AdafruitTFmini Library Documentation

Release 1.0

ladyada

Oct 21, 2019

Contents

A CircuitPython/Python library for Benewake's TF mini distance sensor

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython

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

1.1 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-tfmini
```

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

```
sudo pip3 install adafruit-circuitpython-tfmini
```

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-tfmini
```

CHAPTER 2

Usage Example

```
import time
import board  # comment this out if using pyserial
import busio  # comment this out if using pyserial
import adafruit_tfmini

# Use hardware uart
uart = busio.UART(board.TX, board.RX)

# Or, you can use pyserial on any computer
#import serial
#uart = serial.Serial("/dev/ttyS2", timeout=1)

# Simplest use, connect with the uart bus object
tfmini = adafruit_tfmini.TFmini(uart)

# You can put in 'short' or 'long' distance mode
tfmini.mode = adafruit_tfmini.MODE_SHORT
print("Now in mode", tfmini.mode)

while True:
    print("Distance: %d cm (strength %d, mode %x)" %
          (tfmini.distance, tfmini.strength, tfmini.mode))
    time.sleep(0.1)
```

CHAPTER 3

Contributing

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

CHAPTER 4

Building locally

4.1 Zip release files

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-tfmini --library_
↪location .
```

4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tfmini_simpletest.py

```
import time
import board  # comment this out if using pyserial
import busio  # comment this out if using pyserial
import adafruit_tfmini

# Use hardware uart
uart = busio.UART(board.TX, board.RX)

# Or, you can use pyserial on any computer
#import serial
#uart = serial.Serial("/dev/ttyS2", timeout=1)

# Simplest use, connect with the uart bus object
tfmini = adafruit_tfmini.TFmini(uart)

# You can put in 'short' or 'long' distance mode
tfmini.mode = adafruit_tfmini.MODE_SHORT
print("Now in mode", tfmini.mode)

while True:
    print("Distance: %d cm (strength %d, mode %x)" %
          (tfmini.distance, tfmini.strength, tfmini.mode))
    time.sleep(0.1)
```

5.2 adafruit_tfmini

A CircuitPython/Python library for Benewake's TF mini distance sensor

- Author(s): ladyada

5.2.1 Implementation Notes

Hardware:

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases

class adafruit_tfmini.**TFmini**(*uart*, *, *timeout=1*)

TF mini communication module, use with just RX or TX+RX for advanced command & control. :param uart: the pyseral or busio.uart compatible uart device :param timeout: how long we'll wait for valid data or response, in seconds. Default is 1

distance

The most recent distance measurement in centimeters

mode

The measurement mode can be MODE_SHORT (2) or MODE_LONG (7)

strength

The signal validity, higher value means better measurement

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index