AdafruitExtended$_BusLibraryDocumentation$

Release 1.0

Melissa LeBlanc-Williams

Jun 14, 2021

Contents

1 Dependencies 3

2 Installing from PyPI 5

3 Usage Example 7

4 Contributing 9

5 Documentation 11

6 Table of Contents 13
- 6.1 Simple test 13
- 6.2 adafruit_extended_bus 13

7 Indices and tables 15

Python Module Index 17

Index 19

Helper Library for Blinka to allow creating I2C and SPI busio objects by passing in the Bus ID. This library is not compatible with CircuitPython and is intended to only be run on Linux devices.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit Python

Please ensure all dependencies are available on the Python 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-extended-bus
```

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

```
sudo pip3 install adafruit-extended-bus
```

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-extended-bus
```

CHAPTER 3

Usage Example

```python
from adafruit_extended_bus import ExtendedI2C as I2C
import adafruit_bme280

# Create library object using our Extended Bus I2C port
i2c = I2C(1) # Device is /dev/i2c-1
bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)
print("\nTemperature: %0.1f C" % bme280.temperature)
```

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/extended_bus_simpletest.py

```python
"""
This exmaple demonstrates how to instantiate the
Adafruit BME280 Sensor using this library and just
the I2C bus number.
"""

import adafruit_bme280
from adafruit_extended_bus import ExtendedI2C as I2C

# Create library object using our Extended Bus I2C port
i2c = I2C(1)  # Device is /dev/i2c-1
bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)
print("\nTemperature: %0.1f C" % bme280.temperature)
```

6.2 adafruit_extended_bus

Helper Library for Blinka to allow creating I2C and SPI busio objects by passing in the Bus ID. This library is not compatible with CircuitPython and is intended to only be run on Linux devices.

- Author(s): Melissa LeBlanc-Williams

class adafruit_extended_bus.**ExtendedI2C**(*bus_id*, *frequency=400000*)

Extended I2C is a busio extension that allows creating a compatible I2C object using the Bus ID number. The bus ID is the number at the end of /dev/i2c-# and you can find which I2C devices you have by typing `ls /dev/i2c*`

class `adafruit_extended_bus.`**`ExtendedSPI`**(*bus_id*, *chip_select*)

Extended SPI is a busio extension that allows creating a compatible SPI object using the Bus ID number. The bus ID is the numbers at the end of /dev/spidev#.# and you can find which SPI devices you have by typing `ls /dev/spi*`

class **`Pin`**(*id*)

Fake Pin class

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_extended_bus, 13

Index

A

adafruit_extended_bus (*module*), 13

E

ExtendedI2C (*class in adafruit_extended_bus*), 13
ExtendedSPI (*class in adafruit_extended_bus*), 13
ExtendedSPI.Pin (*class in adafruit_extended_bus*), 14