

## 01076001 Introduction to Computer Engineering

การทดลองนี้จะใช้ I2C EEPROM ภายนอกเพื่อเชื่อมต่อกับ Arduino ไว้สำหรับจัดเก็บข้อมูลแบบถาวร เมื่อมีการทำงานจัดเก็บข้อมูลที่มีขนาดใหญ่ จำเป็นต้องเก็บข้อมูลเพิ่มเติมในหน่วยความจำภายนอก ตัวอย่างเช่น I<sup>2</sup>C EEPROM เบอร์ 24C256 จะมีหน่วยความจำ 256 Kbit หรือมีหน่วยความจำเท่ากับ  $256 \times 1024 = 262,144$  บิต เมื่อแปลงเป็นไบต์โดยการหาร 8 จะได้เท่ากับ 32,768 ไบต์ ในการทดลองนี้จะใช้ EEPROM เบอร์ 24C16 แต่จะใช้ EEPROM เบอร์ 24C256 เป็นโปรแกรมตัวอย่างในการเขียนโปรแกรม เพื่อให้สามารถนำไปประยุกต์ใช้ใน EEPROM เบอร์อื่นๆได้ ซึ่งจะมีความแตกต่างกันที่ PIN และขนาดของหน่วยความจำ รายละเอียดเพิ่มเติมให้เปิดดูได้จากเอกสารของไอซี AT24C256\_DataSheet และ M24C16\_DataSheet

ไอซีที่ใช้ในการทดลองจะเป็นแบบ DIP 8 ขา มีขา 8 เป็นแหล่งจ่ายไฟ Vcc ขา 4 เป็นขากราวด์ GND ขา 7 เป็นขาการป้องกันการเขียน WP ช่วยให้เราสามารถควบคุมป้องกันการเขียนข้อมูลลงใน EEPROM ได้ ส่วนการอ่านจะเปิดใช้งานเสมอ การทดลองให้ต่อขาที่ลงกราวด์ ขา 5 เป็น SDA ขา 6 เป็น SCL และสามขาที่เหลือ 1,2,3 เป็นขาเลือก Address ที่จะนำไปใช้งาน

การเขียนโปรแกรมเพื่อใช้งาน I2C EEPROM โดยการควบคุมสัญญาณแต่ละเส้นจากบอร์ด Arduino จะให้ใช้คำสั่งที่ได้จาก Library สำหรับสื่อสารแบบ I2C ที่ชื่อ Wire เท่านั้น โดยเลือก Address ของอุปกรณ์ให้ตรงกับฮาร์ดแวร์ แล้วใช้คำสั่งใน Wire.h จัดการรับส่งค่า

1. เริ่มจากการหา Address ที่ใช้สำหรับ I2C EEPROM ให้เขียนโปรแกรมเพื่อทดสอบหา Address ดังนี้

```
/*
Use the I2C bus with EEPROM device
EEPROM 8 (Vcc) to Vcc
EEPROM 4 (GND) to GND
EEPROM 5 (SDA) to Arduino Analog Pin 4(SDA)
EEPROM 6 (SCL) to Arduino Analog Pin 5(SCL)
EEPROM 7 (WP) to GND

/* Memory length in bytes
24C01 = 128
24C02 = 256
24C04 = 512
24C08 = 1024
24C16 = 2048
24C64 = 8192
24C128 = 16384
24C256 = 32768
*/
#define memorylength 512
#include <Wire.h>
byte deviceAddress; // Address of EEPROM chip
unsigned int eepromAddress;

void setup()
{
    byte eepromData;
    byte i;

    Serial.begin(9600); // Setup serial for debug
    Wire.begin(); // Start I2C bus

    for(i = 1; i < 127; i++) // sets the value (range from 1 to 127)
    {
        Wire.beginTransmission(i); // transmit to device
        if (Wire.endTransmission() == 0) // I2C devices found
        {
            deviceAddress = i; // Print Device Address
            Serial.print("I2C Device Address: "); // print as an ASCII-encoded hexa);
            Serial.println(deviceAddress, HEX);
        }
        break;
    }
    // จริงแค่ค่าแรก ค่าหลังเป็นmemory เลยต้องมี break; ต่อท้าย
```

```

    }
}

void loop()
{
}

```

- การทดลองให้ต่อ I2C EEPROM เข้ากับบอร์ด Arduino ผ่านการเชื่อมต่อทางพอร์ต I2C
- ให้ทดลองป้อนโปรแกรมเพื่อตรวจหา I2C EEPROM ที่ต่ออยู่กับพอร์ต I2C และบันทึกค่าที่ได้

*I2C Device Address : 50*

- ให้แก้ไขโปรแกรมข้อ 1 โดยเพิ่มในส่วนของ void setup() เป็นโปรแกรมการส่งค่าข้อมูล 1 byte ไปที่พอร์ต I2C เพื่อบันทึกค่าลงใน EEPROM โดยค่าที่ส่งไปมีค่าเท่ากับ 128

```

epromAddress = 0;
epromData = 128;
writeEEPROM_byte(deviceAddress, eepromAddress, eepromData); // write to EEPROM

```

- ให้เขียนฟังก์ชันเพิ่มจากข้อ 4 โดยแก้ไขจากตัวอย่างฟังก์ชันการส่งค่าข้อมูล 1 byte ไปที่ EEPROM 24C256

```

void writeEEPROM_byte(int device, unsigned int address, byte data )
{
    Wire.beginTransmission(device); //เอา Address มา or กับ device
    Wire.write((int)(address >> 8)); // MSB
    Wire.write((int)(address & 0xFF)); // LSB
    Wire.write(data);
    Wire.endTransmission();
    delay(10);
}

```

- ให้แก้ไขโปรแกรมข้อ 4 โดยเพิ่มโปรแกรมการรับค่าขนาด 1 byte จาก EEPROM และพิมพ์ผลค่าที่ส่งออกไปทาง Serial Monitor ดังนี้

```

Serial.print(readEEPROM_byte(deviceAddress, eepromAddress), HEX);

```

- ให้เขียนฟังก์ชันเพิ่มจากข้อ 6 โดยแก้ไขจากตัวอย่างฟังก์ชันการรับค่าข้อมูล 1 byte จาก EEPROM 24C256

```

byte readEEPROM_byte(int device, unsigned int address )
{
    byte rdata = 0;
    Wire.beginTransmission(device);
    Wire.write((int)(address >> 8)); // MSB
    Wire.write((int)(address & 0xFF)); // LSB
    Wire.endTransmission();
    Wire.requestFrom(device, 1);
    if (Wire.available())
        rdata = Wire.read();
    return rdata;
}

```

- ให้อธิบายผลลัพธ์ที่ได้ของ I2C EEPROM จากการรับส่งข้อมูลผ่านทางพอร์ต I2C *ผลลัพธ์ คือ 50*  
*เก็บ address ที่ได้จะค่า 8 bit แล้วเอาค่า 16 bit แล้วลบ 8 bit ออกมาจะได้ 8 bit แล้วเอาค่า 8 bit นั้นไปคูณกับ 256 >> 8 bit แล้วนำมา & กับ 0xFF ( 1111 1111 ) 8 bit แล้วเอา address เดิม ไปเปลี่ยนเป็นค่าจาก 50*

- ให้แก้ไขโปรแกรมข้อ 6 โดยเพิ่มโปรแกรมการส่งค่าข้อมูลที่เป็น Array ไปที่พอร์ต I2C ของ EEPROM ดังนี้

```

char arr[20] = {
    0x41, 0x42, 0x43, 0x44, 0x45, 0x61, 0x62, 0x63, 0x64, 0x65,
    0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x30,
};
writeEEPROM_page(deviceAddress, eepromAddress + 1, (byte *)arr, sizeof(arr));

```

10. ให้เขียนฟังก์ชันเพิ่มจากข้อ 9 โดยแก้ไขจากตัวอย่างฟังก์ชันการส่งชุดข้อมูลไปที่ EEPROM 24C256

```
void writeEEPROM_page(int device, unsigned int address, byte* buffer, byte length)
{
    byte i;
    Wire.beginTransmission(device);
    Wire.write((int)(address >> 8));           // MSB
    Wire.write((int)(address & 0xFF));         // LSB
    for (i = 0; i < length; i++)
        Wire.write(buffer[i]);
    Wire.endTransmission();
    delay(10);
}
```

11. ให้แก้ไขโปรแกรมข้อ 9 โดยเพิ่มโปรแกรมการส่งค่าข้อมูลที่เป็น string ไปที่พอร์ต I2C ของ EEPROM ดังนี้

```
char str[] = "CE-KMITL";
writeEEPROM_page(deviceAddress, eepromAddress + 21, (byte *)str, sizeof(str));
```

12. จากโปรแกรมการส่งค่าข้อมูลทั้งหมดไปที่ EEPROM ผ่านทางพอร์ต I2C หลังจากส่งข้อมูลไปแล้ว ยังไม่ได้มีการแสดงผล ดังนั้นให้แก้ไขโปรแกรมโดยเพิ่มในส่วนของ void loop() ให้อ่านค่าข้อมูลใน EEPROM และพิมพ์ผลลัพธ์ของค่าข้อมูลที่ส่งไปทั้งหมดออกไปทาง Serial Monitor ดังนี้

```
byte ch;

Serial.println(" ");
eepromAddress = 0;
ch = readEEPROM_byte(deviceAddress, eepromAddress); // first address
                                                    // access the first address from the memory

while (ch!=0)
{
    Serial.print((char) ch); // print content to serial port
    eepromAddress++;         // increase address
    ch = readEEPROM_byte(deviceAddress, eepromAddress); // access an address from the memory
}
delay(2000);
```

13. ให้อธิบายผลลัพธ์ที่ได้ของ EEPROM จากค่าที่รับมาทาง Serial Monitor ผ่านทางพอร์ต I2C .....

ผลลัพธ์ : A B C D E a b c d e 1 2 3 4 5 6

7. 8. 9. 0 C E - K M I T L

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

16 page

14. จากการทดลองจะเห็นได้ว่าในส่วนของหน่วยความจำที่ยังไม่ได้เขียนลงไป ไม่มีการแสดงผล ดังนั้นให้แก้ไขโปรแกรมข้อ 12 โดยเพิ่มโปรแกรมเพื่อให้ dump ค่าใน memory ของ EEPROM ออกมาแสดงผลทั้งหมด

```
dumpEEPROM(deviceAddress, 0, memorylength);
delay(2000);
```

โดยฟังก์ชันที่ถูกเรียกใช้เป็นอย่างนี้

```
void dumpEEPROM(byte device, unsigned address, unsigned length)
```

```

{
    unsigned startAddress = address;
    unsigned stopAddress = address + length;

    for (unsigned i = startAddress; i < stopAddress; i += 16)
    {
        char buffer[16]; // page of EEPROM
        char temp[4];
        sprintf(temp, "%04x: ", i); // print address
        Serial.print(temp);
        readEEPROM_page(device, i, buffer, 16);

        for (int j = 0; j < 16; j++)
        {
            sprintf(temp, "%02x ", byte(buffer[j])); // print data (hexa)
            Serial.print(temp);
        }
        Serial.print(" ");
        for (int j = 0; j < 16; j++) // print data (ASCII)
        {
            if (isprint(buffer[j]))
                Serial.print(buffer[j]);
            else
                Serial.print('.');
        }
        Serial.println(" ");
    }
}

```

15. ให้เขียนฟังก์ชันเพิ่มจากข้อ 14 โดยแก้ไขจากตัวอย่างฟังก์ชันการรับชุดของข้อมูลจาก EEPROM 24C256

```

byte readEEPROM_page(int device, unsigned int address, byte *buffer, int length )
{
    byte i;
    Wire.beginTransmission(device);
    Wire.write((int)(address >> 8)); // MSB
    Wire.write((int)(address & 0xFF)); // LSB
    Wire.endTransmission();
    Wire.requestFrom(device, length);
    for ( i = 0; i < length; i++ )
        if (Wire.available())
            buffer[i] = Wire.read();
}

```

16. ให้อธิบายผลลัพธ์ที่ได้ของ EEPROM จากการ dump ค่าที่แสดงผลทาง Serial Monitor .....

```

.....
0000 : 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50
.....

```

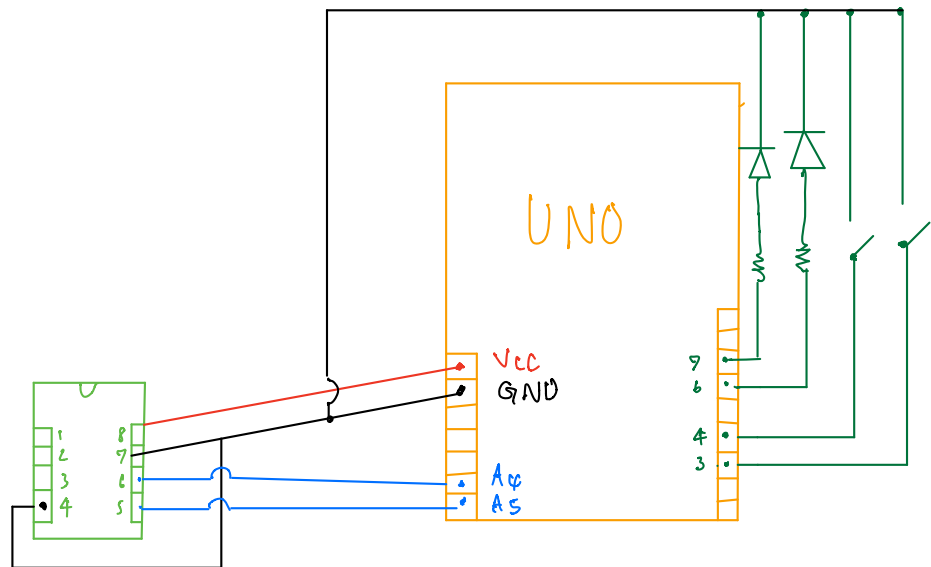
```

0010 : 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60
.....
.....
.....
.....
.....

```

17. ให้เขียนโปรแกรมที่ทำหน้าที่ copy ข้อมูลของ EEPROM โดยใช้สวิตช์ 2 ตัว ตัวแรกเมื่อกดจะทำหน้าที่อ่านแล้วให้แสดงผล dump ข้อมูลออกที่หน้าจอ ตัวที่สองทำหน้าที่เขียนแล้วให้ทำการ verify ตรวจสอบข้อมูลว่าสามารถบันทึกข้อมูลได้ครบทุกบิตหรือไม่ โดยให้แสดงผลที่ LED สีเขียวคือ copy ได้สำเร็จ และแดงคือเกิด error

18. จากข้อ 17 ให้วาดรูปของวงจรทั้งหมดที่ใช้



19. ให้อธิบายความแตกต่างของโปรแกรมที่ใช้กับ EPROM เบอร์ 24C16 กับ 24C256 .....

Memory length in bytes ที่สามารถเก็บได้ต่างกัน

20. ให้เขียนโปรแกรมที่ทำหน้าที่บันทึกข้อมูลลงใน EEPROM โดยให้ชื่อศึกษาอยู่ที่ Address 200H , รหัสนักศึกษาที่ Address 400H และรายละเอียดของนักศึกษาอยู่ที่ Address 600H แล้วสั่งงานให้ LED บนบอร์ด Arduino กระพริบตามค่ารหัสประจำตัวของนักศึกษา