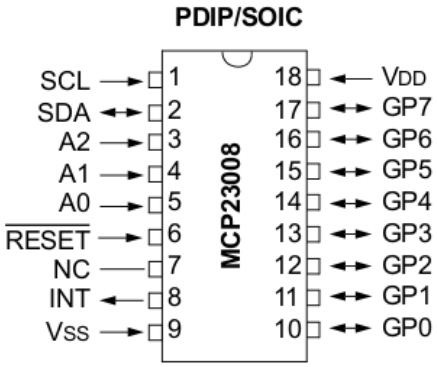
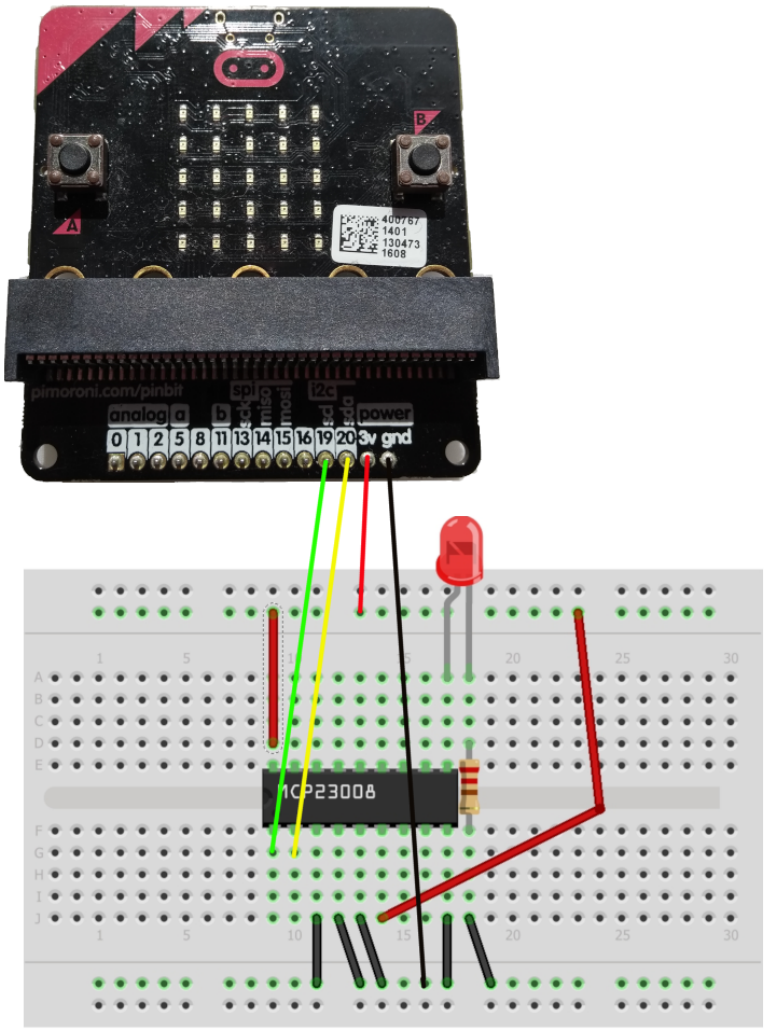
Using a MCP23008 with the Microbit.

This PDF will show you how to add 8 extra I/O pins to the Microbit using an 8 port MCP23008 chip, that communicates over I2C 2 wire protocol using pins 19 & 20 along with 3V and GND.

You can buy the MCP23008 chips on eBay for around £2.25 including postage.



The pins labelled GP0 to GP7 can be configured as inputs or outputs. An LED is attached to GP0 in picture below. The resistor is a 220 ohm. We're using a Pimoroni breakout board.



We'll be using MicroPython and the Mu editor.

This micropython code will scan the I2C bus , find devices and print their port number, if you press button A.

```
from microbit import *

start = 0x08
end = 0x77

while True:
    display.show(Image.ARROW_W)
    if button_a.was_pressed():
        display.show(Image.MEH)
        print("Scanning I2C bus...")
        for i in range(start, end + 1):
            try:
                i2c.read(i, 1)
            except OSError:
                pass
            else:
                print("Found:  [%s]" % hex(i))
        print("Scanning complete")
        print("Magnetometer [0x0e] Accelerometer [0x1d]")
    sleep(10)
```

The onboard magnetometer is at [0x0e] and the accererometer at [0x1d] - If you have the MCP23008 chip connected correctly you should also see "Found: [0x20]" in the scan results.

This code will blink the LED:

```
from microbit import *
i2c.write(0x20, b'\x00\x00') # set all pins as outputs

while True:
    i2c.write(0x20, b'\x09\x01')
    sleep(500)
    i2c.write(0x20, b'\x09\x00')
    sleep(500)
```

The 0x20 refers to the I2C bus address of the MCP23008 chip. The x09 refers to the GPIO pins. Then we send a value between 00 and FF (hexadecimal, 0 to 255 decimal) to the chip. 00 would turn off all 8 pins. FF would turn on all 8 pins. If you think of the pins with LEDs connected as 8 digit binary 00000000, then to turn on the LED on the right we send 00000001 (x01 in hex) and to turn on the LED on the left we'd send 10000000 (x80 in hex).

Converter here: https://www.binaryhexconverter.com/binary-to-hex-converter

The table on the next page shows all the functions you can send to the MCP23008. You can download the full PDF datasheet for the chip from https://www.mouser.co.uk/datasheet/2/268/21919b-65915.pdf

TABLE 1-3: CONFIGURATION AND CONTROL REGISTERS

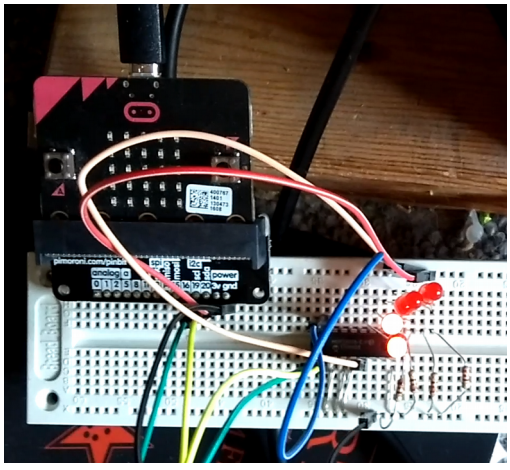
Register Name	Address (hex)	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	POR/RST value
IODIR	00	IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	1111 1111
IPOL	01	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0	0000 0000
GPINTEN	02	GPINT7	GPINT6	GPINT5	GPINT4	GPINT3	GPINT2	GPINT1	GPINT0	0000 0000
DEFVAL	03	DEF7	DEF6	DEF5	DEF4	DEF3	DEF2	DEF1	DEF0	0000 0000
INTCON	04	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	0000 0000
IOCON	05	—	—	SREAD	DISSLW	HAEN*	ODR	INTPOL	—	--00 000-
GPPU	06	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0000 0000
INTF	07	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INTO	0000 0000
INTCAP	08	ICP7	ICP6	ICP5	ICP4	ICP3	ICP2	ICP1	ICP0	0000 0000
GPIO	09	GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0	0000 0000
OLAT	0A	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	0000 0000

* Not used on the MCP23008.

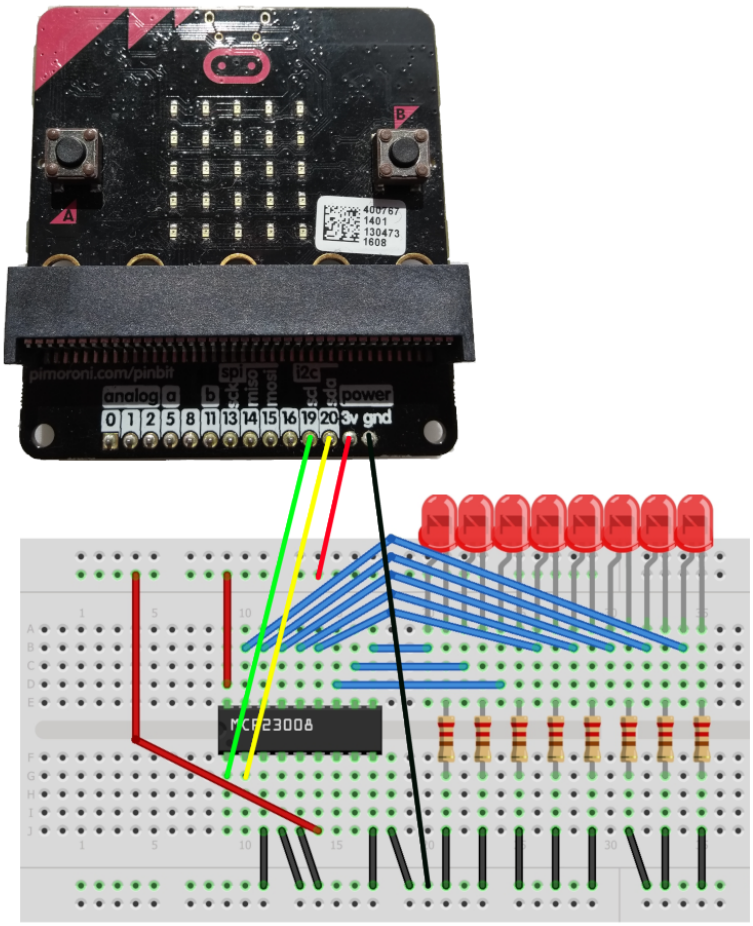
So now we have managed to turn a single LED on and off. Let's now look at some code to cycle through 4 LEDs connected to the pins GP0, GP1, GP2 and GP3 via resistors to GND.

```
# 0x20 is I2C bus address of MCP23008 chip. x09 is GPIO register
# Sending FF to x09 turns on all pins. Sending 00 = all off.
# Sending 1/2/4/8/10/20/40/80 turns on pins 0 through 7
from microbit import *
i2c.write(0x20, b'\x00\x00') # set all pins as outputs
sequence = ['0x01','0x02','0x04','0x08','0x04','0x02']
s = b'\x09'
while True:
    for i in range(0, 6):
        result = bytes([int(sequence[i])])
        s += result
        print(s)
        i2c.write(0x20, s)
        sleep(100)
        i2c.write(0x20, b'\x09\x00')
        s = b'\x09'
```

The code above does a basic "Knight Rider" style backwards and forwards LED cycle, using 4 LEDs.



We can also do it with 8 LEDs.




```
# 0x20 is I2C bus address of MCP23008 chip. x09 is GPIO register
# Sending FF to x09 turns on all pins. Sending 00 = all off.
# Sending 1/2/4/8/10/20/40/80 turns on pins 0 through 7

from microbit import *
i2c.write(0x20, b'\x00\x00') # set all pins as outputs
sequence =
['0x01','0x02','0x04','0x08','0x10','0x20','0x40','0x80','0x40','0
x20','0x10','0x08','0x04','0x02']
s = b'\x09'

while True:
    for i in range(0, 14):
        result = bytes([int(sequence[i])])
        s += result
        print(s)
        i2c.write(0x20, s)
        sleep(100)
        i2c.write(0x20, b'\x09\x00')
        s = b'\x09'
```

Mu

New Load Save Flash Files Repl Zoom-in Zoom-out Theme Check Help Quit

pir-test.py | keypad2.py | kptest1.py | i2cscan.py | i2c-test1.py | i2c-test2.py | keypad.py | i2c-test3.py *

```
# 0x20 is I2C bus address of MCP23008 chip. x09 is GPIO register
# Sending FF to x09 turns on all pins. Sending 00 = all off.
# Sending 1/2/4/8/10/20/40/80 turns on pins 0 through 7

from microbit import *
i2c.write(0x20, b'\x00\x00') # set all pins as outputs
sequence = ['0x01','0x02','0x04','0x08','0x10','0x20','0x40','0x80','0x40','0x20','0x10','0x08','0x04','0x02']
s = b'\x09'

while True:
    for i in range(0, 14):
        result = bytes([int(sequence[i])])
        s += result
        print(s)
        i2c.write(0x20, s)
        sleep(100)
        i2c.write(0x20, b'\x09\x00')
        s = b'\x09'
```