Adafruit IRREMOTE Library Documentation

Release 1.0

Scott Shawcroft

Jan 15, 2019

Contents

CircuitPython driver for use with IR Receivers.

Examples of products to use this library with:

- Circuit Playground Express
- IR Receiver 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.

CHAPTER 2

Usage Example

```
# Circuit Playground Express Demo Code
# Adjust the pulseio 'board.PIN' if using something else
import pulseio
import board
import adafruit_irremote
pulsein = pulseio.PulseIn(board.REMOTEIN, maxlen=120, idle_state=True)
decoder = adafruit_irremote.GenericDecode()

# size must match what you are decoding! for NEC use 4
received_code = bytearray(4)

while True:
    pulses = decoder.read_pulses(pulsein)
    print("Heard", len(pulses), "Pulses:", pulses)
    try:
        code = decoder.decode_bits(pulses, debug=False)
        print("Decoded:", code)
    except adafruit_irremote.IRNECRepeatException:  # unusual short code!
        print("NEC repeat!")
    except adafruit_irremote.IRDecodeException as e:     # failed to decode
        print("Failed to decode: ", e.args)

    print("----------------------------")
```

CHAPTER 3

Contributing

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

CHAPTER 4

Building locally

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-irremote --
↪library_location .
```

4.1 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/irremote_simpletest.py

```
# Circuit Playground Express Demo Code
# Adjust the pulseio 'board.PIN' if using something else
import pulseio
import board
import adafruit_irremote
pulsein = pulseio.PulseIn(board.REMOTEIN, maxlen=120, idle_state=True)
decoder = adafruit_irremote.GenericDecode()

# size must match what you are decoding! for NEC use 4
received_code = bytearray(4)

while True:
    pulses = decoder.read_pulses(pulsein)
    print("Heard", len(pulses), "Pulses:", pulses)
    try:
        code = decoder.decode_bits(pulses, debug=False)
        print("Decoded:", code)
    except adafruit_irremote.IRNECRepeatException:  # unusual short code!
        print("NEC repeat!")
    except adafruit_irremote.IRDecodeException as e:     # failed to decode
        print("Failed to decode: ", e.args)

    print("----------------------------")
```

5.2 adafruit_irremote

Demo code for Circuit Playground Express:

```
# Circuit Playground Express Demo Code
# Adjust the pulseio 'board.PIN' if using something else
import pulseio
import board
import adafruit_irremote
pulsein = pulseio.PulseIn(board.REMOTEIN, maxlen=120, idle_state=True)
decoder = adafruit_irremote.GenericDecode()

# size must match what you are decoding! for NEC use 4
received_code = bytearray(4)

while True:
    pulses = decoder.read_pulses(pulsein)
    print("Heard", len(pulses), "Pulses:", pulses)
    try:
        code = decoder.decode_bits(pulses, debug=False)
        print("Decoded:", code)
    except adafruit_irremote.IRNECRepeatException:  # unusual short code!
        print("NEC repeat!")
    except adafruit_irremote.IRDecodeException as e:     # failed to decode
        print("Failed to decode: ", e.args)

    print("----------------------------")
```

- Author(s): Scott Shawcroft

5.2.1 Implementation Notes

Hardware:

- CircuitPlayground Express
- IR Receiver Sensor

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: https://github.com/adafruit/circuitpython/releases

class adafruit_irremote.**GenericDecode**

Generic decoding of infrared signals

bin_data(*pulses*)

Compute bins of pulse lengths where pulses are +-25% of the average.

Parameters **pulses** (*list*) – Input pulse lengths

decode_bits(*pulses*, *debug=False*)

Decode the pulses into bits.

read_pulses(*input_pulses*, ***, *max_pulse=10000*, *blocking=True*, *pulse_window=0.1*, *blocking_delay=0.1*)

Read out a burst of pulses until pulses stop for a specified period (pulse_window), pruning pulses after a pulse longer than max_pulse.

Parameters

- **input_pulses** (*PulseIn*) – Object to read pulses from
- **max_pulse** (*int*) – Pulse duration to end a burst
- **blocking** (*bool*) – If True, will block until pulses found. If False, will return None if no pulses. Defaults to True for backwards compatibility
- **pulse_window** (*float*) – pulses are collected for this period of time
- **blocking_delay** (*float*) – delay between pulse checks when blocking

class adafruit_irremote.**GenericTransmit**(*header*, *one*, *zero*, *trail*)
Generic infrared transmit class that handles encoding.

transmit(*pulseout*, *data*)
Transmit the `data` using the `pulseout`.

Parameters

- **pulseout** (*pulseio.PulseOut*) – PulseOut to transmit on
- **data** (*bytearray*) – Data to transmit

exception adafruit_irremote.**IRDecodeException**
Generic decode exception

exception adafruit_irremote.**IRNECRepeatException**
Exception when a NEC repeat is decoded

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index