
arduino-serial-mux

Release 1.0.0

Jeroen F.J. Laros

Feb 06, 2022

CONTENTS:

1 Quick start	3
1.1 Introduction	3
1.2 Installation	3
1.3 Usage	4
1.4 Command Line Interface	5
1.5 API documentation	5
1.6 Contributors	6
Python Module Index	7
Index	9

This Python library provides a simple way to create virtual serial interfaces created on an [Arduino](#) using the `serialMux` protocol.

Please see [ReadTheDocs](#) for the latest documentation.

QUICK START

Plug in the Arduino device and run the serial multiplexer.

```
$ serial_mux /dev/ttyUSB0
Detected 2 virtual serial ports.
  Virtual serial device 1: /dev/pts/3
  Virtual serial device 2: /dev/pts/10
```

The virtual serial ports `/dev/pts/8` and `/dev/pts/9` can now be used to communicate to the virtual serial devices created on the device.

For more information, see the [Usage](#) section.

1.1 Introduction

This Python library provides a simple way to create virtual serial interfaces created on an [Arduino](#) using the [serialMux](#) protocol.

For more background information and the reasons that led to this project, see the [motivation](#) section of the device library documentation.

This project serves as a reference implementation for clients using the serialMux protocol.

1.2 Installation

The software is distributed via [PyPI](#), it can be installed with `pip`.

```
pip install arduino-serial-mux
```

1.2.1 From source

The source is hosted on [GitHub](#), to install the latest development version, use the following commands.

```
git clone https://github.com/jfjlaros/arduino-serial-mux.git
cd arduino-serial-mux
pip install .
```

Development

Tests are written in the `pytest` framework which can be installed with `pip`.

```
pip install pytest
```

To run the automated tests, run `py.test` in the root of the project folder.

1.3 Usage

The command line interface can be used to create virtual serial devices. For more information, use the `-h` option.

```
$ serial_mux -h
```

1.3.1 Initialisation

If the Arduino runs code that makes use of the `serialMux` protocol, like in this [sketch](#), the `serial_mux` command will create virtual serial devices.

Start the serial multiplexer on a serial device, e.g., `/dev/ttyUSB0`.

```
$ serial_mux /dev/ttyUSB0
Detected 2 virtual serial ports.
  Virtual serial device 1: /dev/pts/8
  Virtual serial device 2: /dev/pts/9
```

1.3.2 Usage

We can now run the `demo` program that uses two-way communication over the first virtual serial device.

```
$ python demo.py /dev/pts/8
received: 0
received: 1
received: 2
```

Simultaneously, we can look at the log messages that are written to the second virtual serial device.

```
$ picocom -q /dev/pts/9
System time: 3432
System time: 6860
System time: 10290
```


1.4 Command Line Interface

Arduino serial port multiplexer.

```
usage: serial_mux [-h] [-o OUTPUT] [-l LOG] [-b BAUDRATE] [-w WAIT] [-v]
                DEVICE
```

1.4.1 Positional Arguments

DEVICE device

1.4.2 Named Arguments

-o output file
 Default: -

-l log file

-b baud rate
 Default: 9600

-w time before communication starts
 Default: 2

-v show program's version number and exit

Copyright (c) Jeroen F.J. Laros <jlaros@fixedpoint.nl>

1.5 API documentation

1.5.1 SerialMux

class serial_mux.serial_mux.**SerialMux**(*device*, *baudrate*=9600, *wait*=2, *log*=None)

Serial multiplexer.

Parameters

- **device** (*str*) – Device name.
- **baudrate** (*int*) – Baud rate.
- **wait** (*int*) – Time in seconds before communication starts.
- **log** (*Optional[BinaryIO]*) – Open writeable handle to a log file.

send(*port*, *data*)

Send data from a virtual serial device to the serial device.

Parameters

- **port** (*int*) – Virtual serial port.

- **data** (bytes) – Data.

Return type None

update()

Receive serial data and send it to the corresponding virtual serial device.

Return type None

1.5.2 VSerial

class `serial_mux.vserial.VSerial`(*mux*, *port*)

Virtual serial device.

Parameters

- **mux** (object) – Serial multiplexer.
- **port** (int) – Virtual serial port.

receive(*data*)

Receive serial data.

Parameters **data** (bytes) – Data.

Return type None

update()

Send serial data.

Return type None

1.6 Contributors

- Jeroen F.J. Laros <jlarnos@fixedpoint.nl> (Original author, maintainer)

Find out who contributed:

```
git shortlog -s -e
```

PYTHON MODULE INDEX

S

`serial_mux.serial_mux`, 5

`serial_mux.vserial`, 6

INDEX

M

module

 serial_mux.serial_mux, 5

 serial_mux.vserial, 6

R

receive() (*serial_mux.vserial.VSerial method*), 6

S

send() (*serial_mux.serial_mux.SerialMux method*), 5

serial_mux.serial_mux

 module, 5

serial_mux.vserial

 module, 6

SerialMux (*class in serial_mux.serial_mux*), 5

U

update() (*serial_mux.serial_mux.SerialMux method*), 6

update() (*serial_mux.vserial.VSerial method*), 6

V

VSerial (*class in serial_mux.vserial*), 6