	เฉลยใบงานที่ 7	ครั้งที่ 13
	หน่วยที่ GPIO การใช้ I^2C bus	รวม 9 ชั่วโมง
เรื่องการใช้งานGPIO ส่วนการใช้ I^2C bus ของ RPi ติดต่อกับ MCP23017		จำนวน 90 นาที

ลำดับขั้นการทดลอง

1. โปรแกรมทดลองกำหนดให้ขา GPA0-GPA7 เป็นเอาต์พุต ดังนี้

```
import smbus                                    สั่งอิมพอร์ตไลบรารี smbus และ time
import time                                     กำหนดค่าคงที่ เพื่ออ้างถึงรีจิสเตอร์ของ MCP23017
bus = smbus.SMBus(1)                            # สร้าง smbus object ชื่อ “bus”
DEVICE = 0x20                                   # Device address (A0-A2)
IODIRA = 0x00                                   # Pin direction register สั่งให้ขาของ GPA ทุกขาเป็นเอาต์พุต
OLATA  = 0x14                                   # Register for outputs
GPIOA  = 0x12                                   # Register for inputs
                                                # Set all GPA pins as outputs
bus.write_byte_data(DEVICE,IODIRA,0x00)         #by setting all bits of IODIRA register to 0
bus.write_byte_data(DEVICE,OLATA,0)             # Set output all 7 output bits to 0
                                                # Count from 1 to 8 which in binary will count
for MyData in range(1,8):                       # from 001 to 111
        bus.write_byte_data(DEVICE,OLATA,MyData)    สั่งนับเลขจาก 1-7 (0b001 ถึง 0b111)
        print(MyData)                               และกำหนดให้ขาของ GPA มีค่าตามตัวเลขที่กำหนด
        time.sleep(1)                               หยุดรอ 1 วินาที
        bus.write_byte_data(DEVICE,OLATA,0)         # Set all bits to zero เมื่อนับถึง 7 แล้ว ให้ขา GPA ทุกขาเป็น “0”
```

2. โปรแกรมใช้งานอินพุตเอาต์พุตของ MCP23017

```
import smbus
import time
bus = smbus.SMBus(1)                            # Rev 2 Pi uses 1
DEVICE = 0x20                                   # Device address (A0-A2)
IODIRA = 0x00                                   # Pin direction register
IODIRB = 0x01                                   # Pin direction register
OLATB  = 0x15                                   # Output Latch Register port A
GPIOA  = 0x12                                   # Register for inputs
GPPUA  = 0x0C                                   # Register for pull up resistor PortA
bus.write_byte_data(DEVICE,IODIRA,0xff)         # Set first 6 GPA pins as outputs and
bus.write_byte_data(DEVICE,GPPUA,0xff)          # last one as input.
bus.write_byte_data(DEVICE,IODIRB,0x00)
while True:                                     # Loop until user presses CTRL-C
        MySwitch = bus.read_byte_data(DEVICE,GPIOA)  # Read state of GPIOA register
        if(MySwitch & 0b10000000 == 0b00000000):     #check status GPA7
            bus.write_byte_data(DEVICE,OLATB,0b00000001)
            print("Switch was pressed!")
            time.sleep(0.5)
        else:
```

```
            bus.write_byte_data(DEVICE,OLATB,0b00000000)
```

3. ผลการทดลอง

การทำงานของโปรแกรม mcp23017_in_out.py เมื่อกดสวิตช์ที่ต่อกับขา GPA7, LED ที่ต่อกับ GPB0 จะติด หน้าจอขึ้นแสดงข้อความ

.....

.....

.....

4. โปรแกรมใช้งานอ่านค่าอุณหภูมิของ DS1621

```
import time
import smbus
bus = smbus.SMBus(1)
address = 0x48                                  # Address with all the bit pins pulled to ground [A0 - A2]
                                                # Temperature conversion commands
CMD_READ_TEMP = 0xAA                            # Read last converted temperature value from the register
CMD_READ_COUNT = 0xA8                           # Reads value of Count_Remain
CMD_READ_SLOPE = 0xA9                           # Reads value of the Count_Per_C
CMD_START_CONVERT = 0xEE                        # Initiates temperature conversion
CMD_STOP_CONVERT = 0x22                          # Halts temperature conversion
                                                # Thermostat commands
CMD_ACCESS_HIGH = 0xA1                          # Reads or writes high temperature limit value into the TH register
CMD_ACCESS_LOW = 0xA2                           # Reads or writes high temperature limit value into the LH register
CMD_ACCESS_CONFIG = 0xAC                        # Reads or writes configuration data to configuration register
bus.write_byte(address, CMD_START_CONVERT)      # Not sure yet if this line is needed to start the process??
bus.write_byte_data(address, CMD_ACCESS_LOW, 0x00)   # Writing the lower threshold on thermostat at 0x00, hopefully 0
                                                #eg C ??
bus.write_byte_data(address, CMD_ACCESS_HIGH, 0x64) # Writing the higher threshold on thermostat at 0x64, hopefully
                                                #100 dec C??
for i in range (1,60):
    print("\n\nCount number",i,"\n")
     print("CMD_READ_TEMP and bus.read_byte_data returns : ",bus.read_byte_data(address,CMD_READ_TEMP))
    print("CMD_READ_TEMP and bus.read_byte returns : ",bus.read_byte(address))
    print("CMD_READ_TEMP and bus.read_word_data returns : ",bus.read_word_data(address,CMD_READ_TEMP))
    print("CMD_READ_TEMP and bus.read_i2c_block_data returns : ",bus.read_i2c_block_data(address,CMD_READ_TEMP))
    time.sleep(5)
    i += 1
bus.write_byte(address, CMD_STOP_CONVERT)
print("\n\n\nPress Return to continue.....")
```

5. ผลการทดลองแสดงดังรูป

.....

คำถาม

1. ไอซี MCP23017 สามารถ ขยายพอร์ตได้ถึง 16 บิต โดยแต่ละพอร์ตรับหรือจ่ายกระแสได้สูงถึง 25 mA (รวมกันไม่เกิน 125 mA) มีขาแอดเดรส A0,A1,A2 กำหนดแอดเดรสได้ 8 หมายเลข (ต่อร่วมกันได้สูงสุด 8 ตัว)

2. การเปิดใช้งานพอร์ต I^2C ของ RPi

    1. ตรวจสอบโมดูลของระบบ linux ด้วย lsmod

    2. ติดตั้งไลบรารี python-smbus, i2c-tools

3. การทดสอบการติดต่อกับอุปกรณ์ I^2C

    ใช้โปรแกรม i2cdetect ตรวจสอบ ด้วยคำสั่ง sudo I2cdetect - 1 ถ้าต่ออุปกรณ์ถูกต้องจะพบอุปกรณ์อยู่ที่แอดเดรส 0x20

4. MCP23017 byte mode register address นั้น

    00h หมายถึง I/O Direction Register A ‘0’ Out ‘1’ Input

    14h หมายถึง Output Latch Register A เขียนค่าไปยังพอร์ต A (มีผลเฉพาะขาที่เป็นเอาต์พุต)