel.

The behavior of this method is undefined for digital channels in INPUT mode.

SetDigitalIOControlByte(number input)

If a particular bit is set to one, the corresponding channel pin will be in output mode; else it will be in input mode.

SetDigitalOutputByte(number input)

If a particular control field bit is set to one, the channel pin will be in output mode and will reflect the value of the corresponding field bit.

SetI2cPortActionFlag(number port)

Set the port action flag; this flag tells the controller to send the current data in its buffer to the I2C device.

SetLED(number channel, boolean state)

Turn on or off a particular LED.

SetPulseWidthOutputTime(number port, number time)

Set the pulse width output time for this channel.

Typically set to a value between 750 and 2,250 to control a servo.

SetPulseWidthPeriod(number port, number period)

Set the pulse width output period.

Typically set to 20,000 to control servo.

text ToString()

Return text representing the state of the hardware device.

WriteI2cCacheToController(number port)

Write the local cache to the I2C Controller.

NOTE: unless this method is called the internal cache isn't updated.

WriteI2cPortFlagOnlyToController(number port)

Write just the port action flag in the local cache to the I2C controller.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Mode_INPUT

The constant for Mode_INPUT. (read only)

Mode_OUTPUT

The constant for Mode_OUTPUT. (read only)

SerialNumber

Get the USB serial number of this device. (read only)

Version

The version. (read only)

FtcDigitalChannel

A component for a single digital channel of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Mode

The mode of the digital channel.

Valid values are Mode_INPUT or Mode_OUTPUT.

Mode_INPUT

The constant for Mode_INPUT. (read only)

Mode_OUTPUT

The constant for Mode_OUTPUT. (read only)

State

The state of the digital channel.

If it's in OUTPUT mode, this will return the output bit.

If the channel is in INPUT mode, this will return the input bit.

Version

The version. (read only)

FtcElapsedTime

A component for measuring elapsed time, with nanosecond accuracy, for an FTC robot.

This component was added in the 11/05/15 release.

Methods (call blocks)

Log(text label)

Log a message stating how long the timer has been running.

Reset()

Reset the start time to now.

text ToString()

Return text stating the number of seconds that have passed.

Properties

StartTime

Get the relative start time. (read only)

Time

Get the number of seconds since the start time, with nanosecond accuracy. (read only)

FtcGamepad

A component for a gamepad of an FTC robot.

Methods (call blocks)

SetJoystickDeadzone(number joystickDeadzone)

Set the joystick deadzone. Must be between 0 and 1.

text ToString()

Return text representing the state of the gamepad.

Properties

A

The value of the A button: true if pressed, false otherwise. (read only)

AtRest

Are all analog sticks and triggers in their rest position? (read only)

B

The value of the B button: true if pressed, false otherwise. (read only)

Back

The value of the Back button: true if pressed, false otherwise. (read only)

DpadDown

The dpad down value: true if pressed, false otherwise. (read only)

DpadLeft

The dpad left value: true if pressed, false otherwise. (read only)

DpadRight

The dpad right value: true if pressed, false otherwise. (read only)

DpadUp

The dpad up value: true if pressed, false otherwise. (read only)

Guide

The value of the Guide button: true if pressed, false otherwise. The Guide button is often the large button in the middle of the controller. (read only)

LeftBumper

The left bumper value: true if pressed, false otherwise. (read only)

LeftStickButton

The value of the left stick button: true if pressed, false otherwise. (read only)

LeftStickX

The left analog stick horizontal axis value, as a numeric value between -1.0 and +1.0. (read only)

LeftStickY

The left analog stick vertical axis value, as a numeric value between -1.0 and +1.0. (read only)

LeftTrigger

The left trigger value, as a numeric value between 0.0 and +1.0. (read only)

RightBumper

The right bumper value: true if pressed, false otherwise. (read only)

RightStickButton

The value of the right stick button: true if pressed, false otherwise. (read only)

RightStickX

The right analog stick horizontal axis value, as a numeric value between -1.0 and +1.0. (read only)

RightStickY

The right analog stick vertical axis value, as a numeric value between -1.0 and +1.0. (read only)

RightTrigger

The right trigger value, as a numeric value between 0.0 and +1.0. (read only)

Start

The value of the Start button: true if pressed, false otherwise. (read only)

Status

The status of the gamepad. (read only)

X

The value of the X button: true if pressed, false otherwise. (read only)

Y

The value of the Y button: true if pressed, false otherwise. (read only)

FtcGyroSensor

A component for a gyro sensor of an FTC robot.

Methods (call blocks)

Calibrate()

Calibrate the gyro. Not all gyro sensors support this feature. For the Modern Robotics device this will reset the Z axis heading.

This block was added in the 11/05/15 release.

boolean IsCalibrating()

Is the gyro performing a calibration operation? Not all gyro sensors support this feature.

This block was added in the 11/05/15 release.

ResetZAxisIntegrator()

Set the integrated Z axis to zero. Not all gyro sensors support this feature.

This block was added in the 11/05/15 release.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Heading

The integrated Z axis as a cartesian or cardinal heading, as a numeric value between 0 and 360. Not all gyro sensors support this feature. (read only)

This block was added in the 11/05/15 release.

HeadingMode_CARDINAL

The constant for HeadingMode_CARDINAL. (read only)

This block was added in the 11/05/15 release.

HeadingMode_CARTESIAN

The constant for HeadingMode_CARTESIAN. (read only)

This block was added in the 11/05/15 release.

HeadingMode

The heading mode. Valid values are HeadingMode_CARTESIAN or HeadingMode_CARDINAL. Not all gyro sensors support this feature.

This block was added in the 11/05/15 release.

RawX

The gyro sensor's raw X value. Not all gyro sensors support this feature. (read only)

This block was added in the 11/05/15 release.

RawY

The gyro sensor's raw Y value. Not all gyro sensors support this feature. (read only)

This block was added in the 11/05/15 release.

RawZ

The gyro sensor's raw Z value. Not all gyro sensors support this feature. (read only)

This block was added in the 11/05/15 release.

Rotation

The Rotation. (read only)

Status

The Status. (read only)

Version

The version. (read only)

Ftcl2cDevice

A component for an I2C device of an FTC robot.

Events (when-do blocks)

I2cPortIsReady()

This event is triggered when the I2C port is ready. This event is only enabled if EnableI2cReadMode or EnableI2cWriteMode is used.

Methods (call blocks)

CopyBufferIntoWriteBuffer(ByteArray byteArray)

Copy a byte array into the buffer that is set to be written out to the device.

EnableI2cReadMode(number i2cAddress, number memAddress, number length)

Enable read mode for the I2C device and enable the I2cPortIsReady event.

EnableI2cWriteMode(number i2cAddress, number memAddress, number length)

Enable write mode for the I2C device and enable the I2cPortIsReady event.

ByteArray GetCopyOfReadBuffer()

Get a copy of the most recent data read in from the device.

ByteArray GetCopyOfWriteBuffer()

Get a copy of the data that is set to be written out to the device.

boolean IsI2cPortActionFlagSet()

Check whether or not the action flag is set for this I2C port.

boolean IsI2cPortInReadMode()

Query whether or not the port is in Read mode.

boolean IsI2cPortInWriteMode()

Query whether or not this port is in write mode.

boolean IsI2cPortReady()

Query whether or not this I2c port is ready.

ReadI2cCacheFromController()

Trigger a read of the I2C cache.

SetI2cPortActionFlag()

Set the action flag for this I2C port.

text ToString()

Return text representing the state of the hardware device.

WriteI2cCacheToController()

Trigger a write of the I2C cache.

WriteI2cPortFlagOnlyToController()

Write only the action flag.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

Ftcl2cDeviceReader

A component for an I2C device reader of an FTC robot.

Methods (call blocks)

ByteArray GetReadBuffer()

Get a copy of the most recent data read in from the I2C device.

Initialize(number i2cAddress, number memAddress, number length)

Initialize this I2C device reader

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

FtclrSeekerSensor

A component for an IR seeker sensor of an FTC robot.

Methods (call blocks)

number IndividualSensorAngle(number position)

The angle of the individual IR sensor with the given zero-based position.

number IndividualSensorStrength(number position)

The strength of the individual IR sensor with the given zero-based position.

text ToString()

Return text representing the state of the hardware device.

Properties

Angle

The Angle. (read only)

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

I2cAddress

The I2C address of the IR seeker sensor.

IndividualSensorCount

The number of individual IR sensors attached to this seeker. (read only)

MAX_NEW_I2C_ADDRESS

The constant for MAX_NEW_I2C_ADDRESS. (read only)

MIN_NEW_I2C_ADDRESS

The constant for MIN_NEW_I2C_ADDRESS. (read only)

Mode

The mode of the IR seeker sensor.

Valid values are Mode_600HZ or Mode_1200HZ.

Mode_1200HZ

The constant for Mode_1200HZ. (read only)

Mode_600HZ

The constant for Mode_600HZ. (read only)

SignalDetected

Whether a signal is detected by the sensor. (read only)

SignalDetectedThreshold

The signal detection threshold of the IR seeker sensor.

Strength

The Strength. (read only)

Version

The version. (read only)

FtcLED

A component for an LED of an FTC robot.

Methods (call blocks)

Enable(boolean set)

Turn on or turn off the LED light.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

FtcLegacyModule

A component for a legacy module of an FTC robot.

Events (when-do blocks)

I2cPortIsReady(number port)

This event is triggered when an I2C port is ready, after the latest data has been read from the I2C Controller.

This event is only enabled if EnableI2cReadMode or EnableI2cWriteMode is used.

Methods (call blocks)

CopyBufferIntoWriteBuffer(number physicalPort, ByteArray byteArray)

Copy a byte array into the buffer that is set to be written out to the device.

Enable9v(number physicalPort, boolean enable)

Enable or disable 9V power on a port.

EnableAnalogReadMode(number physicalPort)

Enable a physical port in analog read mode.

EnableI2cReadMode(number physicalPort, number i2cAddress, number memAddress, number length)

Enable read mode for a particular I2C device and enable the I2cPortIsReady event for the given port.

EnableI2cWriteMode(number physicalPort, number i2cAddress, number memAddress, number length)

Enable write mode for a particular I2C device and enable the I2cPortIsReady event for the given port.

ByteArray GetCopyOfReadBuffer(number physicalPort)

Get a copy of the most recent data read in from the device.

ByteArray GetCopyOfWriteBuffer(number physicalPort)

Get a copy of the data that is set to be written out to the device.

boolean IsI2cPortActionFlagSet(number port)

Get the port action flag; this flag is set if the particular port is busy.

boolean IsI2cPortInReadMode(number port)

Is the port in read mode?

boolean IsI2cPortInWriteMode(number port)

Is the port in write mode?

boolean IsI2cPortReady(number port)

Determine if a physical port is ready.

ByteArray ReadAnalog(number physicalPort)

Read an analog value from a device and return a byte array; only works in analog read mode.

ReadI2cCacheFromController(number port)

Read the local cache in from the I2C Controller.

NOTE: unless this method is called the internal cache isn't updated.

SetDigitalLine(number physicalPort, number line, boolean set)

Set the value of digital line 0 or 1 while in analog mode.

SetI2cPortActionFlag(number port)

Set the port action flag; this flag tells the controller to send the current data in its buffer to the I2C device.

text ToString()

Return text representing the state of the hardware device.

WriteI2cCacheToController(number port)

Write the local cache to the I2C Controller.

NOTE: unless this method is called the internal cache isn't updated.

WriteI2cPortFlagOnlyToController(number port)

Write just the port action flag in the local cache to the I2C controller.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

FtcLightSensor

A component for a light sensor of an FTC robot.

Methods (call blocks)

EnableLed(boolean enable)

Enable the LED light.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

LightDetected

The light detected by the sensor, on a scale of 0 to 1. (read only)

LightDetectedRaw

The light detected by the sensor, as an integer. (read only)

Status

The Status. (read only)

Version

The version. (read only)

FtcLinearOpMode

A component for a linear operation mode for an FTC robot.

Events (when-do blocks)

RunOpMode()

This event is triggered when this op mode is run.

Methods (call blocks)

number GetRuntime()

Get the number of seconds this op mode has been running.

boolean OpModelsActive()

Returns true as long as the op mode is active.

Sleep(number milliseconds)

Sleep for the given amount of milliseconds.

WaitForNextHardwareCycle()

Wait for the start of the next hardware cycle.

Each cycle of the hardware your commands are sent out to the hardware; and the latest data is read back in.

This method will wait for the current hardware cycle to finish, which is also the start of the next hardware cycle.

WaitForStart()

Pause the Linear Op Mode until start has been pressed.

WaitOneFullHardwareCycle()

Wait for one full cycle of the hardware.

Each cycle of the hardware your commands are sent out to the hardware; and the latest data is read back in.

This method has a strong guarantee to wait for at least one full hardware hardware cycle.

Properties

Time

The number of seconds this op mode had been running when the RunOpMode event was triggered. (read only)

This block was added in the 11/05/15 release.

FtcOpMode

A component for an operation mode for an FTC robot.

Events (when-do blocks)

Init()

This event is triggered once when the INIT button is pressed.

InitLoop()

This event is triggered repeatedly when the INIT button is pressed.

Loop()

This event is triggered repeatedly while this op mode is running.

Start()

This event is triggered once when the PLAY button is first pressed.

Stop()

This event is triggered when this op mode is first disabled.

Methods (call blocks)

number GetRuntime()

Get the number of seconds this op mode has been running.

Properties

Time

The number of seconds this op mode had been running when the Loop event was triggered.
(read only)

This block was added in the 11/05/15 release.

FtcOpticalDistanceSensor

A component for an optical distance sensor of an FTC robot.

Methods (call blocks)

EnableLed(boolean enable)

Enable the LED light.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

LightDetected

The light detected by the sensor, on a scale of 0 to 1. (read only)

LightDetectedRaw

The light detected by the sensor, as an integer. (read only)

Status

The Status. (read only)

Version

The version. (read only)

FtcPwmOutput

A component for a PWM output device of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

PulseWidthOutputTime

The pulse width output time for this port.

Typically set to a value between 750 and 2,250 to control a servo.

PulseWidthPeriod

The pulse width output period.

Typically set to 20,000 to control servo.

Version

The version. (read only)

FtcRobotController

The primary FTC Robot Controller component

Methods (call blocks)

CopyBytes(ByteArray sourceByteArray, number sourcePosition, ByteArray destinationByteArray, number destinationPosition, number length)

Copies one or more bytes from the source byte array to the destination byte array.

ByteArray CreateByteArray(number length)

Create a byte array.

number Get1ByteNumberFromByteArray(ByteArray byteArray, number position, boolean unsigned)

Get a 1-byte number from a byte array.

number Get2ByteNumberFromByteArray(ByteArray byteArray, number position, boolean bigEndian)

Get a 2-byte number from a byte array.

number Get4ByteNumberFromByteArray(ByteArray byteArray, number position, boolean bigEndian)

Get a 4-byte number from a byte array.

number Get8ByteNumberFromByteArray(ByteArray byteArray, number position, boolean bigEndian)

Get a 8-byte number from a byte array.

boolean IsBitSet(number bitField, number bitPosition)

Checks the state of a bit in a bit field. Returns true if the target bit is one.

LogDevices()

Log information about hardware devices.

This block was added in the 11/05/15 release.

number LengthOfByteArray(ByteArray byteArray)

Returns the length of a byte array.

Put1ByteNumberIntoByteArray(text number, ByteArray byteArray, number position)

Put a 1-byte number into a byte array.

Put2ByteNumberIntoByteArray(number number, ByteArray byteArray, number position, boolean bigEndian)

Put a 2-byte number into a byte array.

Put4ByteNumberIntoByteArray(number number, ByteArray byteArray, number position, boolean bigEndian)

Put a 4-byte number into a byte array.

Put8ByteNumberIntoByteArray(number number, ByteArray byteArray, number position, boolean bigEndian)

Put a 8-byte number into a byte array.

number RangeClip(number number, number min, number max)

Clip number if number is less than min or greater than max

number RangeScale(number number, number x1, number x2, number y1, number y2)

Scale a number in the range of x1 to x2, to the range of y1 to y2

TelemetryAddNumericData(text key, text number)

Adds a numeric data point to the telemetry for the active op mode.

TelemetryAddTextData(text key, text text)

Adds a text data point to the telemetry for the active op mode.

Properties

BackgroundColor

Returns the background color

LIBRARY_VERSION

The constant for LIBRARY_VERSION. (read only)

UsbScanTimeInSeconds (designer only)

The time reserved for scanning USB devices, in seconds.

FtcServo

A component for a servo of an FTC robot.

Methods (call blocks)

ScaleRange(number min, number max)

Sets the scale range of this servo.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Direction

Whether this servo should spin forward or reverse.

Valid values are Direction_FORWARD or Direction_REVERSE.

Direction_FORWARD

The constant for Direction_FORWARD. (read only)

Direction_REVERSE

The constant for Direction_REVERSE. (read only)

MAX_POSITION

The constant for MAX_POSITION. (read only)

MIN_POSITION

The constant for MIN_POSITION. (read only)

PortNumber

The port number. (read only)

Position

The current servo position, between 0.0 and 1.0.

Version

The version. (read only)

FtcServoController

A component for a servo controller of an FTC robot.

Methods (call blocks)

PwmStatus GetPwmStatus()

Get the PWM status.

Valid values are PwmStatus_ENABLED or PwmStatus_DISABLED.

number GetServoPosition(number channel)

Get the position of a servo at a given channel.

PwmDisable()

PWM disable.

PwmEnable()

PWM enable.

SetServoPosition(number channel, number position)

Set the position of a servo at the given channel.

SetServoPositionAndSpeed(number channel, number position, number speed)

Set a position and a speed for a servo. The speed parameter is ignored if it is not supported by the controller.

This block was added in the 11/05/15 release.

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

PwmStatus_DISABLED

The constant for PwmStatus_DISABLED. (read only)

PwmStatus_ENABLED

The constant for PwmStatus_ENABLED. (read only)

Version

The version. (read only)

FtcTouchSensor

A component for a touch sensor of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

IsPressed

Return true if the touch sensor is being pressed. (read only)

Status

The status, if supported by the touch sensor. (read only)

This block was added in the 11/05/15 release.

Value

Represents how much force is applied to the touch sensor; for some touch sensors this value will only ever be 0 or 1. (read only)

Version

The version. (read only)

FtcTouchSensorMultiplexer

A component for a touch sensor multiplexer of an FTC robot.

Methods (call blocks)

number GetSwitches()

Get switches

boolean IsTouchSensorPressed(number channel)

Is the touch sensor pressed?

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Status

The status, if supported by the touch sensor multiplexer. (read only)

This block was added in the 11/05/15 release.

Version

The version. (read only)

FtcUltrasonicSensor

A component for an ultrasonic sensor of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Status

The Status. (read only)

UltrasonicLevel

The ultrasonic level. (read only)

Version

The version. (read only)

FtcVoltageSensor

A component for a voltage sensor of an FTC robot.

Methods (call blocks)

text ToString()

Return text representing the state of the hardware device.

Properties

ConnectionInfo

The connection information. (read only)

Device

The manufacturer and name of the device. (read only)

Version

The version. (read only)

Voltage

The Voltage. (read only)