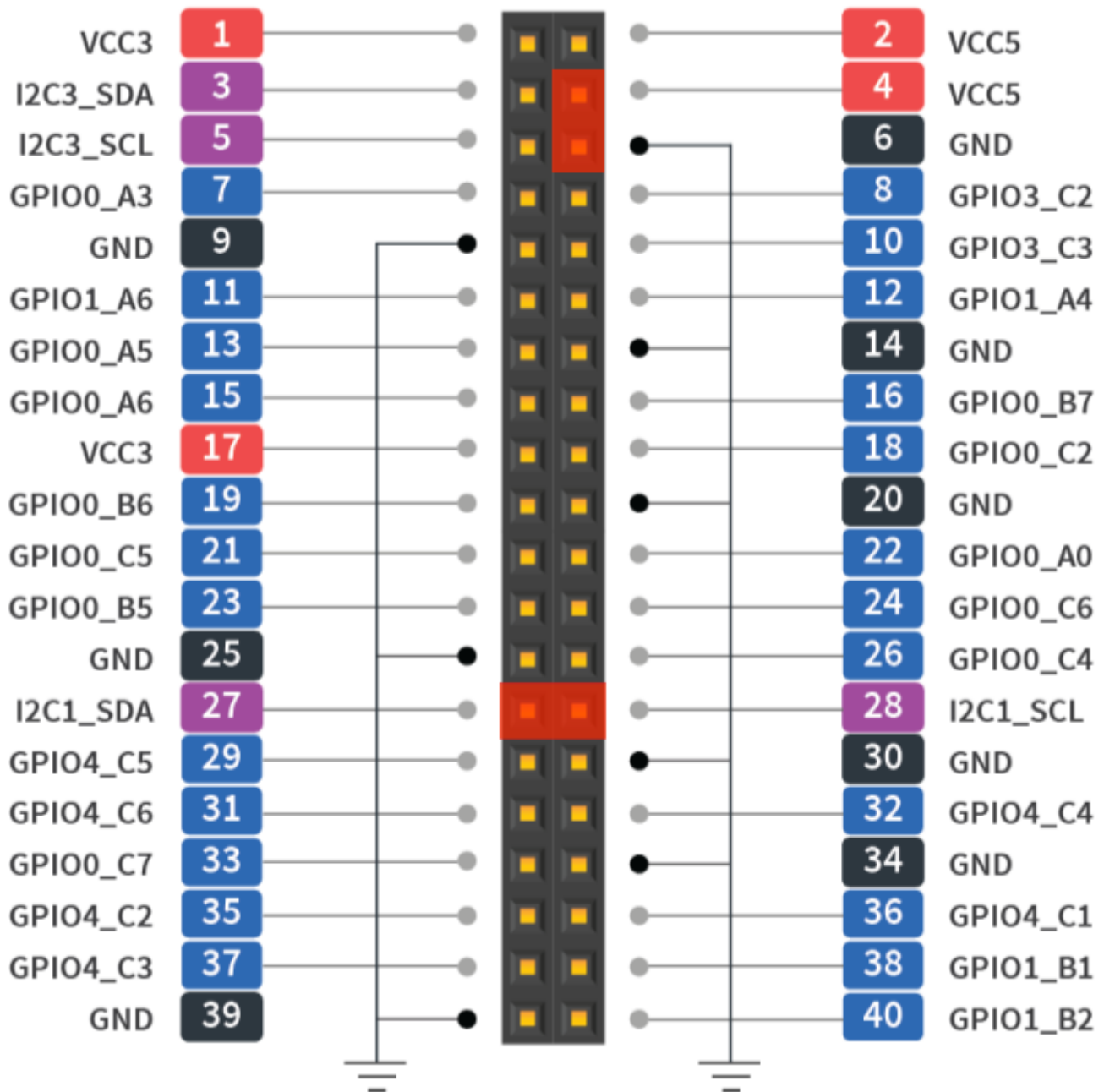


How to use I2C

PCF8574T with 1602A display

As an example we will use PCF8574T with 1602A display. We need to connect URVEPi VCC5, GND, I2C1_SDA and I2C1_SCL to VCC, GND, SDA and SCL pins.

For I2C1 we use Pin27 i Pin28.



Detecting I2C device

After connecting a device with PCF8574T to Pin27 and Pin28 on the URVEPi, you can run the "i2cdetect -y 1" command in the terminal to check if the device is detected on the I2C bus.

This command is used for scanning the I2C bus and detecting devices connected to it. "i2cdetect" is a command-line tool that is available in most Linux distributions.

The "-y" option in the "i2cdetect" command means that the program will scan the I2C bus without confirming each device that is detected.

The digit "1" at the end of the command indicates the I2C bus interface number that will be scanned. In the case of URVEPi, the I2C interface number is usually "1".

```
i2cdetect -y 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00:                - - - - -
10: - - - - -
20: - - - - - 27 - - - - -
30: - - - - -
40: - - - - -
50: - - - - -
60: - - - - -
70: - - - - -
```

In this case I2CDetect have found one device with 0x27 address on URVEPi I2C1 device. Now we can test it, if there are no errors :

```
i2cget -y 1 0x27
0xf7
i2cset -y 1 0x27 0xf0
```

Text on 1602A connected to URVEPi

It's actually very easy. We can use RPLCD library designed for RaspberryPi. We will install it and create lcdtest.py file.

```
apt-get install python3-pip
pip3 install RPLCD
nano lcdtest.py
```

```
# Import LCD library
from RPLCD import i2c
from time import sleep

lcdmode = 'i2c'
cols = 20
rows = 4
```

```
charmap = 'A00'
i2c_expander = 'PCF8574'

# i2cdetect -y 1
address = 0x27
port = 1

# Initialise the LCD
lcd = i2c.CharLCD(i2c_expander, address, port=port, charmap=charmap,
                  cols=cols, rows=rows)

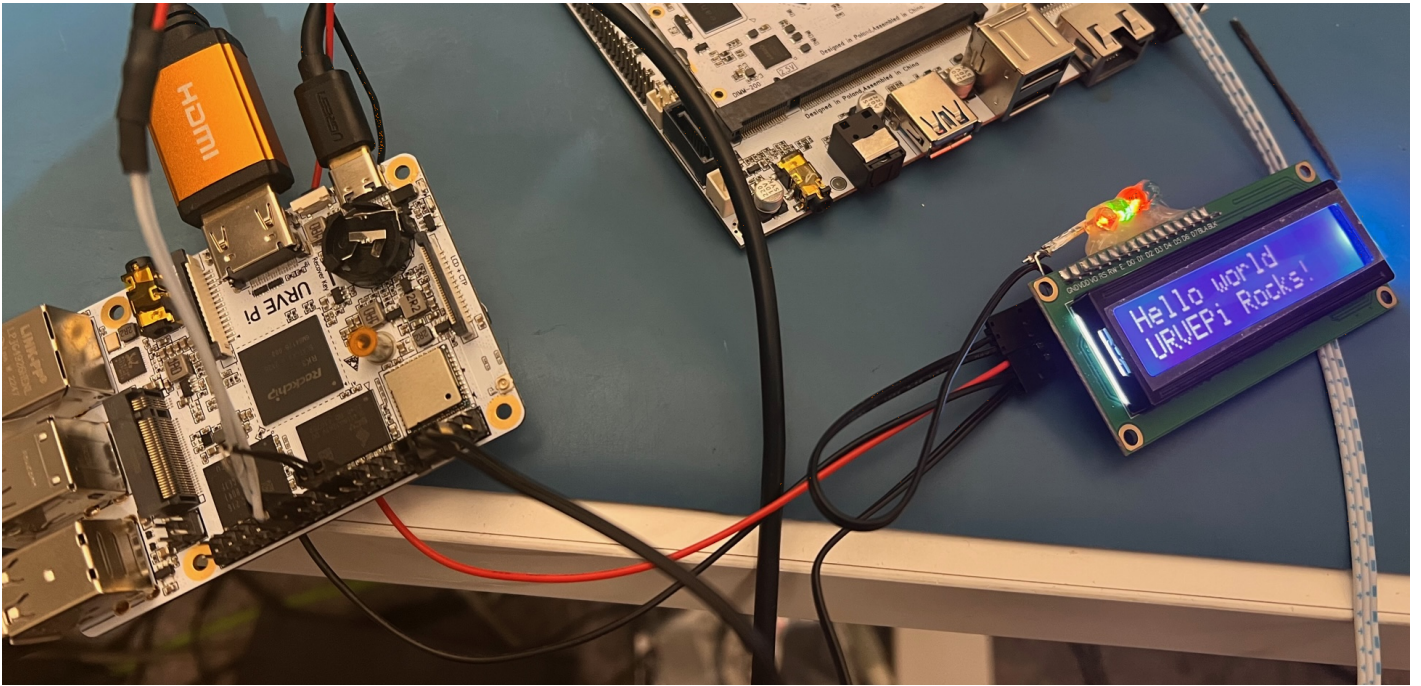
lcd.write_string('Hello world')
lcd.crlf()
lcd.write_string('URVEPi Rocks!')
lcd.crlf()

sleep(5)
# Switch off backlight
lcd.backlight_enabled = False
# Clear the LCD screen
lcd.close(clear=True)
```

and run it with

```
# python3 lcdtest.py
```

Let URVEPi speak for itself:



Text animations on 1602A screen:

```
from RPLCD import i2c
from time import sleep

lcmode = 'i2c'
cols = 16
rows = 2
charmap = 'A00'
i2c_expander = 'PCF8574'
address = 0x27
port = 1
lcd = i2c.CharLCD(i2c_expander, address, port=port, charmap=charmap,
                  cols=cols, rows=rows)
text = 'Hello world! This is a long text.'
while True:
    for i in range(len(text)-15):
        lcd.clear()
        lcd.cursor_pos = (0,0)
        lcd.write_string(text[i:i+16])
        lcd.cursor_pos = (1,0)
        lcd.write_string(text[i+16:i+32])
        sleep(0.5)
```

Revision #3
Created 29 March 2023 20:11:01 by Import
Updated 7 April 2023 06:34:04 by Import