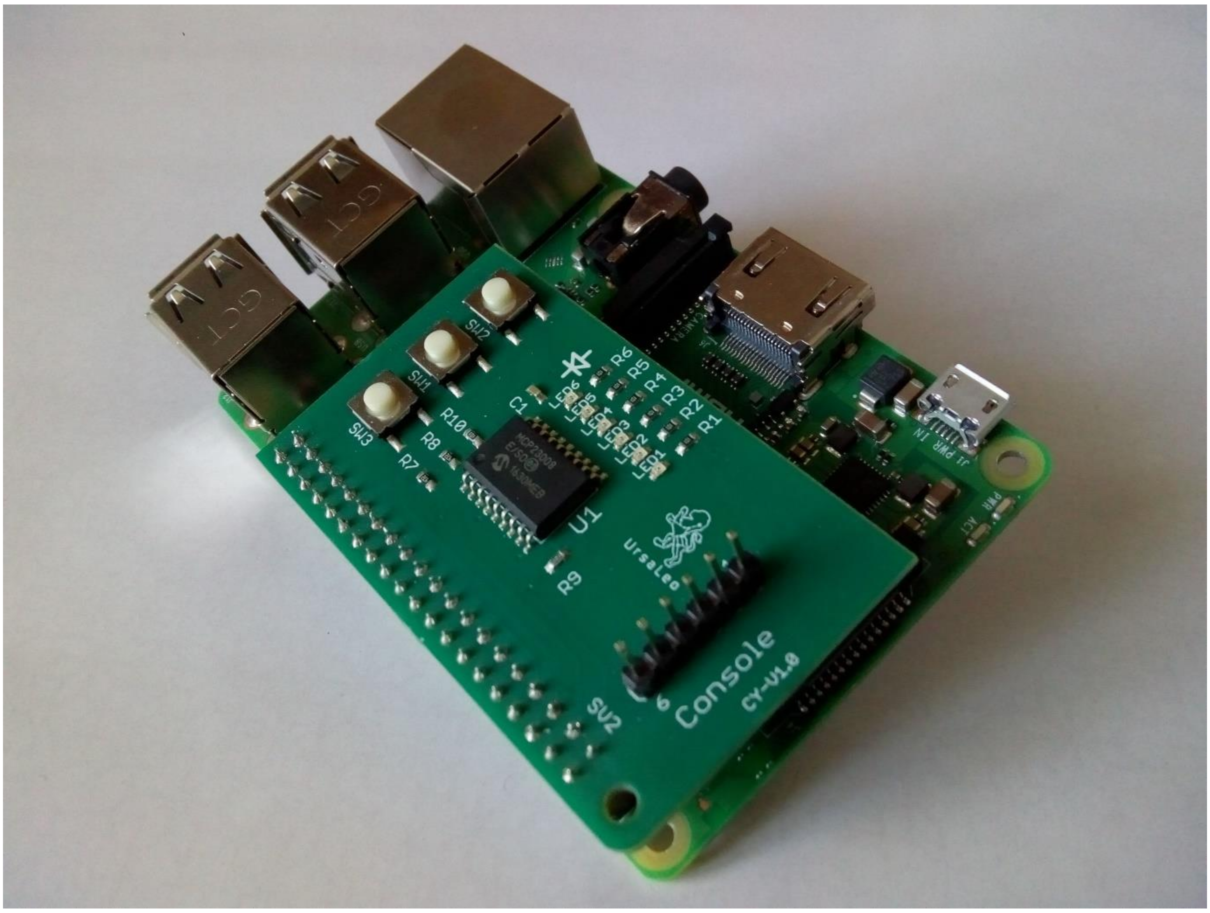
Datasheet
UrsaLeo LED debug board



[Note: Raspberry Pi not included]

Product Details

UrsaLeo LED debug board for Raspberry Pi is designed to add LED status indicators and switched inputs to your projects. It provides debug access to the Pi's serial console for debugging and headless setups via an on board connector.

Based around Microchips MCP23008 8-bit I/O expansion chip for I2C bus, the board comes with a 40-pin connector which fits directly on the Pi's GPIO header. A Python software library is provided along with demo examples which show how to use the board. The board can also be programmed in C/C++ using the information in the Microchip datasheet.

6 LEDs are available that can be individually programmed to react to system events, for example when services start or stop giving a visual indication of status. 1 of the push buttons is dedicated to providing safe system shutdown. The other 2 push buttons can be programmed to trigger events such as starting or stopping processes.

The 6-pin header on the top of the board allows direct access to the Pi's serial console when used with a USB to Serial adapter and host PC (Windows / Mac / Linux). This allows terminal access to the operating system for debugging or configuration, without the need for a monitor and keyboard attached to the Raspberry Pi.

Datasheet

Features

6 programmable LED’s

Safe shutdown button on GPIO4

Serial console header

2 programmable momentary buttons

Compatible with Raspberry Pi 3B / 3B+ / 4B - 40 pin header models

I2C bus interface

Python 3 Library and example code: https://github.com/UrsaLeo/LEDdebug

MCP23008 datasheet: https://www.microchip.com/wwwproducts/en/MCP23008

Technical Details

I2C address: 0x20

I/O expansion: MCP23008 E/SO

Button	**Pin**	**GPIO/I2C Bus**
SW3	GPIO4	RPi Shutdown
SW2	GP7	I2C bus
SW1	GP6	I2C bus

LED	**I2C bus pin**
LED1	GP0
LED2	GP1
LED3	GP2
LED4	GP3
LED5	GP4
LED6	GP5

Console	**GPIO Pin**
Pin 1	GND
Pin 2	N/C
Pin 3	N/C
Pin 4	RXD0 (GPIO15)
Pin 5	TXD0 (GPIO14)
Pin 6	N/C

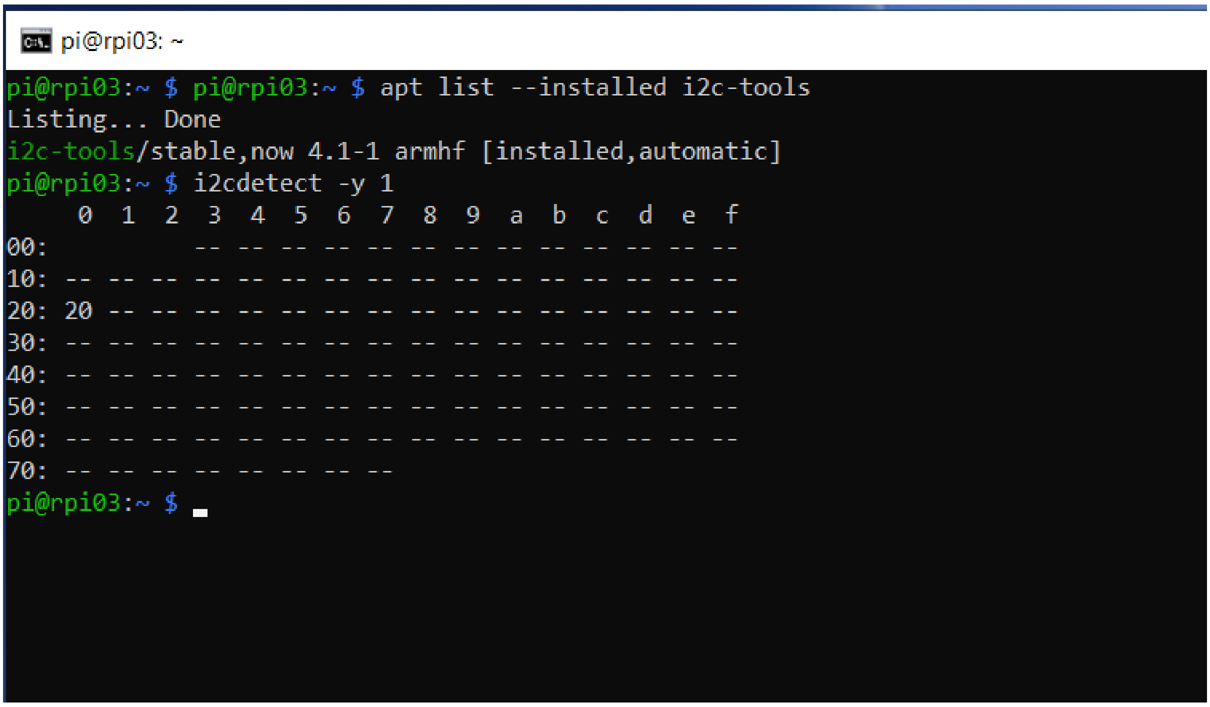
Datasheet

Installation

Align the LED debug board over the Raspberry Pi GPIO header with the push buttons nearest the USB ports and carefully push down until the connector is seated.

Testing

Boot the Raspberry Pi, open a terminal session and check that i2c-tools are installed, then detect any i2c devices – the LED debug board should appear at I2C address 0x20 on Bus 1.

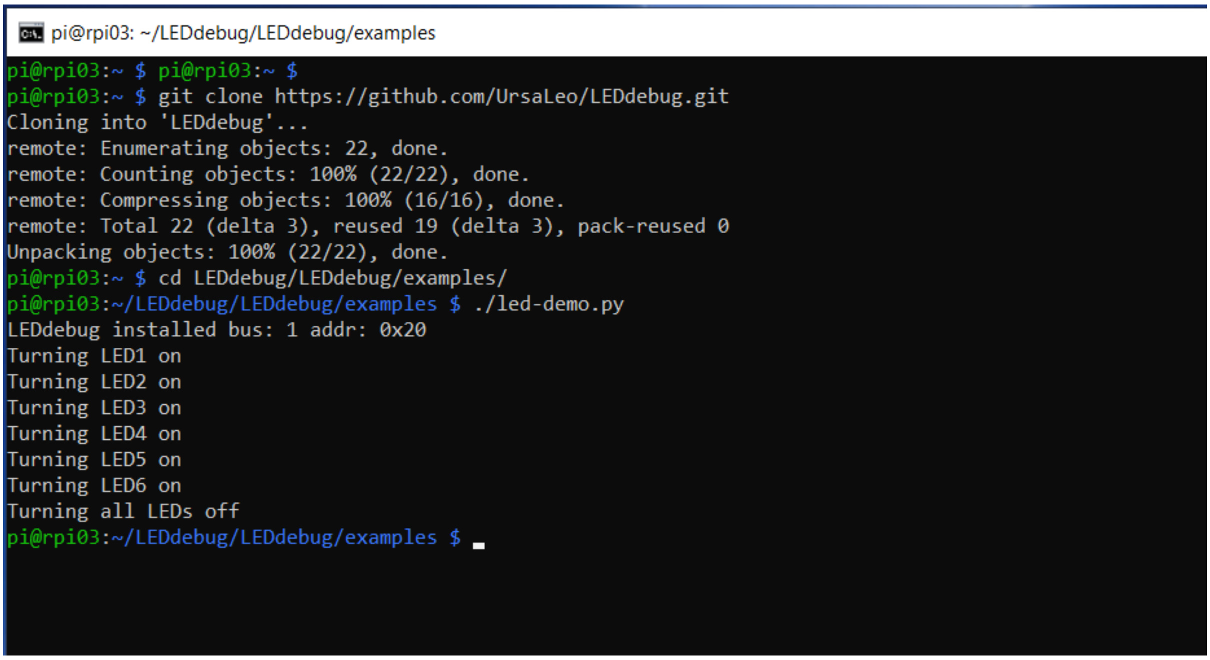


```
pi@rpi03:~ $ pi@rpi03:~ $ apt list --installed i2c-tools
Listing... Done
i2c-tools/stable,now 4.1-1 armhf [installed,automatic]
pi@rpi03:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@rpi03:~ $
```

Clone the Python 3 library from github - https://github.com/UrsaLeo/LEDdebug

Change to the example directory – LEDdebug/LEDdebug/examples

Run the led-demo.py example – the LED’s should go on in sequence, then all turn off



```
pi@rpi03:~ $ pi@rpi03:~ $
pi@rpi03:~ $ git clone https://github.com/UrsaLeo/LEDdebug.git
Cloning into 'LEDdebug'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 22 (delta 3), reused 19 (delta 3), pack-reused 0
Unpacking objects: 100% (22/22), done.
pi@rpi03:~ $ cd LEDdebug/LEDdebug/examples/
pi@rpi03:~/LEDdebug/LEDdebug/examples $ ./led-demo.py
LEDdebug installed bus: 1 addr: 0x20
Turning LED1 on
Turning LED2 on
Turning LED3 on
Turning LED4 on
Turning LED5 on
Turning LED6 on
Turning all LEDs off
pi@rpi03:~/LEDdebug/LEDdebug/examples $
```

Datasheet

Configure Button SW3

In order to use button SW3 to shutdown the Raspberry Pi, edit /boot/config.txt using a text editor (eg vi or nano) and add the following line to the end of the file:

```
dtoverlay=gpio-shutdown,gpio_pin=4
```

Following a reboot, pressing button SW3 will safely shutdown the Raspberry Pi’s operating system avoiding corruption of the SD card.

Note: Do not remove the power supply until the yellow activity LED on the Raspberry Pi has stopped flashing.

Example code

The example directory contains several code samples that can be run on the Raspberry Pi to illustrate how to use the board.

LED Demo – Flashes the LED's then turns them on sequentially.

Button Demo – Flashes the LED's then waits for button presses on SW1 & SW2, turning LED1 & LED2 on and off respectively.

Threaded Button Demo - Flashes LED6 whilst waiting for button presses on SW1 & SW2 in a separate thread.

LED Manager Service - Runs a systemd service to monitor applications and processes.

Console access

Connect a 3.3V USB-to-Serial adapter to the console pin header as detailed in the table above. Open a serial terminal application like “screen” on the attached PC host and enter the following command:

```
sudo screen -L /dev/ttyUSB0 115200
```

When prompted, enter the username and password for your Raspberry Pi. The default username is “pi” and the default password is “raspberry” – if you have changed them, use your own credentials.

Note: Ensure to use a 3.3V USB-to-Serial adapter as 5V versions will damage the Raspberry Pi