sparkfun$_{q}wiic_{i}cm20948$

Release 0.0.1

Apr 17, 2020

Contents:

Python module for the qwiic ICM-20948 sensor, which is included on the SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)

This python package is a port of the existing SparkFun ICM-20948 Arduino Library

This package can be used in conjunction with the overall SparkFun qwiic Python Package

New to qwiic? Take a look at the entire SparkFun qwiic ecosystem.

CHAPTER 1

Contents

CHAPTER 2

Supported Platforms

The qwiic ICM20948 Python package current supports the following platforms:

- Raspberry Pi
- NVidia Jetson Nano
- Google Coral Development Board

CHAPTER 3

Dependencies

This driver package depends on the qwiic I2C driver: Qwiic_I2C_Py

CHAPTER 4

Documentation

The SparkFun qwiic ICM20948 module documentation is hosted at ReadTheDocs

CHAPTER 5

Installation

5.1 PyPi Installation

This repository is hosted on PyPi as the sparkfun-qwiic-icm20948 package. On systems that support PyPi installation via pip, this library is installed using the following commands

For all users (note: the user must have sudo privileges):

```
sudo pip install sparkfun-qwiic-icm20948
```

For the current user:

```
pip install sparkfun-qwiic-icm20948
```

5.2 Local Installation

To install, make sure the setuptools package is installed on the system.

Direct installation at the command line:

```
python setup.py install
```

To build a package for use with pip:

```
python setup.py sdist
```

A package file is built and placed in a subdirectory called dist. This package file can be installed using pip.

```
cd dist
pip install sparkfun_qwiic_icm20948-<version>.tar.gz
```

CHAPTER 6

Example Use

See the examples directory for more detailed use examples.

```
from __future__ import print_function
import qwiic_icm20948
import time
import sys

def runExample():

    print("\nSparkFun 9DoF ICM-20948 Sensor  Example 1\n")
    IMU = qwiic_icm20948.QwiicIcm20948()

    if IMU.connected == False:
        print("The Qwiic ICM20948 device isn't connected to the system. Please check␣
↪your connection", \
            file=sys.stderr)
        return

    IMU.begin()

    while True:
        if IMU.dataReady():
            IMU.getAgmt() # read all axis and temp from sensor, note this also␣
↪updates all instance variables
            print(\
             '{: 06d}'.format(IMU.axRaw)\
            , '\t', '{: 06d}'.format(IMU.ayRaw)\
            , '\t', '{: 06d}'.format(IMU.azRaw)\
            , '\t', '{: 06d}'.format(IMU.gxRaw)\
            , '\t', '{: 06d}'.format(IMU.gyRaw)\
            , '\t', '{: 06d}'.format(IMU.gzRaw)\
            , '\t', '{: 06d}'.format(IMU.mxRaw)\
            , '\t', '{: 06d}'.format(IMU.myRaw)\
            , '\t', '{: 06d}'.format(IMU.mzRaw)\
```

(continues on next page)

(continued from previous page)

```
            )
            time.sleep(0.03)
        else:
            print("Waiting for data")
            time.sleep(0.5)

if __name__ == '__main__':
    try:
        runExample()
    except (KeyboardInterrupt, SystemExit) as exErr:
        print("\nEnding Example 1")
        sys.exit(0)
```

CHAPTER 7

Table of Contents

7.1 API Reference

7.1.1 qwiic_icm20948

Python module for the [SparkFun 9DoF IMU Breakout - ICM-20948 (Qwiic)](https://www.sparkfun.com/products/15335)

This python package is a port of the existing [SparkFun ICM-20948 Arduino Library](https://github.com/sparkfun/SparkFun_ICM-20948_ArduinoLibrary)

This package can be used in conjunction with the overall [SparkFun qwiic Python Package](https://github.com/sparkfun/Qwiic_Py)

New to qwiic? Take a look at the entire [SparkFun qwiic ecosystem](https://www.sparkfun.com/qwiic).

class qwiic_icm20948.**QwiicIcm20948**(*address=None*, *i2c_driver=None*)

Parameters

- **address** – The I2C address to use for the device. If not provided, the default address is used.
- **i2c_driver** – An existing i2c driver object. If not provided a driver object is created.

Returns The ICM20948 device object.

Return type Object

ToSignedInt(*input*)

Takes an input data of 16 bits, and returns the signed 32 bit int version of this data

Returns Signed 32 bit integer

Return type int

begin()

Initialize the operation of the ICM20948 module

Returns Returns true of the initializtion was successful, otherwise False.

Return type bool

connected
Determine if a ICM20948 device is conntected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

dataReady()
Returns status of RAW_DATA_0_RDY_INT the ICM90248 module

Returns Returns true if raw data is ready, otherwise False.

Return type bool

enableDlpfAccel(*on*)
Enables or disables the accelerometer DLPF of the ICM90248 module

Returns Returns true if the DLPF mode setting write was successful, otherwise False.

Return type bool

enableDlpfGyro(*on*)
Enables or disables the Gyro DLPF of the ICM90248 module

Returns Returns true if the DLPF mode setting write was successful, otherwise False.

Return type bool

getAgmt()
Reads and updates raw values from accel, gyro, mag and temp of the ICM90248 module

Returns Returns True if I2C readBlock was successful, otherwise False.

Return type bool

i2cMasterConfigureSlave(*slave*, *addr*, *reg*, *len*, *Rw*, *enable*, *data_only*, *grp*, *swap*)
Configures Master/slave settings for the ICM20948 as master, and slave in slots 0-3

Returns Returns true if the configuration was successful, otherwise False.

Return type bool

i2cMasterEnable(*enable*)
Enables or disables I2C Master

Returns Returns true if the setting write was successful, otherwise False.

Return type bool

i2cMasterPassthrough(*passthrough*)
Enables or disables I2C Master Passthrough

Returns Returns true if the setting write was successful, otherwise False.

Return type bool

i2cMasterReset()
Resets I2C Master Module

Returns Returns true if the i2c write was successful, otherwise False.

Return type bool

isConnected()
Determine if a ICM20948 device is conntected to the system..

Returns True if the device is connected, otherwise False.

Return type bool

lowPower(*on*)
Sets the ICM20948 module in or out of low power mode

Returns Returns true if the power mode setting write was successful, otherwise False.

Return type bool

magWhoIAm()
Checks to see that the Magnatometer returns the correct ID value

Returns Returns true if the check was successful, otherwise False.

Return type bool

setBank(*bank*)
Sets the bank register of the ICM20948 module

Returns Returns true if the bank was a valid value and it was set, otherwise False.

Return type bool

setDLPFcfgAccel(*dlpcfg*)
Sets the digital low pass filter for the accel in the ICM20948 module

Returns Returns true if the dlp setting write was successful, otherwise False.

Return type bool

setDLPFcfgGyro(*dlpcfg*)
Sets the digital low pass filter for the gyro in the ICM20948 module

Returns Returns true if the dlp setting write was successful, otherwise False.

Return type bool

setFullScaleRangeAccel(*mode*)
Sets the full scale range for the accel in the ICM20948 module

Returns Returns true if the full scale range setting write was successful, otherwise False.

Return type bool

setFullScaleRangeGyro(*mode*)
Sets the full scale range for the gyro in the ICM20948 module

Returns Returns true if the full scale range setting write was successful, otherwise False.

Return type bool

setSampleMode(*sensors*, *mode*)
Sets the sample mode of the ICM90248 module

Returns Returns true if the sample mode setting write was successful, otherwise False.

Return type bool

sleep(*on*)
Sets the ICM20948 module in or out of sleep mode

Returns Returns true if the sleep setting write was successful, otherwise False.

Return type bool

startupMagnetometer()
Initialize the magnotometer with default values

Returns Returns true of the initializtion was successful, otherwise False.

Return type bool

swReset()
Performs a software reset on the ICM20948 module

Returns Returns true if the software reset was successful, otherwise False.

Return type bool

7.2 Basic Operation

Listing 1: examples/ex1_qwiic_ICM20948.py

```
#!/usr/bin/env python
#-----------------------------------------------------------------------------
# ex1_qwiic_ICM20948.py
#
# Simple Example for the Qwiic ICM20948 Device
#-----------------------------------------------------------------------------
#
# Written by  SparkFun Electronics, March 2020
#
# This python library supports the SparkFun Electroncis qwiic
# qwiic sensor/board ecosystem on a Raspberry Pi (and compatable) single
# board computers.
#
# More information on qwiic is at https://www.sparkfun.com/qwiic
#
# Do you like this library? Help support SparkFun. Buy a board!
#
#==================================================================================
# Copyright (c) 2019 SparkFun Electronics
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#==================================================================================
```

(continues on next page)

(continued from previous page)

```
# Example 1
#

from __future__ import print_function
import qwiic_icm20948
import time
import sys

def runExample():

        print("\nSparkFun 9DoF ICM-20948 Sensor  Example 1\n")
        IMU = qwiic_icm20948.QwiicIcm20948()

        if IMU.connected == False:
                print("The Qwiic ICM20948 device isn't connected to the system. Please check your connection", \
                        file=sys.stderr)
                return

        IMU.begin()

        while True:
                if IMU.dataReady():
                        IMU.getAgmt() # read all axis and temp from sensor, note this also updates all instance variables
                        print(\
                         '{: 06d}'.format(IMU.axRaw)\
                        , '\t', '{: 06d}'.format(IMU.ayRaw)\
                        , '\t', '{: 06d}'.format(IMU.azRaw)\
                        , '\t', '{: 06d}'.format(IMU.gxRaw)\
                        , '\t', '{: 06d}'.format(IMU.gyRaw)\
                        , '\t', '{: 06d}'.format(IMU.gzRaw)\
                        , '\t', '{: 06d}'.format(IMU.mxRaw)\
                        , '\t', '{: 06d}'.format(IMU.myRaw)\
                        , '\t', '{: 06d}'.format(IMU.mzRaw)\
                        )
                        time.sleep(0.03)
                else:
                        print("Waiting for data")
                        time.sleep(0.5)

if __name__ == '__main__':
        try:
                runExample()
        except (KeyboardInterrupt, SystemExit) as exErr:
                print("\nEnding Example 1")
                sys.exit(0)


```

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

q

Index