
. Documentation

Release 2020.0.0

Author

Mar 16, 2020

1	Authors	3
2	License	5
2.1	ADI API	5
3	Indices and tables	11
	Index	13

RobotPy wrappers for the following Analog Devices products:

- ADIS16448
- ADIS16470

Note: The RobotPy project is not associated with or endorsed by Analog Devices Inc. or FIRST Robotics Competition

CHAPTER 1

Authors

- ADIS16448 support contributed by [@picode98](#)
- ADIS16470 support contributed by [@SHU-matic](#)

MIT License

2.1 ADI API

This is not installed on the robot by default. For installation instructions, see robotpy-adi install docs.

2.1.1 adis16448 Package

<code>adis16448.ADIS16448_IMU(*args, **kwargs)</code>	Overloaded function.
---	----------------------

ADIS16448_IMU

class `adis16448.ADIS16448_IMU(*args, **kwargs)`

Bases: `wpiplib.GyroBase`

Overloaded function.

1. `__init__(self: adis16448._adis16448.ADIS16448_IMU) -> None`

IMU constructor on onboard MXP CS0, Z-up orientation, and complementary AHRS computation.

2. `__init__(self: adis16448._adis16448.ADIS16448_IMU, yaw_axis: adis16448._adis16448.ADIS16448_IMU.IMUAxis, port: wpiplib._wpiplib.SPI.Port, cal_time: int) -> None`

IMU constructor on the specified MXP port and orientation.

Parameters

- **yaw_axis** – The axis where gravity is present. Valid options are kX, kY, and kZ
- **algorithm** – The AHRS algorithm to use. Valid options are kComplementary and kMadgwick

- **port** – The SPI port where the IMU is connected.

class IMUAxis (*arg0: int*) → None

Bases: `pybind11_builtins.pybind11_object`

Members:

kX

kY

kZ

kX = IMUAxis.kX

kY = IMUAxis.kY

kZ = IMUAxis.kZ

name

(self: handle) -> str

calibrate () → None

Initialize the IMU.

Perform gyro offset calibration by collecting data for a number of seconds and computing the center value. The center value is subtracted from subsequent measurements.

It's important to make sure that the robot is not moving while the centering calculations are in progress, this is typically done when the robot is first turned on while it's sitting at rest before the match starts.

The calibration routine can be triggered by the user during runtime.

configCalTime (*new_cal_time: int*) → int

configDecRate (*DecimationRate: int*) → int

getAccelInstantX () → float

getAccelInstantY () → float

getAccelInstantZ () → float

getAngle () → float

Return the actual angle in degrees that the robot is currently facing.

The angle is based on the current accumulator value corrected by offset calibration and built-in IMU calibration. The angle is continuous, that is it will continue from 360->361 degrees. This allows algorithms that wouldn't want to see a discontinuity in the gyro output as it sweeps from 360 to 0 on the second time around. The axis returned by this function is adjusted based on the configured yaw_axis.

Returns the current heading of the robot in degrees. This heading is based on integration of the returned rate from the gyro.

getBarometricPressure () → float

getGyroAngleX () → float

getGyroAngleY () → float

getGyroAngleZ () → float

getGyroInstantX () → float

getGyroInstantY () → float

getGyroInstantZ () → float

getMagInstantX () → float

getMagInstantY () → float

getMagInstantZ () → float

getRate () → float

Return the rate of rotation of the yaw_axis gyro.

The rate is based on the most recent reading of the gyro value

Returns the current rate in degrees per second

getTemperature () → float

getXComplementaryAngle () → float

getXFilteredAccelAngle () → float

getYComplementaryAngle () → float

getYFilteredAccelAngle () → float

getYawAxis () → `adis16448._adis16448.ADIS16448_IMU.IMUAxis`

reset () → None

Reset the gyro.

Resets the gyro accumulations to a heading of zero. This can be used if there is significant drift in the gyro and it needs to be recalibrated after running.

setYawAxis (yaw_axis: `adis16448._adis16448.ADIS16448_IMU.IMUAxis`) → int

2.1.2 adis16470 Package

`adis16470.ADIS16470CalibrationTime`(self, Members:

`arg0`)

`adis16470.ADIS16470_IMU`(*args, **kwargs) Overloaded function.

ADIS16470CalibrationTime

class `adis16470.ADIS16470CalibrationTime` (`arg0: int`) → None

Bases: `pybind11_builtins.pybind11_object`

Members:

`_32ms`

`_64ms`

`_128ms`

`_256ms`

`_512ms`

`_1s`

`_2s`

`_4s`

`_8s`

_16s
_32s
_64s
name
(self: handle) -> str

ADIS16470_IMU

class `adis16470.ADIS16470_IMU (*args, **kwargs)`

Bases: `wpilib.GyroBase`

Overloaded function.

1. `__init__(self: adis16470._adis16470.ADIS16470_IMU) -> None`

Default constructor. Uses CS0 on the 10-pin SPI port, the yaw axis is set to the IMU Z axis, and calibration time is defaulted to 4 seconds.

2. `__init__(self: adis16470._adis16470.ADIS16470_IMU, yaw_axis: adis16470._adis16470.ADIS16470_IMU.IMUAxis, port: wpilib._wpilib.SPI.Port, cal_time: adis16470._adis16470.ADIS16470CalibrationTime) -> None`

Customizable constructor. Allows the SPI port and CS to be customized, the yaw axis used for `GetAngle()` is adjustable, and initial calibration time can be modified.

Parameters

- **yaw_axis** – Selects the “default” axis to use for `GetAngle()` and `GetRate()`
- **port** – The SPI port and CS where the IMU is connected.
- **cal_time** – The calibration time that should be used on start-up.

class `IMUAxis (arg0: int) → None`

Bases: `pybind11_builtins.pybind11_object`

Members:

`kX`

`kY`

`kZ`

kX = `IMUAxis.kX`

kY = `IMUAxis.kY`

kZ = `IMUAxis.kZ`

name
(self: handle) -> str

calibrate () → None

Switches the active SPI port to standard SPI mode, writes the command to activate the new null configuration, and re-enables auto SPI.

configCalTime (*new_cal_time: adis16470._adis16470.ADIS16470CalibrationTime*) → int

Switches the active SPI port to standard SPI mode, writes a new value to the `NULL_CNFG` register in the IMU, and re-enables auto SPI.

configDecRate (*reg: int*) → int

getAccelInstantX () → float

getAccelInstantY() → float

getAccelInstantZ() → float

getAngle() → float

Returns the current integrated angle for the axis specified.

The angle is based on the current accumulator value corrected by offset calibration and built-in IMU calibration. The angle is continuous, that is it will continue from 360->361 degrees. This allows algorithms that wouldn't want to see a discontinuity in the gyro output as it sweeps from 360 to 0 on the second time around. The axis returned by this function is adjusted based on the configured yaw_axis.

Returns the current heading of the robot in degrees. This heading is based on integration of the returned rate from the gyro.

getGyroInstantX() → float

getGyroInstantY() → float

getGyroInstantZ() → float

getRate() → float

getXComplementaryAngle() → float

getXFilteredAccelAngle() → float

getYComplementaryAngle() → float

getYFilteredAccelAngle() → float

getYawAxis() → `adis16470._adis16470.ADIS16470_IMU.IMUAxis`

initSendable (*builder*: `wpilib._wpilib.SendableBuilder`) → None

m_yaw_axis

reset() → None

Resets (zeros) the xgyro, ygyro, and zgyro angle integrations.

Resets the gyro accumulations to a heading of zero. This can be used if the “zero” orientation of the sensor needs to be changed in runtime.

setYawAxis (*yaw_axis*: `adis16470._adis16470.ADIS16470_IMU.IMUAxis`) → int

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

A

ADIS16448_IMU (class in *adis16448*), 5
ADIS16448_IMU.IMUAxis (class in *adis16448*), 6
ADIS16470_IMU (class in *adis16470*), 8
ADIS16470_IMU.IMUAxis (class in *adis16470*), 8
ADIS16470CalibrationTime (class in *adis16470*), 7

C

calibrate() (*adis16448.ADIS16448_IMU* method), 6
calibrate() (*adis16470.ADIS16470_IMU* method), 8
configCalTime() (*adis16448.ADIS16448_IMU* method), 6
configCalTime() (*adis16470.ADIS16470_IMU* method), 8
configDecRate() (*adis16448.ADIS16448_IMU* method), 6
configDecRate() (*adis16470.ADIS16470_IMU* method), 8

G

getAccelInstantX() (*adis16448.ADIS16448_IMU* method), 6
getAccelInstantX() (*adis16470.ADIS16470_IMU* method), 8
getAccelInstantY() (*adis16448.ADIS16448_IMU* method), 6
getAccelInstantY() (*adis16470.ADIS16470_IMU* method), 8
getAccelInstantZ() (*adis16448.ADIS16448_IMU* method), 6
getAccelInstantZ() (*adis16470.ADIS16470_IMU* method), 9
getAngle() (*adis16448.ADIS16448_IMU* method), 6
getAngle() (*adis16470.ADIS16470_IMU* method), 9
getBarometricPressure() (*adis16448.ADIS16448_IMU* method), 6

getGyroAngleX() (*adis16448.ADIS16448_IMU* method), 6
getGyroAngleY() (*adis16448.ADIS16448_IMU* method), 6
getGyroAngleZ() (*adis16448.ADIS16448_IMU* method), 6
getGyroInstantX() (*adis16448.ADIS16448_IMU* method), 6
getGyroInstantX() (*adis16470.ADIS16470_IMU* method), 9
getGyroInstantY() (*adis16448.ADIS16448_IMU* method), 6
getGyroInstantY() (*adis16470.ADIS16470_IMU* method), 9
getGyroInstantZ() (*adis16448.ADIS16448_IMU* method), 6
getGyroInstantZ() (*adis16470.ADIS16470_IMU* method), 9
getMagInstantX() (*adis16448.ADIS16448_IMU* method), 6
getMagInstantY() (*adis16448.ADIS16448_IMU* method), 7
getMagInstantZ() (*adis16448.ADIS16448_IMU* method), 7
getRate() (*adis16448.ADIS16448_IMU* method), 7
getRate() (*adis16470.ADIS16470_IMU* method), 9
getTemperature() (*adis16448.ADIS16448_IMU* method), 7
getXComplementaryAngle() (*adis16448.ADIS16448_IMU* method), 7
getXComplementaryAngle() (*adis16470.ADIS16470_IMU* method), 9
getXFilteredAccelAngle() (*adis16448.ADIS16448_IMU* method), 7
getXFilteredAccelAngle() (*adis16470.ADIS16470_IMU* method), 9
getYawAxis() (*adis16448.ADIS16448_IMU* method), 7
getYawAxis() (*adis16470.ADIS16470_IMU* method), 9

`getYComplementaryAngle()`
 (*adis16448.ADIS16448_IMU method*), 7
`getYComplementaryAngle()`
 (*adis16470.ADIS16470_IMU method*), 9
`getYFilteredAccelAngle()`
 (*adis16448.ADIS16448_IMU method*), 7
`getYFilteredAccelAngle()`
 (*adis16470.ADIS16470_IMU method*), 9

I

`initSendable()` (*adis16470.ADIS16470_IMU method*), 9

K

`kX` (*adis16448.ADIS16448_IMU.IMUAxis attribute*), 6
`kX` (*adis16470.ADIS16470_IMU.IMUAxis attribute*), 8
`kY` (*adis16448.ADIS16448_IMU.IMUAxis attribute*), 6
`kY` (*adis16470.ADIS16470_IMU.IMUAxis attribute*), 8
`kZ` (*adis16448.ADIS16448_IMU.IMUAxis attribute*), 6
`kZ` (*adis16470.ADIS16470_IMU.IMUAxis attribute*), 8

M

`m_yaw_axis` (*adis16470.ADIS16470_IMU attribute*), 9

N

`name` (*adis16448.ADIS16448_IMU.IMUAxis attribute*),
 6
`name` (*adis16470.ADIS16470_IMU.IMUAxis attribute*),
 8
`name` (*adis16470.ADIS16470CalibrationTime attribute*),
 8

R

`reset()` (*adis16448.ADIS16448_IMU method*), 7
`reset()` (*adis16470.ADIS16470_IMU method*), 9

S

`setYawAxis()` (*adis16448.ADIS16448_IMU method*),
 7
`setYawAxis()` (*adis16470.ADIS16470_IMU method*),
 9