AdafruitOneWire Library Documentation

Release 1.0

Carter Nelson

Mar 07, 2018

Contents

Classes for use in communicating with devices on a 1-Wire bus.

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

```
import board
from adafruit_onewire.bus import OneWireBus
ow_bus = OneWireBus(board.D2)
devices = ow_bus.scan()
for d in devices:
    print("ROM={}\tFamily=0x{:02x}".format(d.rom, d.family_code))
```

CHAPTER 3

API Reference

3.1 OneWire Bus

Provide access to a 1-Wire bus.

- Author(s): Carter Nelson

class adafruit_onewire.bus.**OneWireAddress**(*rom*)
A class to represent a 1-Wire address.

crc
The 8 bit CRC.

family_code
The 8 bit family code.

rom
The unique 64 bit ROM code.

serial_number
The 48 bit serial number.

class adafruit_onewire.bus.**OneWireBus**(*pin*)
A class to represent a 1-Wire bus.

static crc8(*data*)
Perform the 1-Wire CRC check on the provided data.

Parameters **data** (*bytearray*) – 8 byte array representing 64 bit ROM code

readinto(*buf*, *, *start=0*, *end=None*)
Read into `buf` from the device. The number of bytes read will be the length of `buf`.

If `start` or `end` is provided, then the buffer will be sliced as if `buf[start:end]`. This will not cause an allocation like `buf[start:end]` will so it saves memory.

Parameters

- **buf** (*bytearray*) – buffer to write into

- **start** (*int*) – Index to start writing at
- **end** (*int*) – Index to write up to but not include

reset(*required=False*)
Perform a reset and check for presence pulse.

Parameters **required** (*bool*) – require presence pulse

scan()
Scan for devices on the bus and return a list of addresses.

write(*buf*, *, *start=0*, *end=None*)
Write the bytes from `buf` to the device.

If `start` or `end` is provided, then the buffer will be sliced as if `buffer[start:end]`. This will not cause an allocation like `buffer[start:end]` will so it saves memory.

Parameters

- **buf** (*bytearray*) – buffer containing the bytes to write
- **start** (*int*) – Index to start writing from
- **end** (*int*) – Index to read up to but not include

exception adafruit_onewire.bus.**OneWireError**
A class to represent a 1-Wire exception.

3.2 OneWire Device

Provides access to a single device on the 1-Wire bus.

- Author(s): Carter Nelson

class adafruit_onewire.device.**OneWireDevice**(*bus*, *address*)
A class to represent a single device on the 1-Wire bus.

readinto(*buf*, *, *start=0*, *end=None*)
Read into `buf` from the device. The number of bytes read will be the length of `buf`.

If `start` or `end` is provided, then the buffer will be sliced as if `buf[start:end]`. This will not cause an allocation like `buf[start:end]` will so it saves memory.

Parameters

- **buf** (*bytearray*) – buffer to write into
- **start** (*int*) – Index to start writing at
- **end** (*int*) – Index to write up to but not include

write(*buf*, *, *start=0*, *end=None*)
Write the bytes from `buf` to the device.

If `start` or `end` is provided, then the buffer will be sliced as if `buffer[start:end]`. This will not cause an allocation like `buffer[start:end]` will so it saves memory.

Parameters

- **buf** (*bytearray*) – buffer containing the bytes to write
- **start** (*int*) – Index to start writing from
- **end** (*int*) – Index to read up to but not include

CHAPTER 4

Contributing

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

CHAPTER 5

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-onewire --
↪library_location .
```

Python Module Index

a

Index