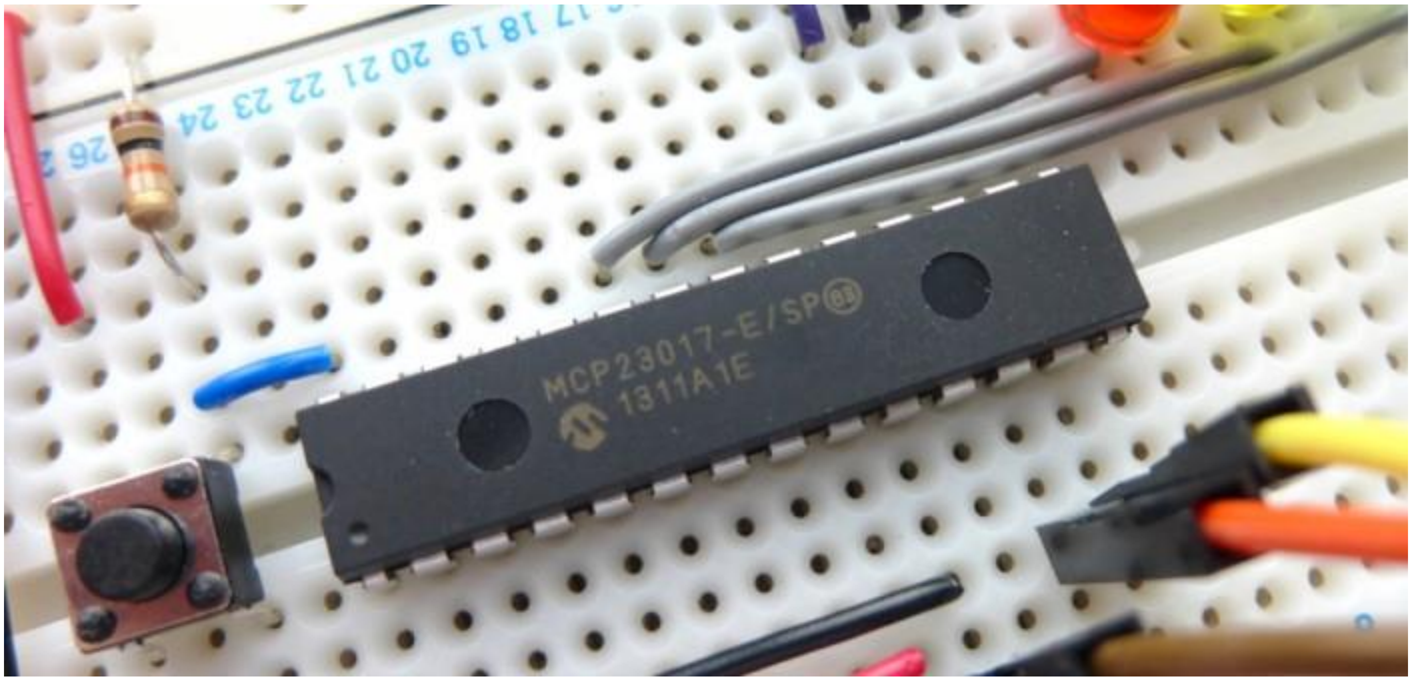
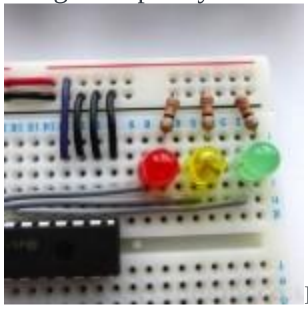
How To Use A MCP23017 I2C Port Expander With The Raspberry Pi – Part 2

4
BY MATT ON JULY 31, 2013I2C, TUTORIALS & HELP

HTTP://WWW.RASPBERRYPI-SPY.CO.UK/2013/07/HOW-TO-USE-A-MCP23017-I2C-PORT-EXPANDER-WITH-THE-RASPBERRY-PI-PART-2/



In How To Use A MCP23017 I2C Port Expander With The Raspberry Pi – Part 1 I explained how to configure your Pi to use I2C so you could connect an MCP23017 16-bit port expander to it. If you've followed that article and got your circuit ready you are only a few steps away from controlling the chip using a simple Python script.



In this post we will concentrate on controlling outputs. We are using the first three pins of the "GPA" set. There are another set of eight pins (GPB0 – GPB7) which can be controlled in the same way with a few code tweaks.

For this sort of project a calculator that can convert between binary, decimal and hex is extremely useful. There are plenty of calculators that can do this available for your operating system, whether you are using a desktop or mobile computer. An easy to usebinary convertor can be found over at our sister site.

Python Script For Outputs

Here is an example script which will count from 1 to 8 setting the first three GPA pins as it goes :

```
import smbus
import time

#bus = smbus.SMBus(0)  # Rev 1 Pi uses 0
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

DEVICE = 0x20 # Device address (A0-A2)
IODIRA = 0x00 # Pin direction register
OLATA  = 0x14 # Register for outputs
GPIOA  = 0x12 # Register for inputs

# Set all GPA pins as outputs by setting
# all bits of IODIRA register to 0
bus.write_byte_data(DEVICE,IODIRA,0x00)

# Set output all 7 output bits to 0
bus.write_byte_data(DEVICE,OLATA,0)

for MyData in range(1,8):
  # Count from 1 to 8 which in binary will count
  # from 001 to 111
  bus.write_byte_data(DEVICE,OLATA,MyData)
  print MyData
  time.sleep(1)

# Set all bits to zero
bus.write_byte_data(DEVICE,OLATA,0)
```

You can download direct to your Pi using :

```
wget https://bitbucket.org/MattHawkinsUK/rpispy-misc/raw/master/mcp23017/mcp2
3017_outputs.py
```

You can run the script using the following command :

```
sudo python mcp23017_outputs.py
```

The script above performs the following actions :

- Imports the smbus and time libraries
- Creates an smbus object named “bus”
- Configures some constants
- Sets all the GPA pins as outputs
- Sets all the GPA pins to zero
- Counts from 1 to 8 and sets the GPA pins to the binary equivalent waiting 1 second between each step
- Finally sets all GPA to zero

What you should see is the LEDs light up in a binary sequence :

```
000,001,010,011,100,101,110,111
```

The most important thing to notice is that when sending data to a register you have to set all 8 bits in one go. So to configure all the GPA pins as outputs you send 0000000 to the IODIRA register. To set them all as inputs you need to send 1111111 binary, or 255 decimal or 0xFF hex. To set GPA0-3 as outputs and GPA4-8 as inputs you need to set 1111000 binary, 240 decimal or 0xF0 hex.

It’s the same story for setting the pins of outputs. If you’ve set the GPA pins as outputs and you want to set GPA1 high and all the other pins low you need to send 00000010 to the OLATA register. If you want to set GPA2 and GPA7 high you would send 10000100.

Visit the How To Use A MCP23017 I2C Port Expander With The Raspberry Pi – Part 3 page which explains how to configure pins as inputs and read their state.
SHARE.