AdafruitBNO08X Library Documentation

Release 1.0

Bryan Siepert

Jun 07, 2021

Contents

Helper library for the Hillcrest Laboratories BNO08x IMUs

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-bno08x
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-bno08x
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-bno08x
```

CHAPTER 3

Usage Example

```
import board
import busio
from adafruit_bno08x.i2c import BNO08X_I2C
from adafruit_bno08x import BNO_REPORT_ACCELEROMETER

i2c = busio.I2C(board.SCL, board.SDA)
bno = BNO08X_I2C(i2c)
bno.enable_feature(BNO_REPORT_ACCELEROMETER)

while True:
    accel_x, accel_y, accel_z = bno.acceleration  # pylint:disable=no-member
    print("X: %0.6f  Y: %0.6f Z: %0.6f  m/s^2" % (accel_x, accel_y, accel_z))
```

CHAPTER 4

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/bno08x_simpletest.py

```
# SPDX-FileCopyrightText: 2020 Bryan Siepert, written for Adafruit Industries
#
# SPDX-License-Identifier: Unlicense
import time
import board
import busio
from adafruit_bno08x import (
    BNO_REPORT_ACCELEROMETER,
    BNO_REPORT_GYROSCOPE,
    BNO_REPORT_MAGNETOMETER,
    BNO_REPORT_ROTATION_VECTOR,
)
from adafruit_bno08x.i2c import BNO08X_I2C

i2c = busio.I2C(board.SCL, board.SDA, frequency=400000)
bno = BNO08X_I2C(i2c)

bno.enable_feature(BNO_REPORT_ACCELEROMETER)
bno.enable_feature(BNO_REPORT_GYROSCOPE)
bno.enable_feature(BNO_REPORT_MAGNETOMETER)
bno.enable_feature(BNO_REPORT_ROTATION_VECTOR)

while True:

    time.sleep(0.5)
    print("Acceleration:")
    accel_x, accel_y, accel_z = bno.acceleration  # pylint:disable=no-member
```

(continues on next page)

(continued from previous page)

```
    print("X: %0.6f  Y: %0.6f Z: %0.6f  m/s^2" % (accel_x, accel_y, accel_z))
    print("")

    print("Gyro:")
    gyro_x, gyro_y, gyro_z = bno.gyro  # pylint:disable=no-member
    print("X: %0.6f  Y: %0.6f Z: %0.6f rads/s" % (gyro_x, gyro_y, gyro_z))
    print("")

    print("Magnetometer:")
    mag_x, mag_y, mag_z = bno.magnetic  # pylint:disable=no-member
    print("X: %0.6f  Y: %0.6f Z: %0.6f uT" % (mag_x, mag_y, mag_z))
    print("")

    print("Rotation Vector Quaternion:")
    quat_i, quat_j, quat_k, quat_real = bno.quaternion  # pylint:disable=no-member
    print(
        "I: %0.6f  J: %0.6f K: %0.6f  Real: %0.6f" % (quat_i, quat_j, quat_k, quat_
↪real)
    )
    print("")
```

6.2 adafruit_bno08x

Helper library for the Hillcrest Laboratories BNO08x IMUs

- Author(s): Bryan Siepert

6.2.1 Implementation Notes

Hardware:

- Adafruit BNO08x Breakout

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https:# github.com/adafruit/circuitpython/releases
- Adafruit's Bus Device library

class adafruit_bno08x.**BNO08X**(*reset=None*, *debug=False*)

Library for the BNO08x IMUs from Hillcrest Laboratories

Parameters **i2c_bus** (*I2C*) – The I2C bus the BNO08x is connected to.

acceleration

A tuple representing the acceleration measurements on the X, Y, and Z axes in meters per second squared

activity_classification

- "Unknown"
- "In-Vehicle"
- "On-Bicycle"
- "On-Foot"
- "Still"

- "Tilting"
- "Walking"
- "Running"
- "On Stairs"

Type Returns the sensor's assessment of the activity that is creating the motions that it is sensing, one of

begin_calibration()
Begin the sensor's self-calibration routine

calibration_status
Get the status of the self-calibration

enable_feature(*feature_id*)
Used to enable a given feature of the BNO08x

game_quaternion
A quaternion representing the current rotation vector expressed as a quaternion with no specific reference for heading, while roll and pitch are referenced against gravity. To prevent sudden jumps in heading due to corrections, the *game_quaternion* property is not corrected using the magnetometer. Some drift is expected

geomagnetic_quaternion
A quaternion representing the current geomagnetic rotation vector

gyro
A tuple representing Gyro's rotation measurements on the X, Y, and Z axes in radians per second

hard_reset()
Hardware reset the sensor to an initial unconfigured state

initialize()
Initialize the sensor

linear_acceleration
A tuple representing the current linear acceleration values on the X, Y, and Z axes in meters per second squared

magnetic
A tuple of the current magnetic field measurements on the X, Y, and Z axes

quaternion
A quaternion representing the current rotation vector

raw_acceleration
Returns the sensor's raw, unscaled value from the accelerometer registers

raw_gyro
Returns the sensor's raw, unscaled value from the gyro registers

raw_magnetic
Returns the sensor's raw, unscaled value from the magnetometer registers

save_calibration_data()
Save the self-calibration data

shake
True if a shake was detected on any axis since the last time it was checked

This property has a "latching" behavior where once a shake is detected, it will stay in a "shaken" state until the value is read. This prevents missing shake events but means that this property is not guaranteed to reflect the shake state at the moment it is read

soft_reset()
Reset the sensor to an initial unconfigured state

stability_classification

- "Unknown" - The sensor is unable to classify the current stability
- "On Table" - The sensor is at rest on a stable surface with very little vibration
- "Stationary" - The sensor's motion is below the stable threshold but the stable duration requirement has not been met. This output is only available when gyro calibration is enabled
- "Stable" - The sensor's motion has met the stable threshold and duration requirements.
- "In motion" - The sensor is moving.

Type Returns the sensor's assessment of it's current stability, one of

steps
The number of steps detected since the sensor was initialized

class adafruit_bno08x.**Packet**(*packet_bytes*)
A class representing a Hillcrest LaboratorySensor Hub Transport packet

channel_number
The packet channel

classmethod header_from_buffer(*packet_bytes*)
Creates a *PacketHeader* object from a given buffer

classmethod is_error(*header*)
Returns True if the header is an error condition

report_id
The Packet's Report ID

exception adafruit_bno08x.**PacketError**
Raised when the packet couldnt be parsed

class adafruit_bno08x.**PacketHeader**(*channel_number*, *sequence_number*, *data_length*, *packet_byte_count*)

channel_number
Alias for field number 0

data_length
Alias for field number 2

packet_byte_count
Alias for field number 3

sequence_number
Alias for field number 1

adafruit_bno08x.**parse_sensor_id**(*buffer*)
Parse the fields of a product id report

- Adafruit BNO08x Breakout

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index