

รายงานวิชาปฏิบัติการ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

รหัสวิชา 242-302 ตอน 02 วัน พฤหัสบดี

รหัสหัวข้อปฏิบัติการ 3HB07

ชื่อหัวข้อปฏิบัติการ Arduino Programming and Communication

วันที่ลงปฏิบัติการ 5 เมษายน 2561

อาจารย์ผู้สอน อาจารย์กุลวรรณ เขาวนวาที

ผู้จัดทำรายงานชื่อ นายธนากร ก้าโหด

รหัส 5835512048

สำหรับเจ้าหน้าที่

วันที่ตรวจรับ _____

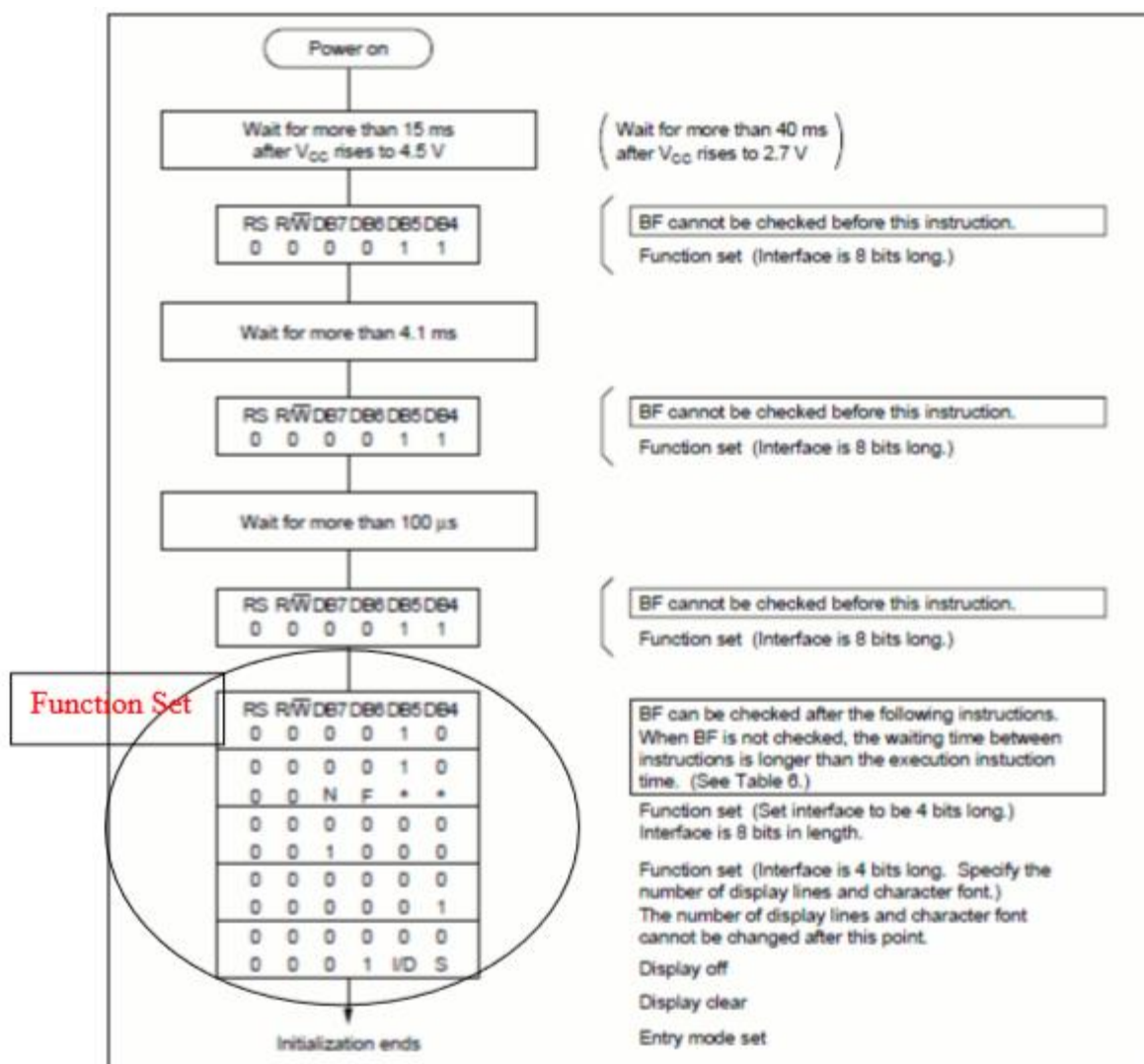
ลงชื่อ _____



3HB07

Arduino Programming and Communication

ขั้นตอนการตั้งค่า LCD ให้ทำงานโหมด 4 bit communication



จากภาพเป็นขั้นตอนการตั้งค่า LCD ให้ทำงานในโหมด 4 bit communication ซึ่งต้องเขียนโค้ดตั้งค่าตามขั้นตอนนี้ เพื่อให้ LCD ทำงานได้ถูกต้อง

Timing Diagram ของการเขียนค่าลง LCD

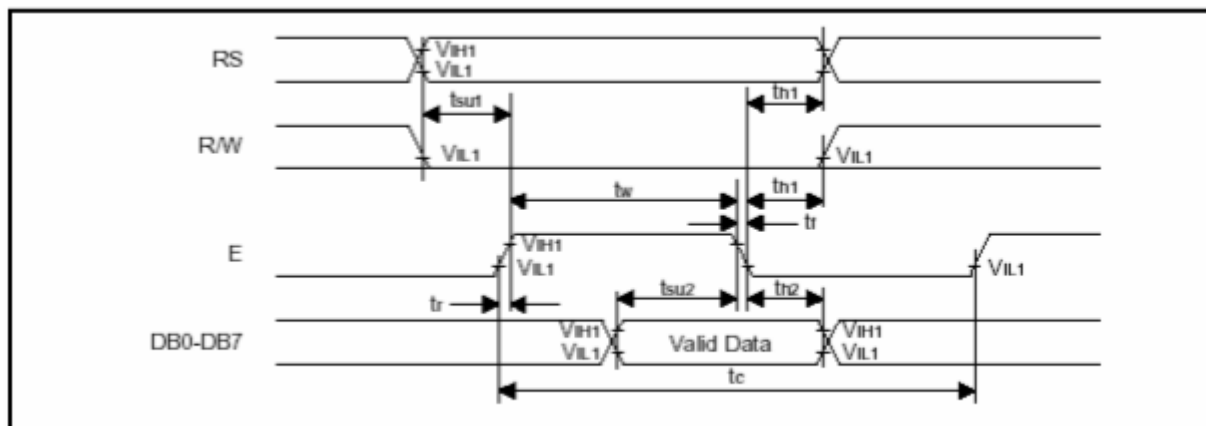


Figure 6. Write Mode Timing Diagram

จากภาพเป็น Timing Diagram ของการเขียนข้อมูลลง LCD ซึ่งต้องมีการทำตามขั้นตอนนี้ มีการ delay ตาม Timing Diagram ทุกครั้งที่มีการเขียนข้อมูลลง LCD

โค้ดการตั้งค่าการทำงานของ Board Arduino กับ LCD

```
const int rs_pin = 12;    //กำหนดตัวแปร rs_pin สำหรับติดต่อระหว่างขา 12 ของ Arduino กับขา rs ของ LED

const int en_pin = 11;    //กำหนดตัวแปร en_pin สำหรับติดต่อระหว่างขา 11 ของ Arduino กับขา en ของ LED

const int data_pin[] = {5,4,3,2};    //กำหนดตัวแปรอาร์เรย์ data_pin สำหรับติดต่อระหว่างขา 5,4,3,2 ของ Arduino
                                     กับขา data ของ LED
```

```
void write4bits(int value) //ฟังก์ชัน write 4 bits สำหรับเขียนค่าลงบนบอร์ด Arduino ส่งให้กับ LED ขนาด 4 บิต
```

```
{
```

```
    delayMicroseconds(1); //ดีเลย์ 1 microsec
```

```
    digitalWrite(en_pin,HIGH); //ให้ขา en มีค่า high
```

```
    for(int i=0;i<4;i++)
```

```
        digitalWrite(data_pin[i],(value >> i) & 1); //เขียนข้อมูลที่ละบิตจากหลังมาน้ำ
```

```
    delayMicroseconds(1); //ดีเลย์ 1 microsec
```

```
    digitalWrite(en_pin,LOW); //ให้ขา en มีค่า low
```

```

delayMicroseconds(100); //ดีเลย์ 100 microsec ก่อนมีการเขียนครั้งถัดไป

}

void sendLCD(int value, bool isData) //ฟังก์ชัน sendLCD สำหรับการส่งข้อมูลที่สามารถเลือกประเภทว่าเป็น
                                   //ข้อมูล หรือ คำสั่ง

{

    if(isData)

        digitalWrite(rs_pin,HIGH); //หากเป็นข้อมูล ให้ตั้งค่า rs เป็น High

    else

        digitalWrite(rs_pin,LOW); //หากเป็นคำสั่ง ให้ตั้งค่า rs เป็น Low

    delayMicroseconds(1); //ดีเลย์ 1 microsec

    write4bits(value >> 4); //ส่งข้อมูล 4 บิตบนไปก่อน

    write4bits(value); //ส่งข้อมูล 4 บิตล่างตามไป

}

void setupLCD() //ฟังก์ชัน setupLCD สำหรับตั้งค่า LCD

{

    pinMode(rs_pin,OUTPUT); //กำหนดขาที่ 12 ของ Arduino เป็นเอาต์พุต

    pinMode(en_pin,OUTPUT); //กำหนดขาที่ 11 ของ Arduino เป็นเอาต์พุต

    for(int i=0 ; i<4 ; i++) //กำหนดขาที่ 5,4,3,2 ของ Arduino เป็นเอาต์พุต

        pinMode(data_pin[i],OUTPUT);

    //เริ่มการตั้งค่าแบบ 4 bits communication

```

```

digitalWrite(rs_pin,LOW);           //ให้ขา rs เป็น low

digitalWrite(en_pin,LOW);           //ให้ขา en เป็น low

delay(80);                           //delay 80 millisecs

write4bits(0x03);                     //เขียนค่า 0011 ไปยัง LCD

delay(5);                             //delay 5 millisecs

write4bits(0x03);                     //เขียนค่า 0011 ไปยัง LCD

delay(5);                             //delay 5 millisecs

write4bits(0x03);                     //เขียนค่า 0011 ไปยัง LCD

delay(5);                             //delay 5 millisecs

write4bits(0x02);                     //เขียนค่า 0010 ไปยัง LCD

delay(5);                             //delay 5 millisecs

//ตั้งค่า Function Set

sendLCD(0x28, false);                 //ส่งคำสั่ง 0x28 ไปยัง LCD เพื่อตั้งค่ารูปแบบข้อมูลเป็น 4 bit แสดงผล 2 บรรทัด และ
                                     ตัวอักษรขนาด 5x8 จุด

sendLCD(0x06, false);                 //ส่งคำสั่ง 0x06 ไปยัง LCD เพื่อบอกทิศทางการเลื่อน Cursor และ ไม่อนุญาตการ shift
                                     ในหน้าแสดงผล

sendLCD(0x0F, false);                 //ส่งคำสั่ง 0x0F ไปยัง LCD เพื่อเปิดการแสดงผลผลลัพธ์ Cursor และการกะพริบ
                                     Cursor

sendLCD(0x01, false);                 //ส่งคำสั่ง 0x01 ไปยัง LCD เพื่อ Clear การแสดงผล

delay(10);                           //delay 10 millisecs

}

void setup()

{

```

```
setupLCD(); //เรียกใช้ฟังก์ชัน setupLCD() เพื่อตั้งค่า LCD }
```

Checkpoint 1 : ค่าในตาราง Function Set เป็นการกำหนดอะไรบ้าง

Instruction	Instruction Code										Description Instruction Code	Execution time (fsoc=270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC.	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	X	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39μs
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39μs
Function Set	0	0	0	0	1	DL	N	F	X	X	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39μs
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39μs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39μs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0μs
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43μs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43μs

```
sendLCD(0x28, false);
```

ส่งค่า 0010 1000 ไป เป็นการตั้งค่าแถว Function Set

DL = 1 ตั้งค่ารูปแบบข้อมูลเป็น 4 bit

N = 0 แสดงผล 2 บรรทัด

F = 1 ตัวอักษรขนาด 5x8 จุด

```
sendLCD(0x06, false);
```

ส่งค่า 0000 0110 ไป เป็นการตั้งค่าในแถว Entry Mode set

I/D = 1 บอกทิศทางการเลื่อน Cursor

SH = 0 ไม่อนุญาตการ shift ในหน้าแสดงผล

```
sendLCD(0x0F, false);
```

ส่งค่า 0000 1111 ไป เป็นการตั้งค่าในแถว Display On/Off Control

D = 1 เปิดการแสดงผลผลลัพธ์

C = 1 เปิดการแสดง LCD Cursor

B = 1 เปิดการกระพริบ Cursor

```
sendLCD(0x01, false);
```

เป็นการ Clear การแสดงผล

Checkpoint 2 : ทำการเขียนโปรแกรมเพื่อแสดงรหัสนักศึกษาแสดงบน LCD ในบรรทัดที่ 1 และ ชื่อในบรรทัดที่ 2

void loop() // หลังจากส่วน void setup() มีการตั้งค่า LCD แล้ว เติมโค้ดส่วนนี้เพื่อแสดงชื่อ และรหัสนักศึกษา

```
{
```

```
//tanakorn
```

```
sendLCD(0x54, true); //t
```

```
sendLCD(0x41, true); //a
```

```
sendLCD(0x4E, true); //n
```

```
sendLCD(0x41, true); //a
```

```
sendLCD(0x4B, true); //k
```

```
sendLCD(0x4F, true); //o
```

```
sendLCD(0x52, true); //r

sendLCD(0x4E, true); //n

sendLCD(0x3A, true); //:

//048

sendLCD(0x30, true); //0

sendLCD(0x34, true); //4

sendLCD(0x38, true); //8


sendLCD(0xC0, false); //new line


//masran

sendLCD(0x4D, true); //m

sendLCD(0x41, true); //a

sendLCD(0x53, true); //s

sendLCD(0x52, true); //r

sendLCD(0x41, true); //a

sendLCD(0x4E, true); //n

sendLCD(0x3A, true); //:

//083

sendLCD(0x30, true); //0

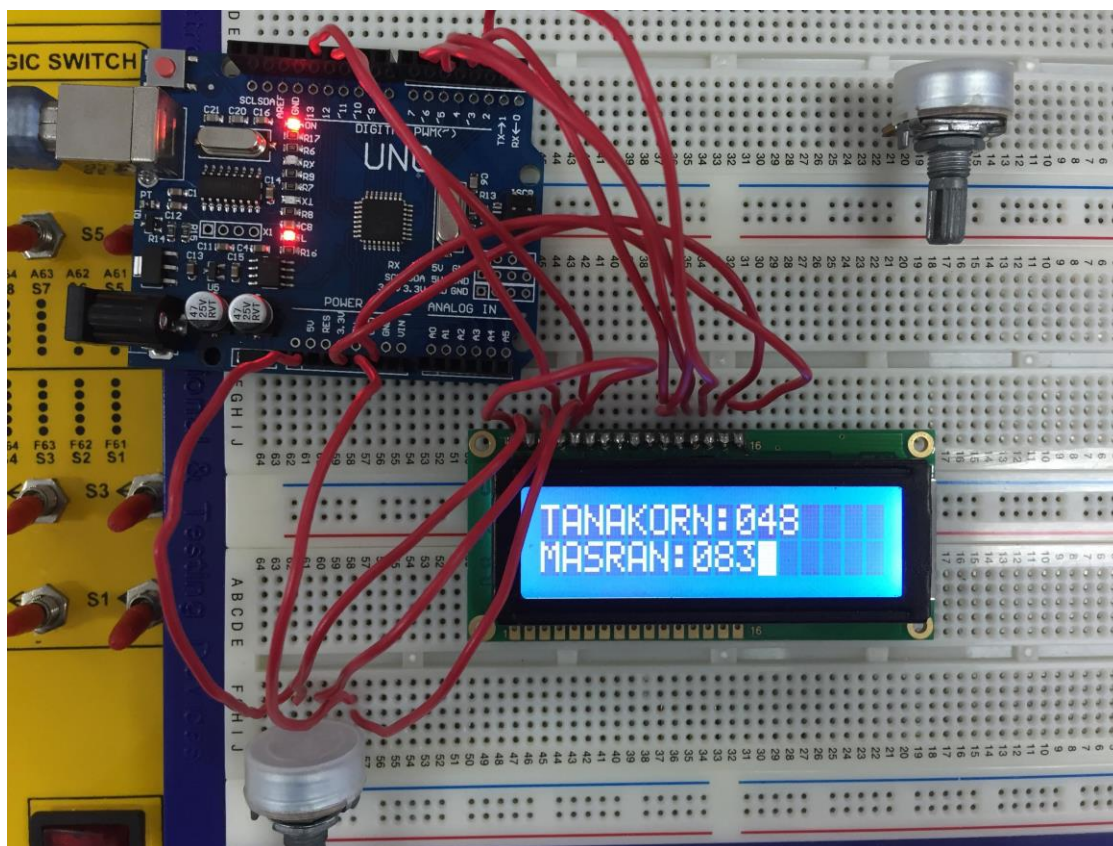
sendLCD(0x38, true); //8

sendLCD(0x33, true); //3
```



```
while(1);
```

```
}
```



Checkpoint 3 : ทำการเขียนโปรแกรมติดต่อ ADC ขา 0 โดยแสดงผลหน้าจอ LCD ในรูปแบบ ดังนี้

Sensor : XXXX

Voltage : y.yy

xxxx สัญญาณค่าระดับ Integer ที่อ่านได้โดยตรงผ่าน ADC ความละเอียด 10 bits

y.yy ระดับแรงดันสัญญาณ Input ที่มาจาก ADC โดยมีระดับแรงดันเต็มสเกลที่ 5 volts

```
const int rs_pin = 12;
```

```
const int en_pin = 11;
```

```

const int data_pin[] = {5,4,3,2};

int analog_in = A0;           //ตั้งขา A0 สำหรับรับค่าแรงดันอนาลอก

int adcValue = 0;             //adcValue สำหรับเก็บค่า digital ที่ได้รับการแปลงจาก analog แล้ว


void write4bits(int value)

{

    delayMicroseconds(1);

    digitalWrite(en_pin,HIGH);

    for(int i=0;i<4;i++)

        digitalWrite(data_pin[i],(value >> i) & 1);

    delayMicroseconds(1);

    digitalWrite(en_pin,LOW);

    delayMicroseconds(100);

}

void sendLCD(int value, bool isData)

{

    if(isData)

        digitalWrite(rs_pin,HIGH);

    else

        digitalWrite(rs_pin,LOW);

    delayMicroseconds(1);

    write4bits(value >> 4);

```

```

    write4bits(value);

}

void setupLCD()

{

    pinMode(rs_pin,OUTPUT);

    pinMode(en_pin,OUTPUT);

    pinMode(analog_in,INPUT);    //ตั้งขา A0 เป็นอินพุต

    for(int i=0 ; i<4 ; i++)

        pinMode(data_pin[i],OUTPUT);

    digitalWrite(rs_pin,LOW);

    digitalWrite(en_pin,LOW);

    delay(80);

    write4bits(0x03);

    delay(5);

    write4bits(0x03);

    delay(5);

    write4bits(0x03);

    delay(5);

    write4bits(0x02);

    delay(5);

    sendLCD(0x28, false);

```

```

    sendLCD(0x06, false);

    sendLCD(0x0F, false);

    sendLCD(0x01, false);

    delay(10);

}

void setup()

{

    setupLCD();

}

void loop()

{

    sendLCD(0x80, false); // clear หน้าจอ

    //read analog value

    adcValue = analogRead(analog_in); //รับค่า analog จากขา A0


    sendLCD(0x53, true); //S

    sendLCD(0x45, true); //E

    sendLCD(0x4E, true); //N

    sendLCD(0x53, true); //S

    sendLCD(0x4F, true); //O

```

```
sendLCD(0x52, true); //R
```

```
sendLCD(0x3A, true); // :
```

```
int div = 1000; // สร้างตัวหาร เพื่อทำการแยก
```

```
int m = adcValue; // เริ่มต้นให้ m เก็บค่าเต็มที่ได้รับมาจาก analog input
```

```
for(int i=3;i>=0;i--)
```

```
{
```

```
    int x = m/div; // x เก็บ หลักหน้าสุดของ m
```

```
    int dis = 0x30+x; // dis เก็บตัวอักษรสำหรับแสดงผล (เมื่อได้ค่า x ที่เป็นหลักหน้าสุด นำมาบวก 0x30 เพื่อแปลงเป็นค่า ascii ของ x ซึ่ง 0x30 เป็นค่า ascii ของเลข 0 และเลขอื่นๆ ก็อยู่ถัดไปตามลำดับ)
```

```
    sendLCD(dis, true); // แสดง dis ออกทาง LCD
```

```
    m = m%div; // ให้ m เป็นเลขที่เหลือ เมื่อตัด x ออก
```

```
    div = div/10; // ลดค่า div โดยหาร 10 เพื่อจัดการหลักถัดไปของค่า m
```

```
}
```

```
sendLCD(0xC0, false); // เลื่อน Cursor ลงบรรทัดใหม่
```

```
sendLCD(0x56, true); // V
```

```
sendLCD(0x4F, true); // O
```

```
sendLCD(0x4C, true); // L
```

```
sendLCD(0x54, true); // T
```

```
sendLCD(0x41, true); // A
```

```
sendLCD(0x47, true); // G
```

```
sendLCD(0x45, true); // E
```

```
sendLCD(0x3A, true); // :
```

int volt = (float)(adcValue*5/1023.00*100); // ให้ค่า Volt มีค่าระหว่าง 0 – 5 โดยเทียบกับค่า adcValue ซึ่งมีค่า 0 – 1023
(ใช้การเทียบบัญญัติไตรยางค์) แล้วนำค่าที่ได้คูณ 100 เพื่อเลื่อนค่าที่ได้มาอยู่
หน้าทศนิยมให้หมด จึงสามารถใช้ mod ในการแยกหลักได้

```
int e = volt/100; // e เก็บตัวหน้าสุดของ volt
```

```
int f = (volt%100)/10; // f เก็บทศนิยมตำแหน่งที่ 1 ของ volt
```

```
int g = volt%10; // g เก็บทศนิยมตำแหน่งที่ 2 ของ volt
```

```
sendLCD(0x30+e, true); // แสดงค่า e ลงบน LCD
```

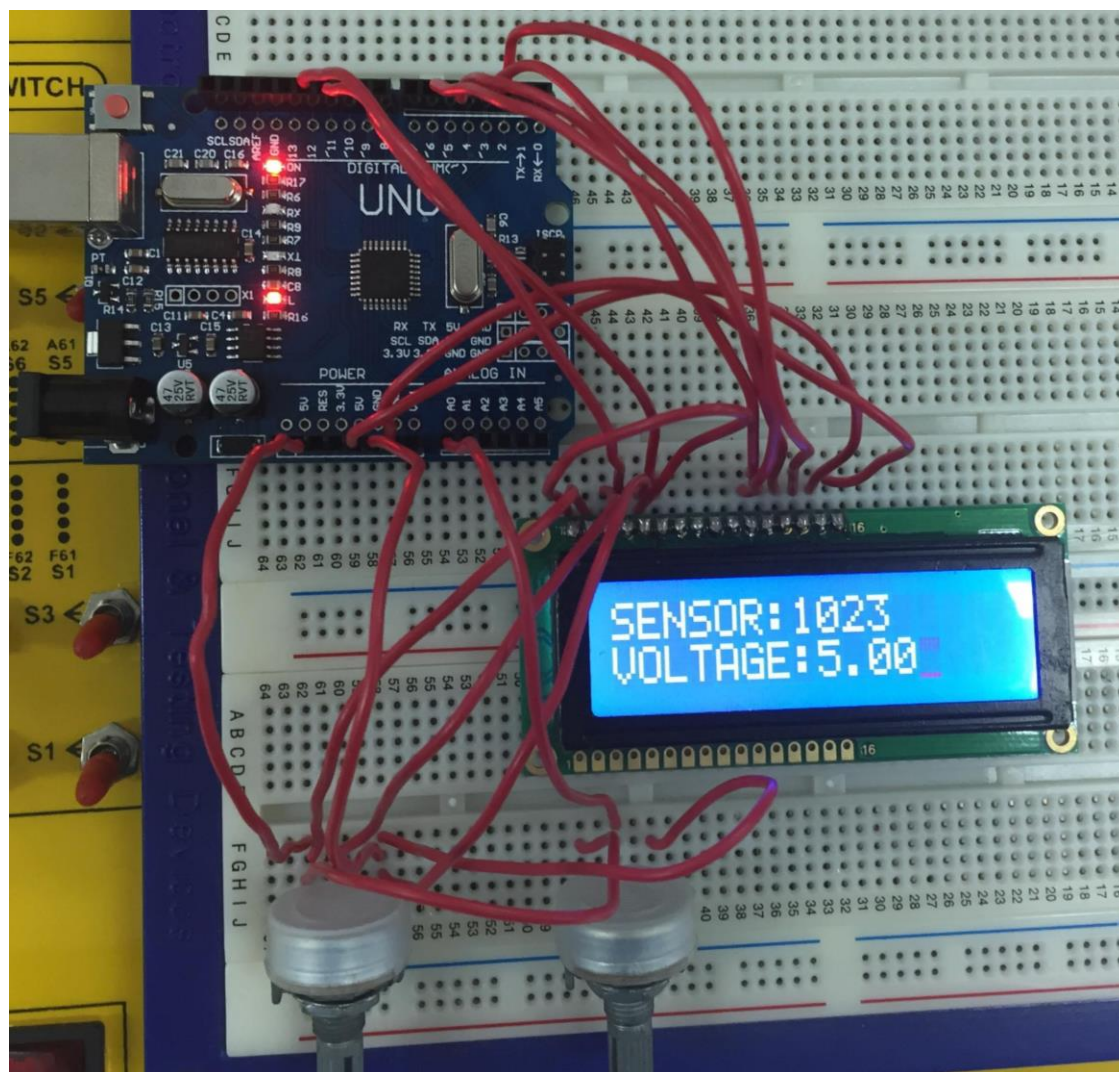
```
sendLCD(0x2E, true); // :
```

```
sendLCD(0x30+f, true); // แสดงค่า f ลงบน LCD
```

```
sendLCD(0x30+g, true); // แสดงค่า g ลงบน LCD
```

```
delay(100); // delay 100 millisec เพื่อแสดงคงค่าให้ผู้ใช้นี้เห็นระยะหนึ่ง
```

```
}
```



สรุปผลการทดลอง

จากการทดลอง Arduino Programming and Communication ทำให้ได้เรียนรู้ขั้นตอนการเขียนโปรแกรมเพื่อติดต่อกันระหว่าง Arduino กับ LCD ซึ่งมีขั้นตอนที่ค่อนข้างละเอียด ในขั้นตอนการตั้งค่าต้องมีการส่งข้อมูลไปยังขาต่าง ๆ ของ LCD ตามข้อมูลใน Datasheet

เมื่อตั้งค่าการติดต่อกันระหว่าง Arduino กับ LCD เสร็จสิ้นก็สามารถเขียนโปรแกรมเพื่อแสดงข้อความต่าง ๆ ลงบน LCD ได้ อีกทั้งสามารถอ่านค่า Analog input จาก Arduino แล้วส่งไปแสดงผลบน LCD ได้อีกด้วย