

ใบงานแบบฝึกหัด

การเขียนโปรแกรม

ควบคุมโมดูลแสดงผล LCD I2C



วิชาการเขียนโปรแกรม

ระดับชั้นมัธยมศึกษาปีที่ 3

กลุ่มสาระการเรียนรู้วิทยาศาสตร์และเทคโนโลยี

ครูสิริลักษณ์ ยศพุ่มกุล

โรงเรียนวัดเขียนเขต

ใบงานการทดลอง

เรื่อง โปรแกรมควบคุมโมดูล LCD I2C

จุดประสงค์การเรียนรู้

1. นักเรียนสามารถอธิบายหน้าที่และการวิธีการใช้งานโมดูลแสดงผล LCD ได้ (K)
2. นักเรียนสามารถต่อวงจรและเขียนโปรแกรม Arduino ควบคุมโมดูลแสดงผล LCD ได้อย่างถูกต้อง (P)
3. นักเรียนปฏิบัติงานด้วยความตั้งใจและจัดเก็บอุปกรณ์เรียบร้อยเมื่อใช้งานเสร็จ (A)

เครื่องมือและอุปกรณ์การทดลอง

1. จอแสดงผล LCD I2C
2. สายจัมเปอร์
3. โฟโต้บอร์ด
4. บอร์ด Arduino
5. สาย USB
6. โปรแกรม Arduino IDE
7. คอมพิวเตอร์/โน้ตบุ๊ก ที่ติดตั้ง Arduino IDE

ภาคทฤษฎี

โมดูลแสดงผล LCD 16x2 (Liquid Crystal Display 16 ตัวอักษร 2 บรรทัด) เป็นอุปกรณ์ที่ใช้กันอย่างแพร่หลายในการแสดงผลข้อความและตัวเลข โดยเฉพาะเมื่อมีการต่อร่วมกับ โมดูล I2C (Inter-Integrated Circuit) ซึ่งเป็นโปรโตคอลการสื่อสารแบบอนุกรม

ลักษณะการทำงานและส่วนประกอบ

โมดูล LCD 16x2 I2C ประกอบด้วยส่วนสำคัญ 2 ส่วนที่ทำงานร่วมกัน

1. จอ LCD ทำหน้าที่แสดงผลตัวอักษร โดยมีโครงสร้างหลักเป็นตาราง 16 คอลัมน์ (ตัวอักษร) และ 2 บรรทัด



ส่วนประกอบของจอแสดงผล LCD 16x2 (Liquid Crystal Display)

- ผลึกเหลว (Liquid Crystals): เป็นวัสดุหลักที่ทำให้เกิดภาพ โดยโมเลกุลของผลึกเหลวจะถูกจัดเรียงตัวใหม่เมื่อได้รับสนามไฟฟ้า ทำให้แสงที่ผ่านเข้ามาเกิดการโพลาไรซ์ (Polarization) และปรากฏเป็นข้อความหรือตัวอักษร
- แผ่นกรองแสงโพลาไรซ์ (Polarizing Filters): เป็นแผ่นฟิล์มที่ใช้กรองแสงที่ผ่านผลึกเหลว เพื่อควบคุมให้แสงผ่านหรือไม่ผ่าน ซึ่งส่งผลให้เกิดความมืดหรือสว่าง (ตัวอักษร)
- หลอดไฟพื้นหลัง (Backlight): มักทำจาก LED (Light Emitting Diodes) สีเขียวหรือสีน้ำเงิน เพื่อให้มองเห็นข้อความได้ในที่มืด
- ตัวควบคุม (Controller Chip): โดยทั่วไปคือชิป Hitachi HD44780 หรือชิปที่เข้ากันได้ (Compatible) ซึ่งทำหน้าที่รับชุดคำสั่ง (Data และ Command) เพื่อจัดการตำแหน่งและรูปแบบของตัวอักษรบนหน้าจอ

2. โมดูล I2C คือส่วนที่สำคัญที่สุดในการเชื่อมต่อกับ Arduino ทำหน้าที่เป็นตัวแปลงสัญญาณขนานจากจอ LCD ให้เป็นสัญญาณอนุกรม ข้อดีคือช่วย ลดจำนวนสายไฟ ที่ต้องใช้ควบคุมจอจากกว่า 10 สาย เหลือเพียง 4 สายหลักเท่านั้น ทำให้ประหยัดขาพินของ Arduino



ส่วนประกอบของ โมดูลอินเทอร์เฟซ I2C (I2C Interface Module)

- ชิป PCF8574 (I/O Expander Chip): เป็นส่วนประกอบที่สำคัญที่สุดของโมดูลนี้ ทำหน้าที่เป็น ตัวขยายพอร์ต (I/O Expander) โดยแปลงสัญญาณอนุกรม (Serial Data) ที่มาจาก Arduino (SDA/SCL) ให้เป็นสัญญาณขนาน (Parallel Data) เพื่อป้อนเข้าสู่ขาควบคุมของจอ LCD (DB0-DB7, RS, E, RW)
- วัสดุ: ชิปนี้ทำจากวัสดุ สารกึ่งตัวนำ (Semiconductor) เช่น ซิลิคอน (Silicon)
- โพลเทนซิโอมิเตอร์ (Potentiometer): ทำหน้าที่เป็นตัวต้านทานแบบปรับค่าได้ ทำมาจากวัสดุประเภท ตัวต้านทาน (Resistive Material) ใช้สำหรับ ปรับความคมชัด (Contrast) ของตัวอักษรบนหน้าจอ
- วงจรอื่นๆ: ได้แก่ ตัวต้านทาน (Resistor) และตัวเก็บประจุ (Capacitor) ทั่วไปที่ใช้ในวงจรอิเล็กทรอนิกส์
- ขั้วต่อ (Pins): มักทำจากโลหะผสม เช่น ทองเหลือง หรือทองแดง เพื่อใช้เชื่อมต่อกับสายไฟ VCC, GND, SDA, SCL

การเชื่อมต่อและวิธีใช้งาน

การใช้งานโมดูล LCD I2C ทำได้ง่ายและรวดเร็วเนื่องจากการเชื่อมต่อที่เรียบง่าย:

- ขา VCC และ GND: สำหรับจ่ายไฟเลี้ยง 5V และต่อกราวด์ให้กับโมดูล
- ขา SDA (Serial Data) และ SCL (Serial Clock): เป็นขาที่ใช้ในการสื่อสารข้อมูลแบบอนุกรม โดยขา SDA จะใช้ส่งข้อมูล และ SCL จะใช้ส่งสัญญาณนาฬิกาควบคุมการรับ-ส่งข้อมูล ซึ่งบนบอร์ด Arduino Uno จะต่อกับขา A4 (SDA) และ A5 (SCL) ตามลำดับ

การเขียนโปรแกรมควบคุม

1. ต้องมีการติดตั้งและเรียกใช้ไลบรารี LiquidCrystal_I2C.h
2. ต้องระบุ ที่อยู่ (Address) ของโมดูล I2C (มักจะเป็น 0x27 หรือ 0x3F) เมื่อสร้าง Object ของ LCD เช่น LiquidCrystal_I2C lcd(0x27, 16, 2);
3. ใช้คำสั่ง lcd.init(); และ lcd.backlight(); เพื่อเริ่มต้นการทำงานและเปิดไฟพื้นหลังของจอ
4. ใช้คำสั่ง lcd.setCursor(col, row); เพื่อกำหนดตำแหน่ง ก่อนใช้ lcd.print("ข้อความ"); เพื่อแสดงผลตามที่ต้องการ

ภาคปฏิบัติ

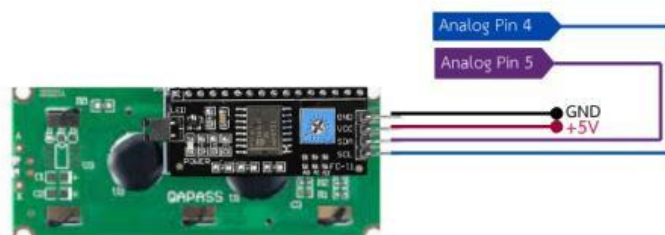
โปรแกรมที่1 แสดงข้อความ "HELLO...!"

โจทย์ ให้นักเรียนเขียนโปรแกรมเพื่อควบคุมโมดูล LCD I2C ให้แสดงผลดังนี้

1. แสดงข้อความ "HELLO...!" ในบรรทัดที่ 1
2. แสดงข้อความ "MR.ARDUINO" ในบรรทัดที่ 2
3. ข้อความทั้งสองบรรทัดปรากฏอยู่บนหน้าจอเป็นเวลา 2 วินาที
4. จากนั้นข้อความทั้งหมดจะถูกลบออก (หน้าจอว่างเปล่า) เป็นเวลา 2 วินาที
5. วงซ้ำขั้นตอนที่ 1-4 อย่างต่อเนื่อง

ขั้นตอนการต่อวงจร

ให้นักเรียนต่อสายไฟจากโมดูล LCD I2C เข้ากับบอร์ด Arduino Uno ตามตารางและภาพที่กำหนด



ขั้นตอนการเขียนโปรแกรม

1. เปิดโปรแกรม Arduino IDE และสร้างไฟล์สเก็ทใหม่
2. แก้ไขโค้ดต่อไปนี้ โดยตรวจสอบ ที่อยู่ (Address) ของโมดูลให้ถูกต้อง (ส่วนใหญ่คือ 0x27 หรือ 0x3F)

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16,
2);

void setup() {
    lcd.begin(16,2);
    lcd.setCursor(0,0);
    lcd.print("HELLO...!");
    lcd.setCursor(0,1);
    lcd.print("MR.ARDUINO");
    lcd.backlight();
}

void loop() {
    lcd.display();
    delay(2000);
    lcd.noDisplay();
    delay(2000);
}
```

3. อัปโหลดโค้ดไปยังบอร์ด Arduino และสังเกตผลลัพธ์ที่หน้าจอ LCD แล้วบันทึกผลการทดลอง

ตารางบันทึกผลการทดลอง

โปรแกรมที่ 1 แสดงข้อความ "HELLO...!"

ลำดับการทำงาน	สถานะหน้าจอ	คำสั่งหลักที่ใช้	ระยะเวลา (วินาที)
1			
2			
3			
4			

เมื่อทำการทดลองโปรแกรมที่ 1 เสร็จแล้ว ให้ทำการแก้ไขโปรแกรมในบรรทัดต่อไปนี้

lcd.setCursor(0,0); เป็น lcd.setCursor(4,0);

lcd.setCursor(0,1); เป็น lcd.setCursor(3,1);

เมื่อแก้ไขเสร็จแล้วทำการทดลองใหม่อีกครั้ง และบันทึกผลการทดลองในตารางด้านล่าง

สังเกตการทำงานของ	อธิบายลักษณะของการแสดงผล
LCD	

คำถามท้ายการทดลอง

1. จงระบุ ที่อยู่ (Address) ของโมดูล LCD I2C ของคุณที่ใช้ในการสร้าง Object (ตัวอย่าง: 0x27)

ตอบ

2. หากต้องการให้ข้อความปรากฏเป็นเวลา 5 วินาที และหายไป 1 วินาที จะต้องเปลี่ยนค่าในคำสั่ง delay() อย่างไรบ้าง?

ตอบ

3. คำสั่ง lcd.setCursor(5, 1); มีความหมายอย่างไรในการแสดงผลข้อความบนหน้าจอ LCD 16x2? (ระบุตำแหน่งบรรทัดและคอลัมน์)

ตอบ

4. หากต้องการทำให้ข้อความปรากฏอยู่บนหน้าจอ แต่ปิดไฟพื้นหลัง (Backlight) ของจอลงชั่วคราว ควรใช้คำสั่งใด?

ตอบ

5. สมมติว่าคุณใช้คำสั่ง lcd.noDisplay(); ในโค้ด และต่อมาต้องการให้ข้อความที่เคยแสดงผลไว้ก่อนหน้านี้ ปรากฏกลับมาอีกครั้ง โดยไม่ต้องสั่ง lcd.print() ซ้ำ ควรใช้คำสั่งใด?

ตอบ

โปรแกรมที่2 โปรแกรมนับ 0-100

โจทย์ ให้นักเรียนเขียนโปรแกรมเพื่อควบคุมโมดูล LCD I2C ให้แสดงผลการนับตัวเลขตั้งแต่ 0 ไปจนถึง 100 โดยมีเงื่อนไขดัง

1. แสดงข้อความ "COUNTER" ในบรรทัดที่ 1 (แสดงเพียงครั้งเดียวใน setup())
2. แสดงผลตัวเลขที่กำลังนับ (0 ถึง 100) ในบรรทัดที่ 2
3. การนับตัวเลขควรเพิ่มขึ้นทีละ 1 ทุกๆ 500 มิลลิวินาที (0.5 วินาที)

แนวคิดหลัก ใช้คำสั่ง for loop หรือตัวแปรนับที่เพิ่มขึ้นใน void loop() ร่วมกับคำสั่ง lcd.print(); เพื่อแสดงผลตัวแปร

ขั้นตอนการเขียนโปรแกรม

ข้อควรระวัง:

- ก่อนเริ่มโปรแกรมที่ 2 ให้นักเรียน ลบโค้ดทั้งหมด ของโปรแกรมที่ 1 ออกไป เพื่อป้องกันความสับสน

1. เปิดไฟล์สเก็ทเดิม (หรือสร้างใหม่)
2. แก้ไขโค้ดต่อไปนี้

```
#include <LiquidCrystal_I2C.h>          // ประกาศไลบรารี LiquidCrystal_I2C.h
LiquidCrystal_I2C lcd(0x27, 16, 2);    // เป็นการกำหนดค่าที่ใช้ในการเชื่อมต่อ LCD

void setup() {
    lcd.backlight();                    //สั่งให้จอแสดงผลมีไฟสว่าง
    lcd.begin(16,2);                    //กำหนดขนาดโมดูล LCD ขนาด 16x
    lcd.setCursor(5,0);                 //กำหนดเคอร์เซอร์ตำแหน่งที่ 5 ในบรรทัดที่ 1
    lcd.print("COUNTER");               //แสดงข้อความที่ต้องการแสดงผล
    delay(100);                         //หน่วงเวลา 0.1 วินาที
    int a=0;                            //กำหนดตัวแปร a=0
    lcd.setCursor(7, 1);                //กำหนดเคอร์เซอร์ตำแหน่งที่ 7 ในบรรทัดที่ 2
    lcd.print(a);                       //แสดงค่าตัวเลขในตัวแปร a
    while(a<=100) {                    //ตรวจสอบเงื่อนไขตัวแปร a ถ้า a มีค่าน้อยกว่าหรือเท่ากับ 100
        a=a+1;                          //ให้ทำตามคำสั่งในลูป while ตัวแปร a+1 ผลลัพธ์เก็บไว้ที่ a
        delay(500);                     //หน่วงเวลา 0.5 วินาที
        lcd.setCursor(7,1);             //กำหนดเคอร์เซอร์ตำแหน่งที่ 7 ในบรรทัด 2
        lcd.print(a);                   //แสดงค่าตัวเลขในตัวแปร a
    }
}

void loop() {
    lcd.clear();                        // ลบข้อความบนหน้าจอ LCD ทั้งหมด
}
```

ตารางบันทึกผลการทดลอง

โปรแกรมที่ 2 โปรแกรมนับ 0-100

ช่วงการนับ	บันทึกสิ่งที่พบเห็นแสดงบน LCD
เริ่มต้น (นับ 0-9)	
เปลี่ยนหลัก (9 - 10)	
เปลี่ยนหลัก (99 - 100)	
สิ้นสุดการนับ (เมื่อถึง 100)	

คำถามท้ายการทดลอง

1. ข้อใดอธิบายความแตกต่างของผลลัพธ์ระหว่างคำสั่ง `lcd.clear()`; กับคำสั่ง `lcd.noBacklight()`; ได้ถูกต้องที่สุด?

ตอบ

2. หากต้องการทำให้ข้อความที่เคยแสดงผลอยู่ หายไป (หน้าจอดับ) โดยที่ยังคงมีไฟพื้นหลังติดอยู่ และเมื่อสั่งให้แสดงผลอีกครั้งข้อความจะกลับมาทันทีโดยไม่ต้องใช้คำสั่ง `lcd.print()`

ตอบ

3. ต้องการให้โปรแกรมแสดงผลตัวเลขนับ 0-100 ซ้ำหลายๆ รอบ โดยให้หน้าจอว่างเปล่าทุกครั้งที่การนับเสร็จสิ้นและเริ่มต้นใหม่ นักเรียนควรวางคำสั่ง `lcd.clear()`; ไว้ที่ตำแหน่งใดในโค้ด?

ตอบ

4. หากต้องการให้แสดงตัวเลขในบรรทัดที่ 2 โดยให้อยู่ กึ่งกลาง ของหน้าจอ (คอลัมน์ 7) ควรใช้คำสั่ง `lcd.setCursor()` อย่างไร?

ตอบ

5. คำสั่งใดที่ใช้ในการ ปิดไฟพื้นหลัง (Backlight) และคำสั่งใดที่ใช้ในการ ลบข้อความ เพื่อให้หน้าจอ LCD ดับลงทั้งหมดเมื่อการนับเสร็จสิ้น?

ตอบ ปิดไฟพื้นหลัง :

ลบข้อความ :

สรุปผลการทดลอง (บรรยายความรู้ความเข้าใจที่ได้รับจากการทดลองเกี่ยวกับการใช้งานคำสั่งควบคุม LCD I2C)

คำถามท้ายการทดลอง

- โมดูล LCD เป็นอุปกรณ์ที่ทำหน้าที่
 - รับค่าอินพุต
 - หน่วยความจำ
 - Liquid Crystal Delay
 - Limit Crystal Display
 - เก็บค่าตัวแปร
 - แสดงผลข้อมูล
 - Liquid Crystal Display
 - Limit Crystal Delay
- LCD ย่อมาจากคำว่า
 - Liquid Crystal Delay
 - Limit Crystal Display
 - Liquid Crystal Display
 - Limit Crystal Delay
- โมดูล LCD I2C มีจำนวนสายไฟหลักที่ใช้เชื่อมต่อกับ Arduino Uno กี่สาย
 - 2 สาย
 - 8 สาย
 - 4 สาย
 - 16 สาย
- ขาพิน SDA (Serial Data Line) ของโมดูล LCD I2C บนบอร์ด Arduino Uno มาตรฐาน ควรเชื่อมต่อเข้ากับขาพินใด
 - Digital Pin 13
 - Analog Pin 4 (A4)
 - Analog Pin 0 (A0)
 - Analog Pin 5 (A5)
- ส่วนประกอบใดในโมดูล I2C ที่มีหน้าที่หลักในการ แปลงสัญญาณอนุกรม จาก Arduino ให้เป็นสัญญาณขนานเพื่อควบคุมจอ LCD
 - โพเทนชิโอมิเตอร์ (Potentiometer)
 - จอ LCD 16x2
 - ชิปตัวขยายพอร์ต PCF8574
 - หลอดไฟพื้นหลัง (Backlight LED)
- คำสั่งใดที่ใช้สำหรับเริ่มต้น (Initialize) การทำงานของโมดูล LCD I2C หลังจากที่เราสร้าง Object ในโค้ดแล้ว
 - lcd.begin();
 - lcd.start();
 - lcd.init();
 - lcd.print();
- หากต้องการให้ข้อความ "Welcome" ปรากฏบนบรรทัดที่สองและเริ่มที่ตำแหน่งคอลัมน์แรก (ซ้ายสุด) จะต้องใช้ชุดคำสั่งใดตามลำดับ
 - lcd.setCursor(1, 0); และ lcd.print("Welcome");
 - lcd.setCursor(0, 1); และ lcd.print("Welcome");
 - lcd.setCursor(0, 0); และ lcd.print("Welcome");
 - lcd.print("Welcome"); และ lcd.setCursor(0, 1);

8. โมดูลแสดงผล LCD แบบอักษระ มีจอแสดงผลสีใด

- ก. ขาว-ดำ
- ข. ขาว-เหลือง
- ค. ขาว-ส้ม
- ง. ขาว-แดง

9. คำสั่งใดเป็นการเขียนข้อความที่ต้องการแสดงบนโมดูลแสดงผล LCD

- ก. lcd.setCursor();
- ข. lcd.print();
- ค. LiquidCrystal();
- ง. lcd.begin();

10. คำสั่งใดเป็นการกำหนดตำแหน่งและบรรทัดของเคอร์เซอร์

- ก. lcd.setCursor();
- ข. lcd.print();
- ค. LiquidCrystal();
- ง. lcd.begin();

11. ต้องการทำให้ข้อความที่กำลังแสดงอยู่บนจอ หายไป (หน้าจอดำ) แต่ไม่ต้องการให้ข้อมูลที่เคยพิมพ์ถูกลบ เพื่อให้ข้อความกลับมาแสดงผลทันทีเมื่อใช้คำสั่ง lcd.display(); ควรใช้คำสั่งใดในการซ่อนข้อความ

- ก. lcd.clear();
- ข. lcd.noBacklight();
- ค. lcd.off();
- ง. lcd.noDisplay();

12. คำสั่งใดต่อไปนีที่ถูกต้องใช้เพื่อกำหนด ที่อยู่ (Address) ของโมดูล I2C (สมมติที่อยู่คือ 0x27) พร้อมระบุขนาดจอ 16x2

- ก. lcd.address(0x27, 16, 2);
- ข. LiquidCrystal_I2C lcd(0x27, 16, 2);
- ค. lcd.config(0x27, 16, 2);
- ง. lcd.setup(0x27, 16, 2);

13. ข้อใดคือความหมายของคำสั่ง lcd.display(); ที่ใช้ในไลบรารี LiquidCrystal_I2C

- ก. เป็นคำสั่งที่ใช้ในการพิมพ์ข้อความลงบนจอ
- ข. เป็นคำสั่งที่ใช้เปิดไฟพื้นหลังของจอ LCD
- ค. เป็นคำสั่งที่ใช้แสดงข้อความที่ถูกซ่อนไว้ด้วย lcd.noDisplay();
- ง. เป็นคำสั่งที่ใช้กำหนดความคมชัด (Contrast) ของตัวอักษร

14. หากนักเรียนสังเกตเห็นว่าจอ LCD แสดงผลข้อความได้ตามปกติ แต่ ไฟพื้นหลัง (Backlight) ไม่สว่าง นักเรียนควรใช้คำสั่งใดเพื่อแก้ไขปัญหา

- ก. lcd.clear();
- ข. lcd.noBacklight();
- ค. lcd.backlight();
- ง. lcd.init();

15. ในกิจกรรมที่ 2 (นับ 0-100) หลังจากทีนับเสร็จแล้ว (ถึง 100) ครูต้องการให้ ข้อความตัวเลขหายไปทันที เพื่อให้หน้าจอสะอาด แต่ยังคงมี ไฟพื้นหลังติดสว่างอยู่ ควรใช้คำสั่งใด

- ก. lcd.backlight();
- ข. lcd.noDisplay();
- ค. lcd.clear();
- ง. lcd.noBacklight();