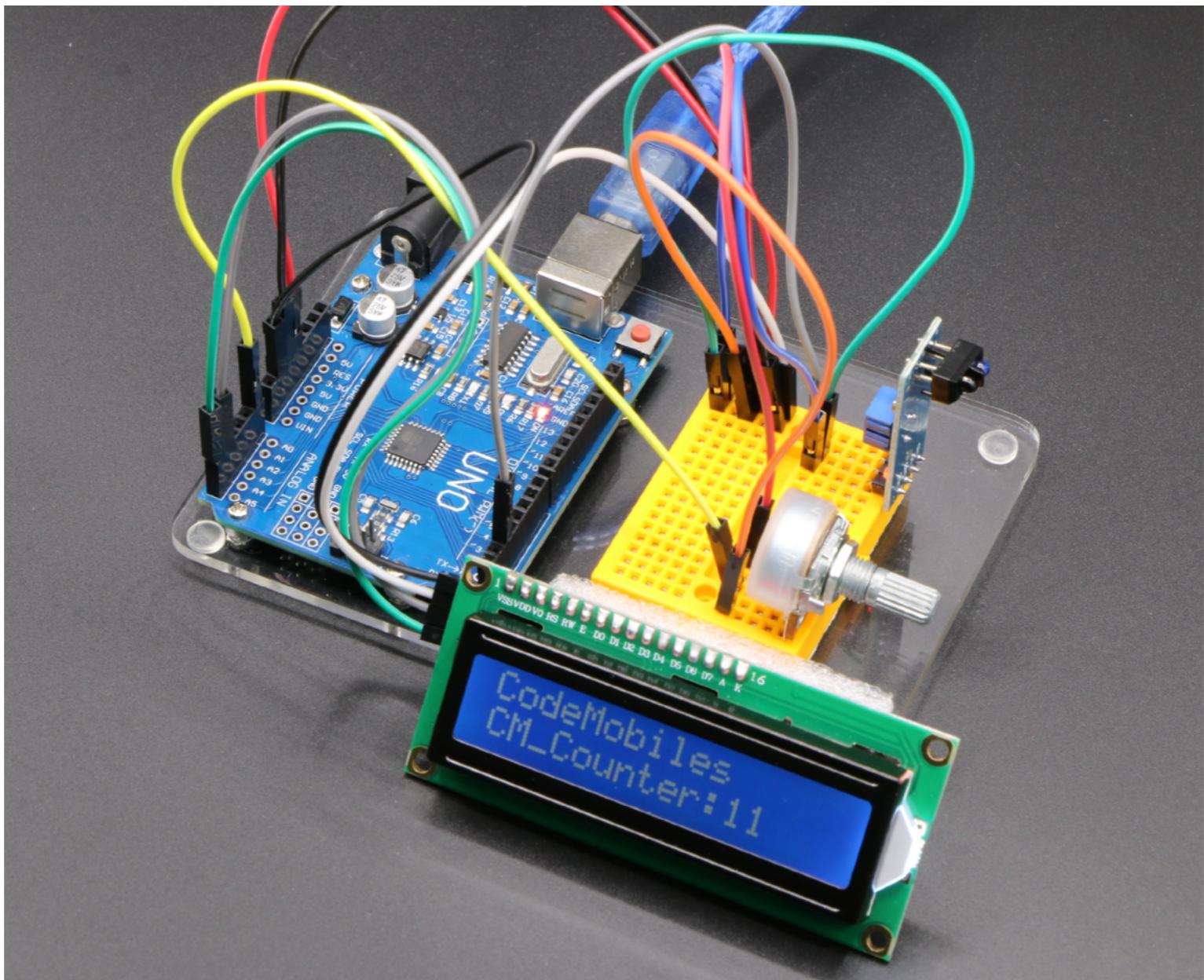


CODEMOBILES
COMPANY LIMITED



Arduino Starter Kit V1

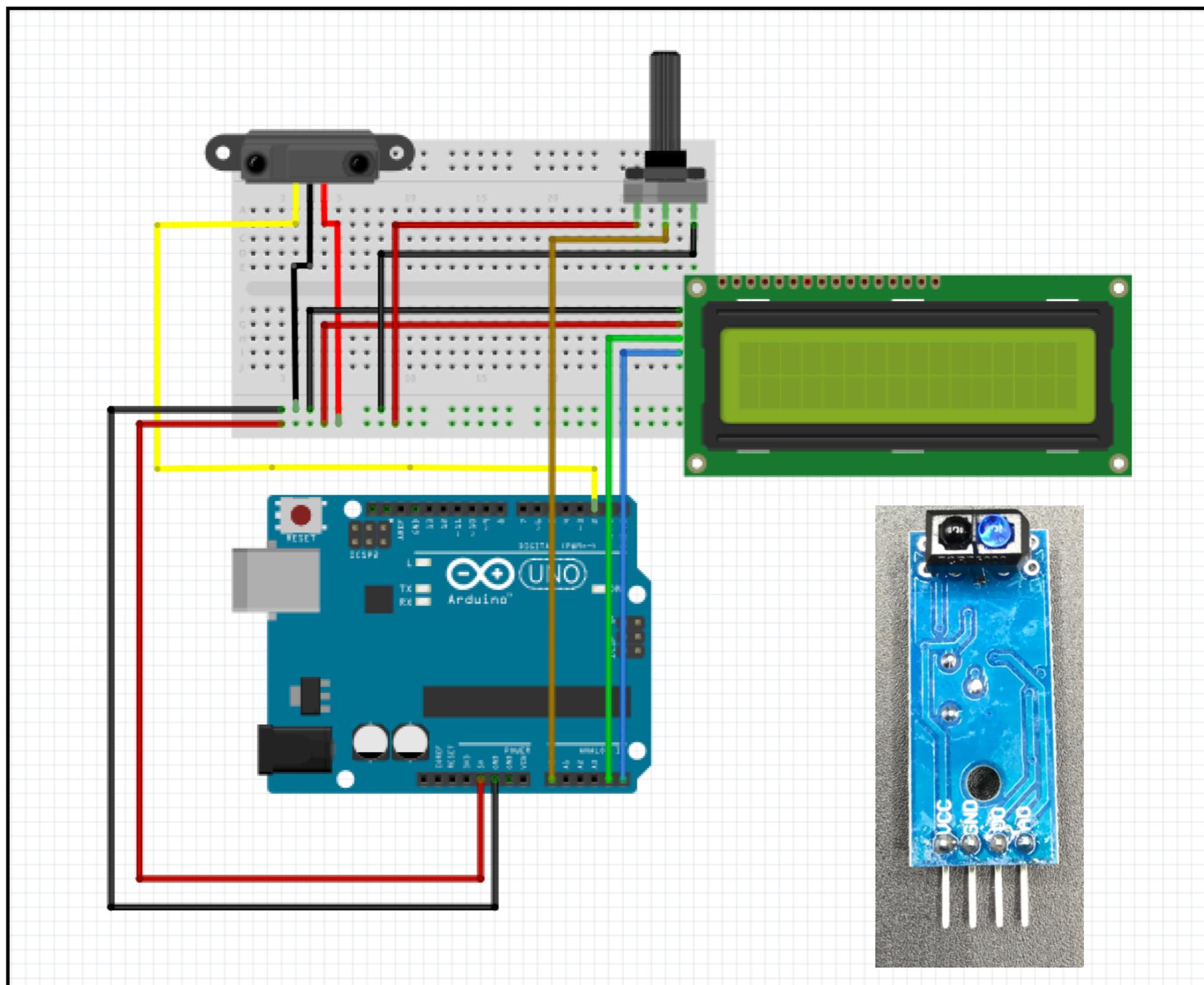
Arduino + Infrared + LCD + A2D

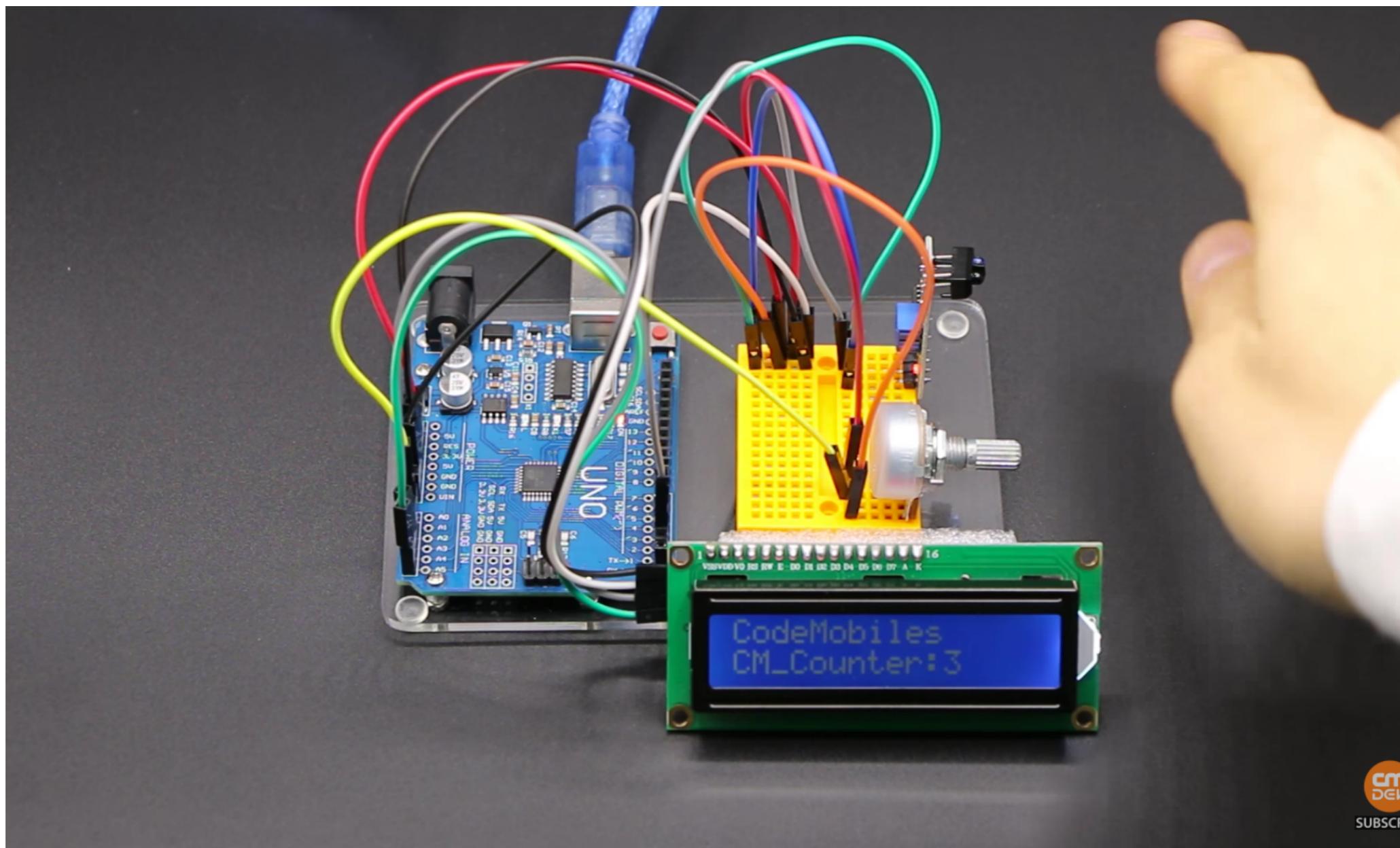
เนื้อหา เวิร์คช็อป

- พื้นฐานการเขียนบน Arduino
- การใช้ Serial Communication สำหรับการ Debug
- การเชื่อมต่อกับ Infrared IR Obstacle Sensor เพื่อตรวจสอบสิ่งกีดขวาง หรือ ประยุกต์ในการนับจำนวนวัตถุที่วิ่งผ่าน
- การแสดงผ่านบน LCD ผ่าน I2C
- การแปลงค่าแรงดันแบบ Analog เป็น Digital



ตัวอย่าง Diagram การเชื่อมต่อวงจรของ Workshop Arduino Starter Kit (Arduino + Infrared + LCD + Analog2Digital)





วิดีโอหลังทำสำเร็จ

https://www.youtube.com/watch?v=_8DaxO68E-4

การต่อสาย

Arduino Uno R3	Infrared Sensor
5V	VCC
GND	GND
2 (ฝั่งขา D)	D0

Arduino Uno R3	LCD Display
5V	VCC
GND	GND
A4	SDA
A5	SCL

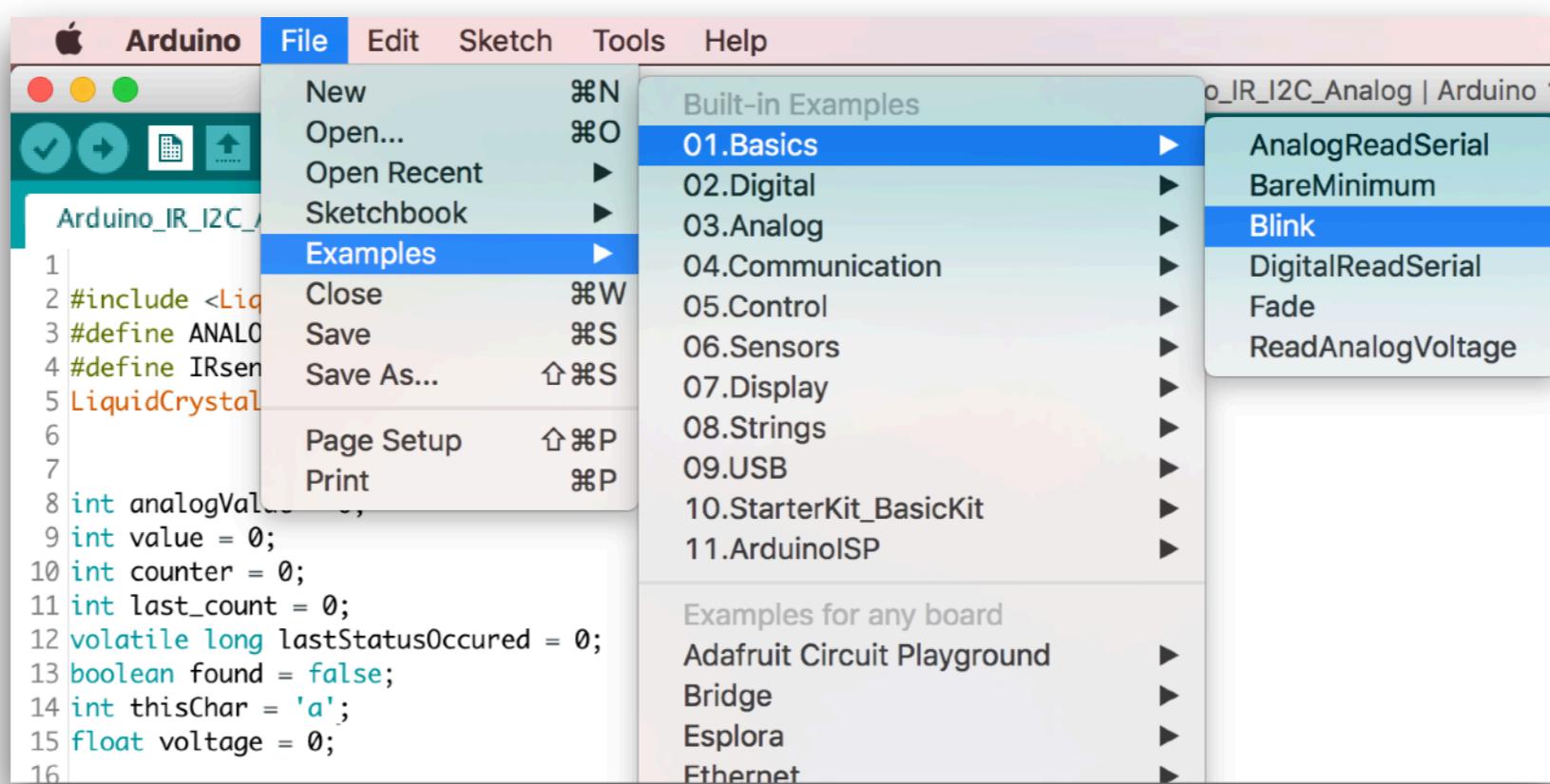
Arduino Uno R3	Analog2Digital
5V	ขาซ้าย หรือ ขวา
GND	ขาซ้าย หรือ ขวา
A0	กลาง

Workshop

เตรียมความพร้อม

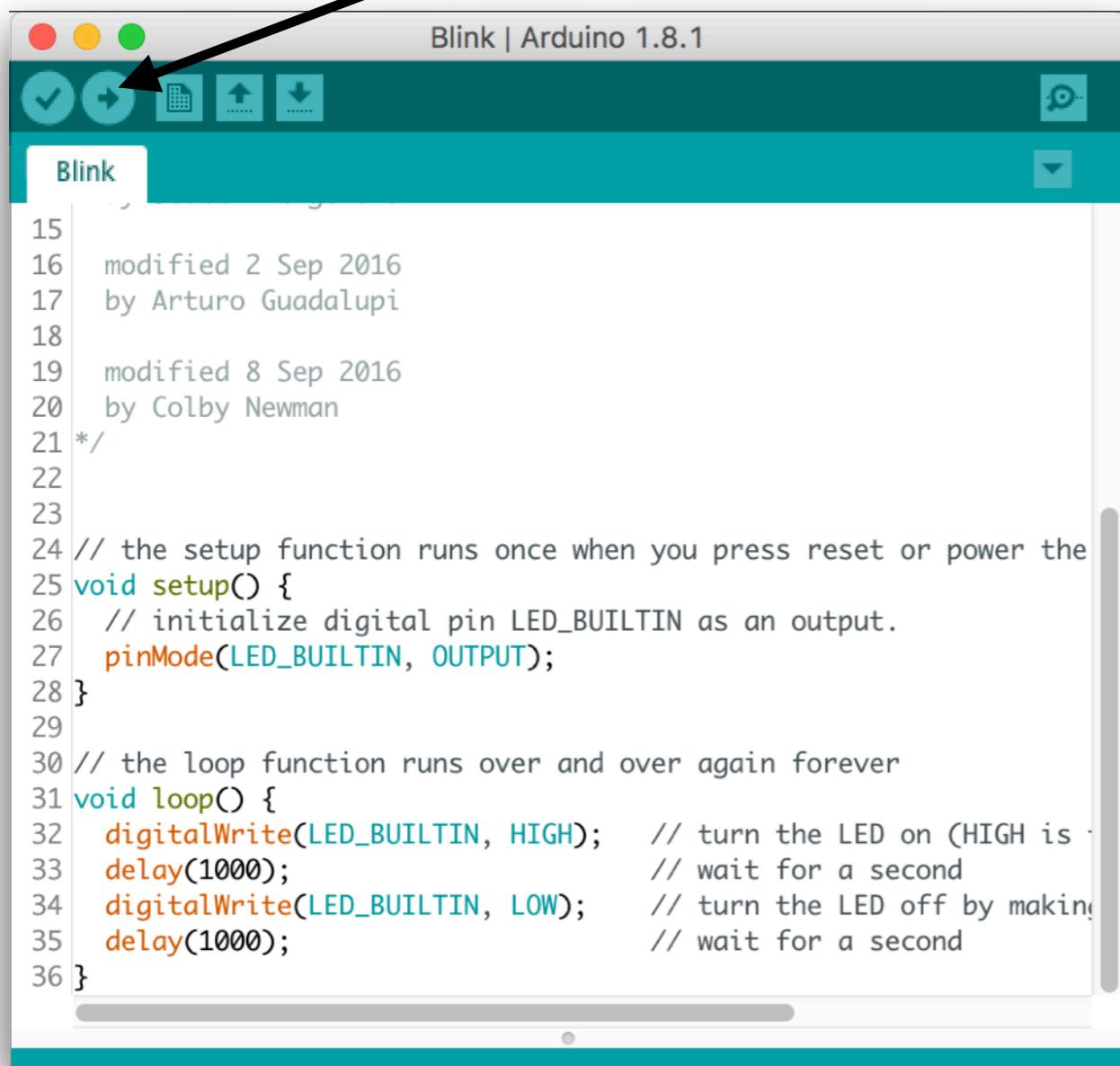
การทดสอบว่าบอร์ด Arduino UNO R3 สามารถทำงานได้ปกติหรือไม่
โดยทดสอบโดยใช้วิธี Blink หรือไฟกระพริบ ที่ LED BUILTIN หรือ LED ขา 13

1. ไปที่ File => Examples => 01.Basics => Blink



การทดสอบว่าบอร์ด Arduino UNO R3 สามารถทำงานได้ปกติหรือไม่ โดยทดสอบโดยใช้วิธี Blink หรือไฟกระพริบ ที่ LED BUILTIN หรือ LED ขา 13

2. กดปุ่ม Upload

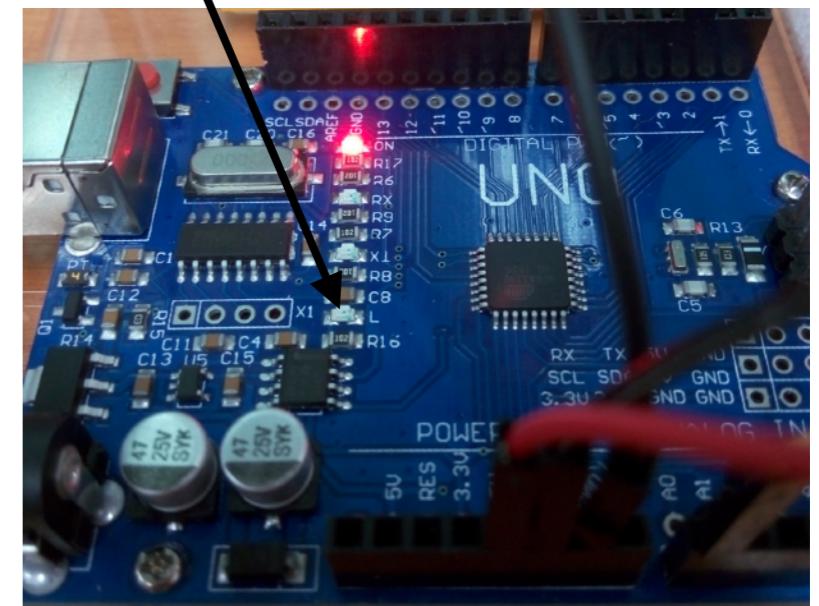
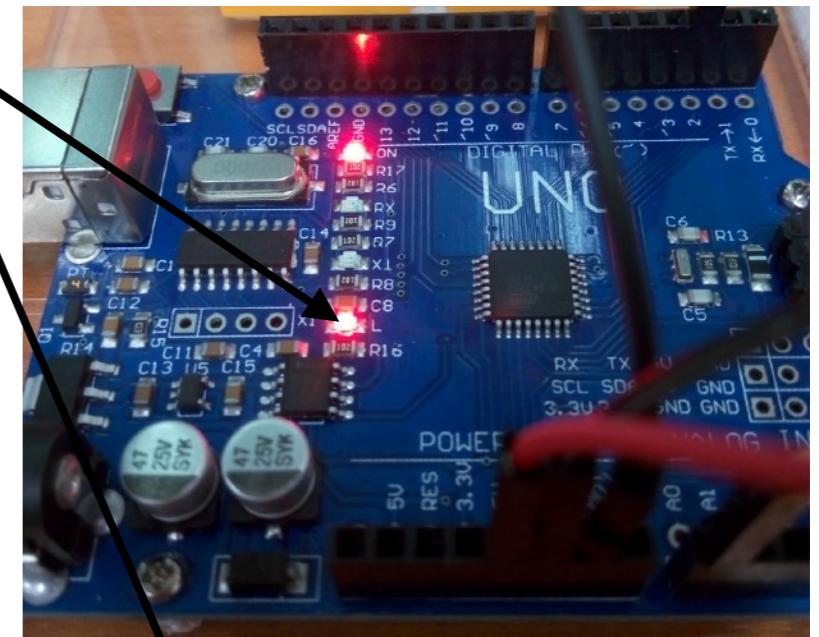


```
Blink | Arduino 1.8.1

Blink

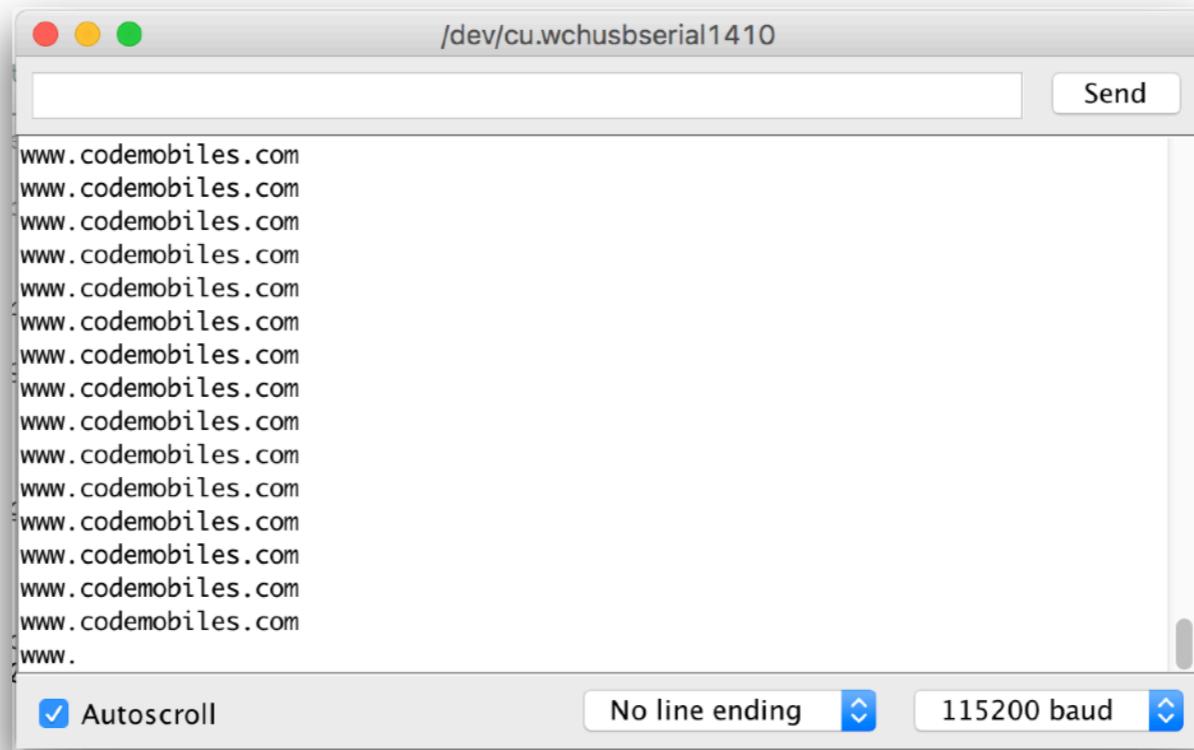
15
16 modified 2 Sep 2016
17 by Arturo Guadalupi
18
19 modified 8 Sep 2016
20 by Colby Newman
21 */
22
23
24 // the setup function runs once when you press reset or power the
25 void setup() {
26     // initialize digital pin LED_BUILTIN as an output.
27     pinMode(LED_BUILTIN, OUTPUT);
28 }
29
30 // the loop function runs over and over again forever
31 void loop() {
32     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is
33     delay(1000);                      // wait for a second
34     digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making
35     delay(1000);                      // wait for a second
36 }
```

LED BUILTIN



Workshop การใช้ Serial Communication สำหรับการ Debug

Serial Communication



การสั่งงานให้ NodeMCU สื่อสารกับคอมพิวเตอร์ของเรา ผ่านทางพอร์ตอนุกรม (Serial Port) โดยการให้มันส่งข้อความอะไรบางอย่างมาที่คอมพ์ของเรา กันสำหรับ Workshop นี้ยังไม่มีการต่อวงจรเพิ่มเติมครับเพียงแค่มีสาย USB กับบอร์ด NodeMCU

คำสั่งแรกที่ต้องใช้ในการเริ่มต้นสื่อสารคือ การกำหนดความเร็วในการสื่อสาร ตัวเลขยิ่งน้อยจะส่งข้อมูลได้ช้า แต่ระยะไกลและข้อมูลมีความสมบูรณ์ไม่เสียหายระหว่างการส่งข้อมูล หรือ ตัวเลขมากจะส่งข้อมูลได้ไว แต่ข้อมูลอาจเสียหายระหว่างการส่งข้อมูล คำสั่งที่ใช้ในการกำหนดความเร็วคือ

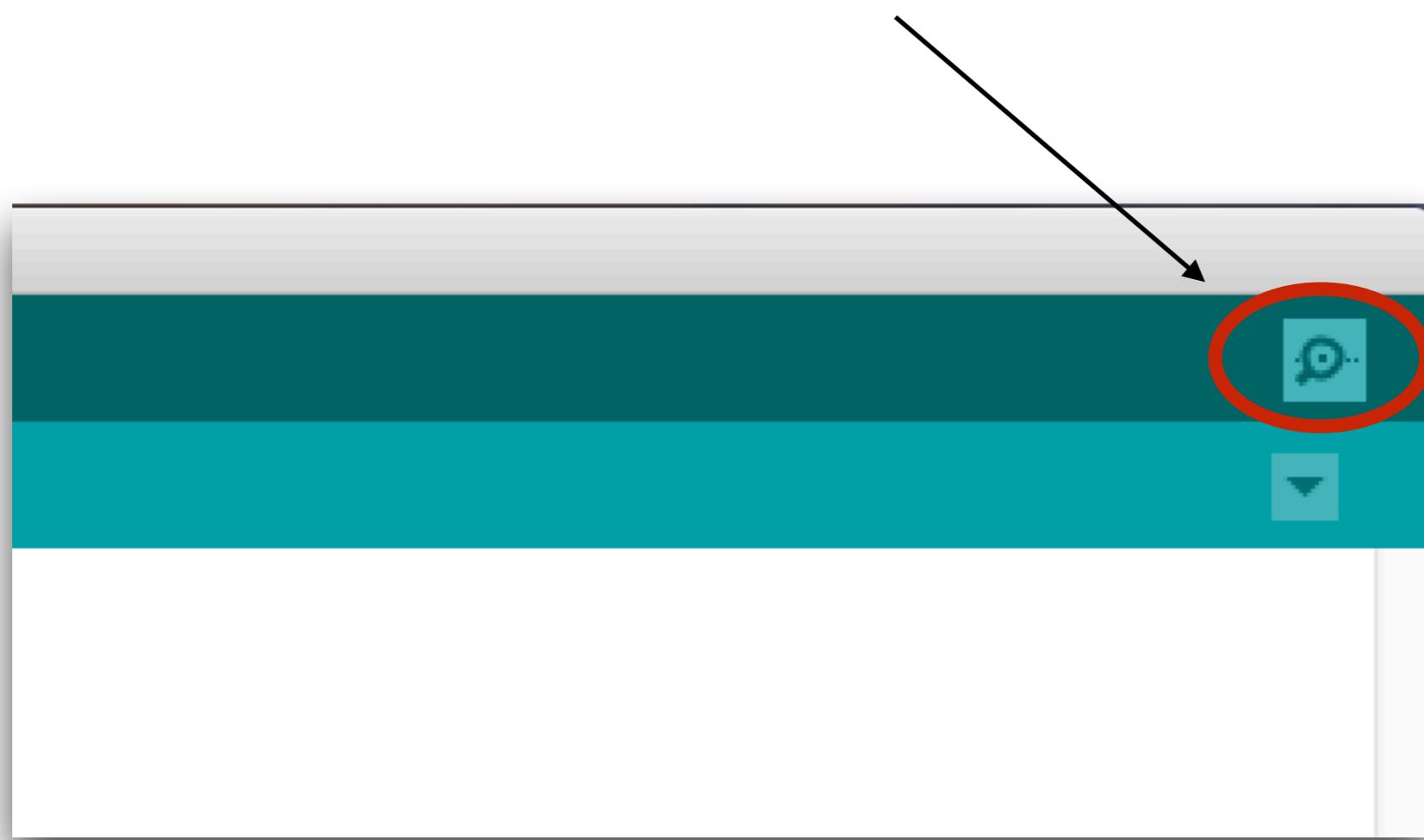
```
Serial.begin(115200);
```

คำสั่งสำหรับสั่งให้บอร์ดส่งข้อความมาที่คอมพิวเตอร์ของเรา คือ

```
Serial.print("codemobiles");
Serial.println("codemobiles");
```

ความแตกต่างระหว่าง print กับ println คือ print จะแสดงข้อความต่อ กันในแนวอนส่วน println จะแสดงข้อความแล้วขึ้นบรรทัดใหม่

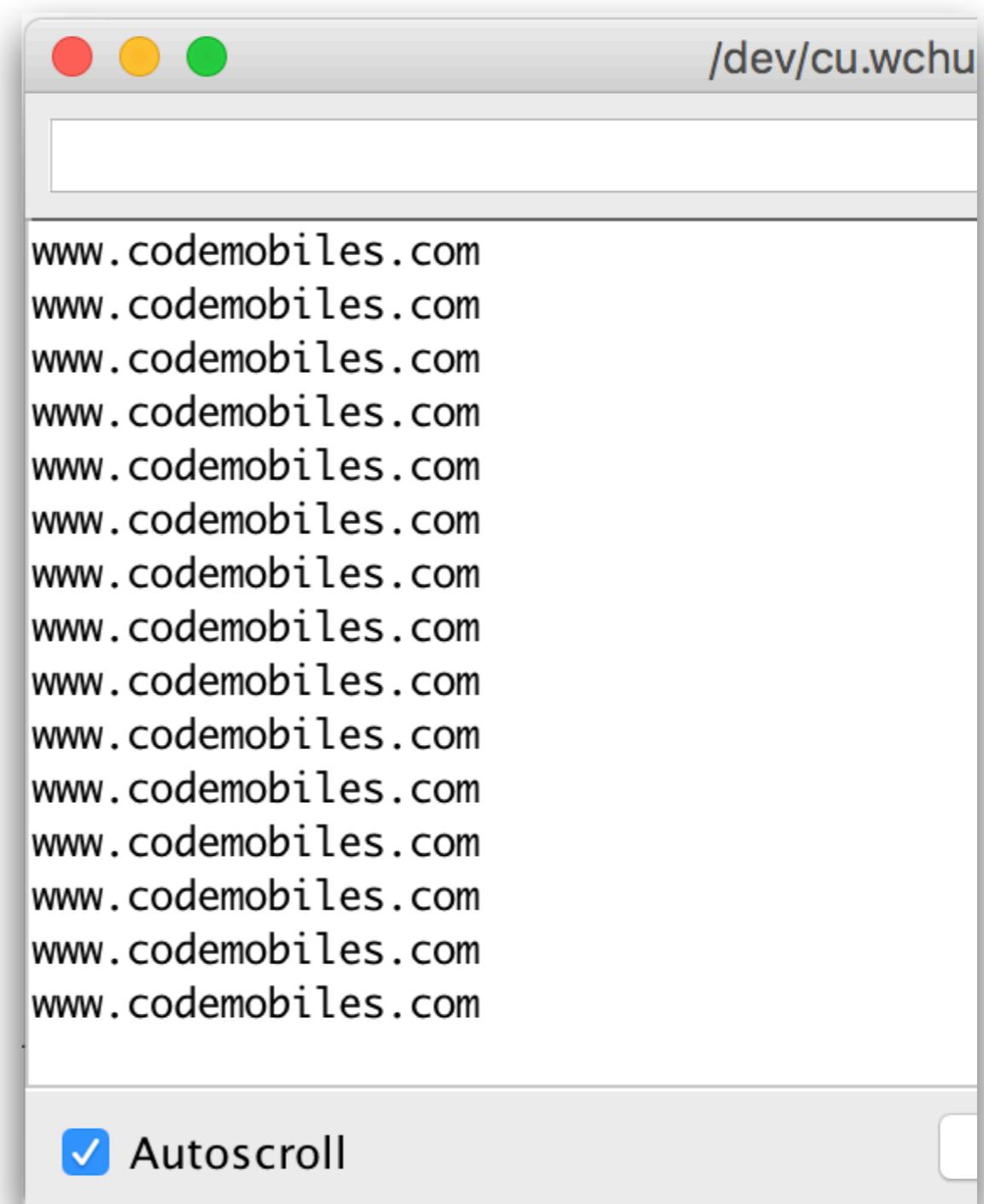
ถ้าเราต้องการดูว่า บอร์ดส่งข้อความอะไรมาให้เรา ให้เปิดดูที่ ไอคอน Serial Monitor ด้านขวาบนของตัวโปรแกรม Arduino โดยทุกครั้งที่เรากดไอคอนนี้ บอร์ด NodeMCU จะรีเซ็ตัวเอง



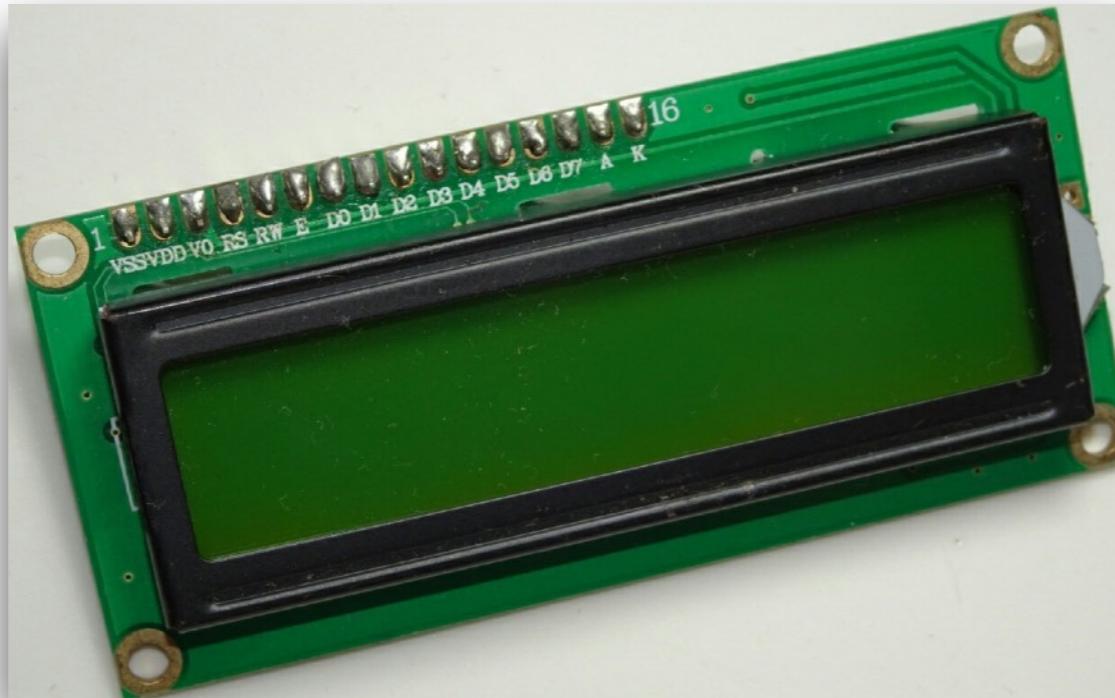
ตัวอย่างโค้ดโปรแกรม Serial Communication

```
void setup(){
    Serial.begin(115200);
}

void loop(){
    Serial.print("www");
    delay(1000);
    Serial.print(".");
    delay(1000);
    Serial.print("codemobiles");
    delay(1000);
    Serial.print(".");
    delay(1000);
    Serial.println("com");
    delay(1000);
}
```

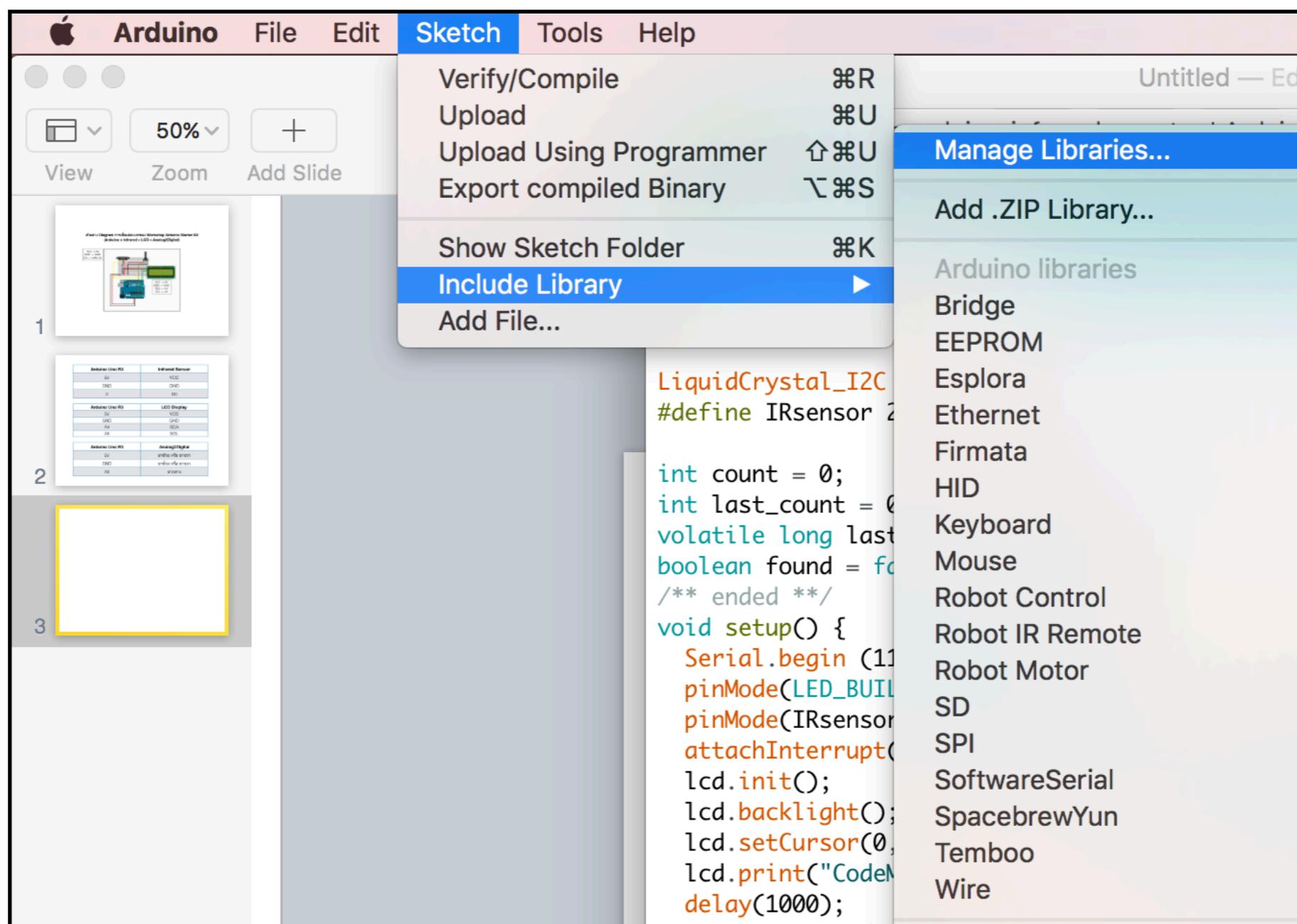


Workshop การแสดง ผ่านบาน LCD ผ่าน I2C

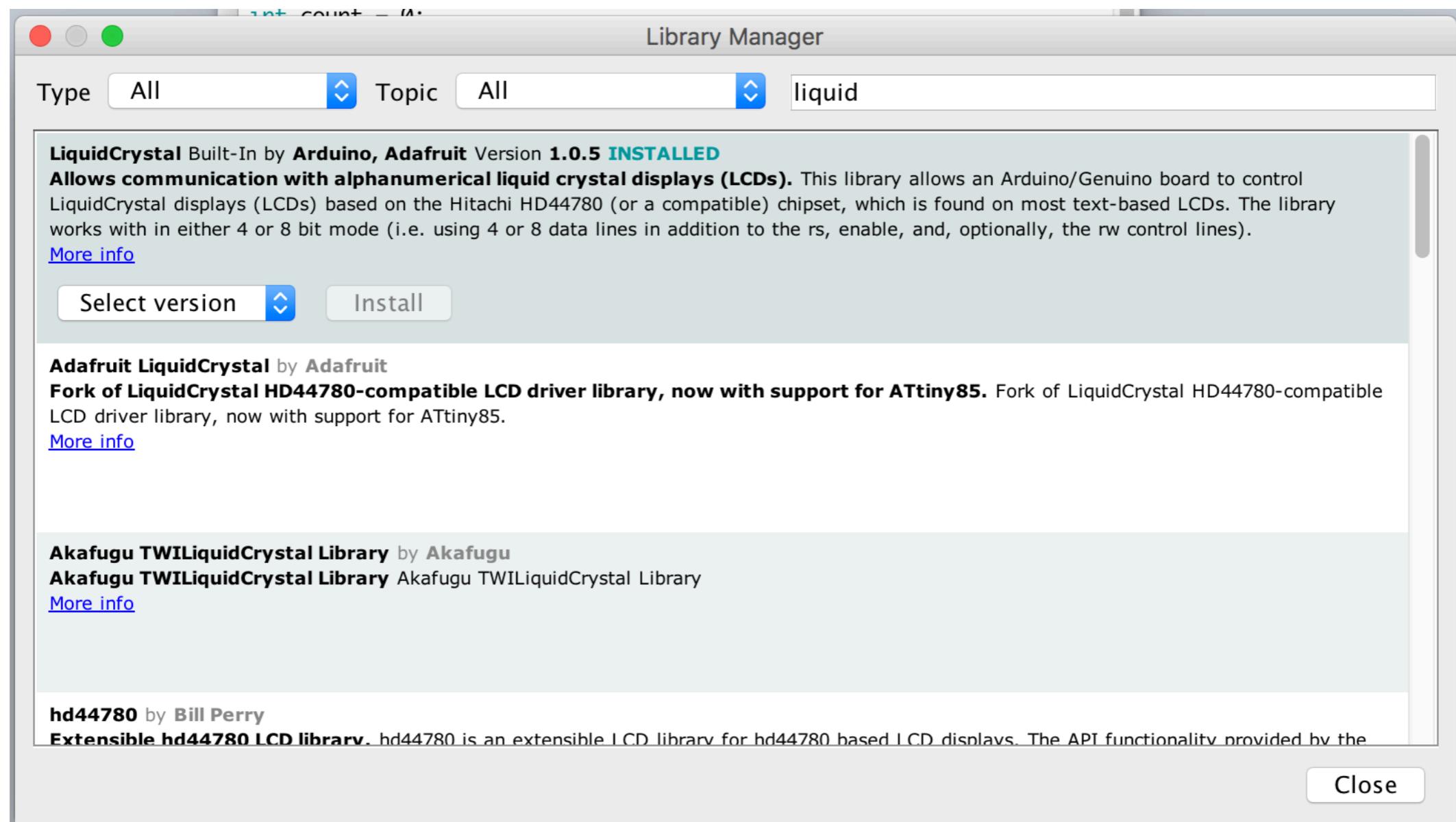


ก่อนอื่นต้องติดตั้ง Library LiquidCrystal เพิ่มเติมเพื่อใช้งานจอ LCD Display

1. ไปที่ Sketch -> Include Library -> Manager Libraries



2. ค้นหาคำว่า Liquid เลือก LiquidCrystal กดปุ่ม Install



การใช้งานจอ LCD นั้นจะมีการแสดงผล 2 บรรทัด แต่ละบรรทัดสามารถแสดงได้ 16 ตัวอักษร

คำสั่งแรกที่ต้องใช้คือ การเรียกใช้ Library LiquidCrystal เพื่อใช้งาน LCD

```
#include <LiquidCrystal_I2C.h>
```

คำสั่งต่อมาคือการประกาศตัวแปรเพื่อกำหนดข้อความบน LCD

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

คำสั่งต่อมาคือกำหนดค่าเริ่มต้นเพื่อใช้งาน จอ LCD

```
lcd.init();
lcd.backlight();
```

คำสั่งที่ใช้สำหรับกำหนดบรรทัด และตำแหน่งต้องการจะแสดงข้อความ

```
Lcd.setCursor(ตำแหน่งข้อความ, บรรทัดที่จะแสดงข้อความ);
```

คำสั่งต่อมาคือกำหนดข้อความลงบนจอ LCD

```
Lcd.setCursor(0, 0);
Lcd.print("CodeMobiles");
Lcd.setCursor(0, 1);
Lcd.print("Ready");
```

ตัวอย่างที่ 1 โค้ดโปรแกรม LCD การแสดงข้อความ

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

void setup(){
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("CodeMobiles");
    lcd.setCursor(0, 1);
    lcd.print("Ready");
}
```



ตัวอย่างที่ 2 โค้ดโปรแกรม LCD การแสดงข้อความ a-z

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

void setup(){
    initialDisplay();
}

void loop(){
    textDirection();
}

void initialDisplay(){
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
}
```

ตัวอย่างที่ 2 โค้ดโปรแกรม LCD การแสดงข้อความ a-z (ต่อ)

```
void textDirection(){  
    //เขียนย้อนกลับเมื่อ ค่ามีเท่ากับ 'm'  
    if (thisChar == 'm') {  
        lcd.rightToLeft();  
    }  
    //เขียนย้อนกลับเมื่อ ค่ามีเท่ากับ 's'  
    if (thisChar == 's') {  
        lcd.leftToRight();  
    }  
    if (thisChar > 'z') {  
        lcd.clear(); //เริ่มต้นใหม่  
        lcd.home();  
        thisChar = 'a';  
    }  
    lcd.write(thisChar); // เขียนตัวอักษร  
    delay(1000);  
    thisChar++; //เพิ่มค่าตัวอักษร  
}
```



ตัวอย่างที่ 3 โค้ดโปรแกรม LCD การแสดงตัวอักษรวิ่ง

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

void setup(){
    initialDisplay();
    setTextDisplay();
}

void loop(){
    scrollTextDisplay();
}

void initialDisplay(){
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
}

void setTextDisplay(){
    lcd.setCursor(0, 0);
    lcd.print("CodeMobiles"); delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("Ready"); delay(2000);
}
```

ตัวอย่างที่ 3 โค้ดโปรแกรม LCD การแสดงตัวอักษรวิ่ง (ต่อ)

```
void scrollTextDisplay(){
```

```
    for (int positionCounter = 0; positionCounter < 13; positionCounter++) {  
        lcd.scrollDisplayLeft();  
        delay(200);  
    }  
    for (int positionCounter = 0; positionCounter < 29; positionCounter++) {  
        lcd.scrollDisplayRight();  
        delay(200);  
    }  
    for (int positionCounter = 0; positionCounter < 16; positionCounter++) {  
        lcd.scrollDisplayLeft();  
        delay(200);  
    }  
    delay(5000);  
}
```

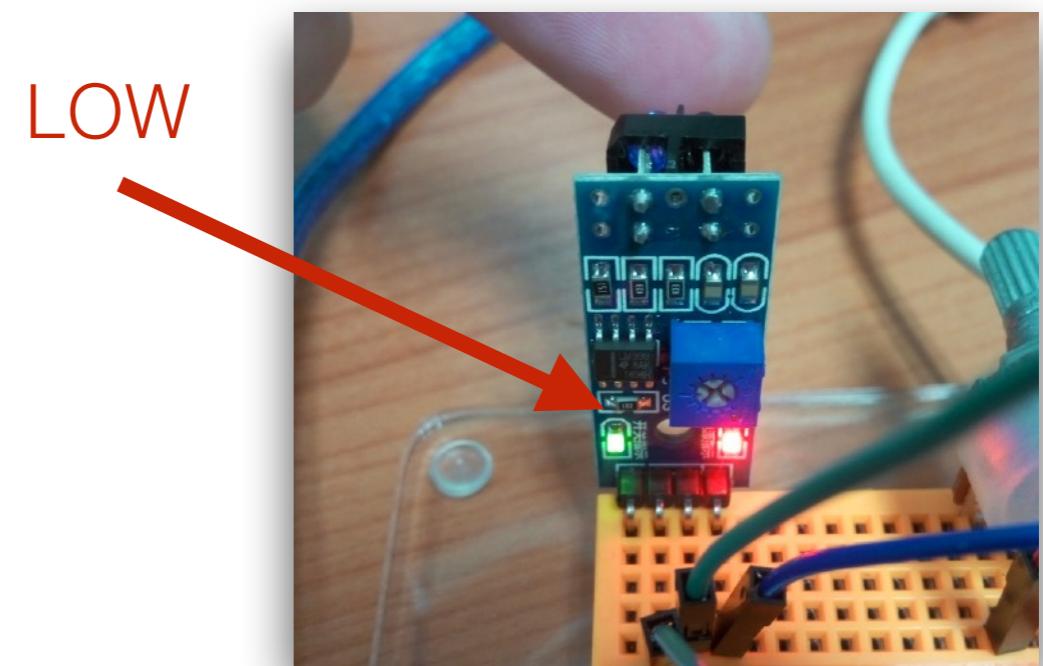
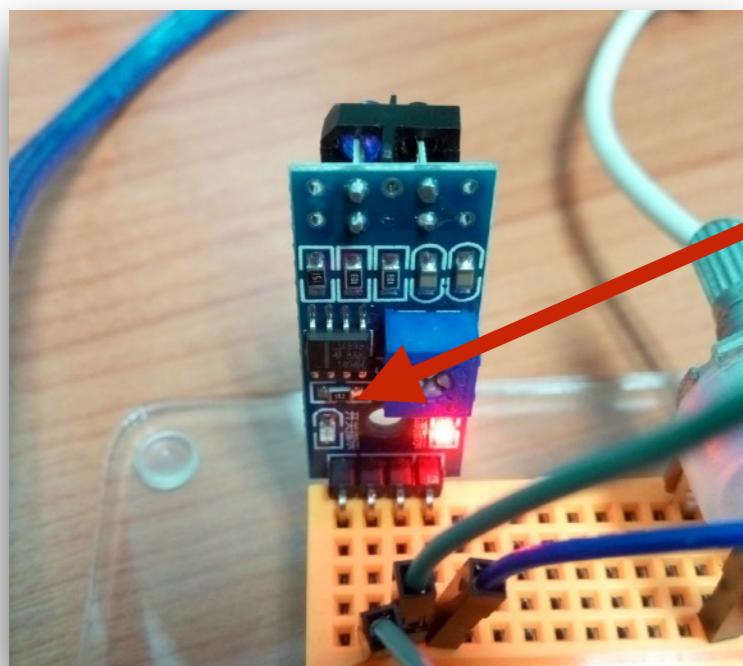


การใช้มุมต่อกับ Infrared IR Obstacle Sensor



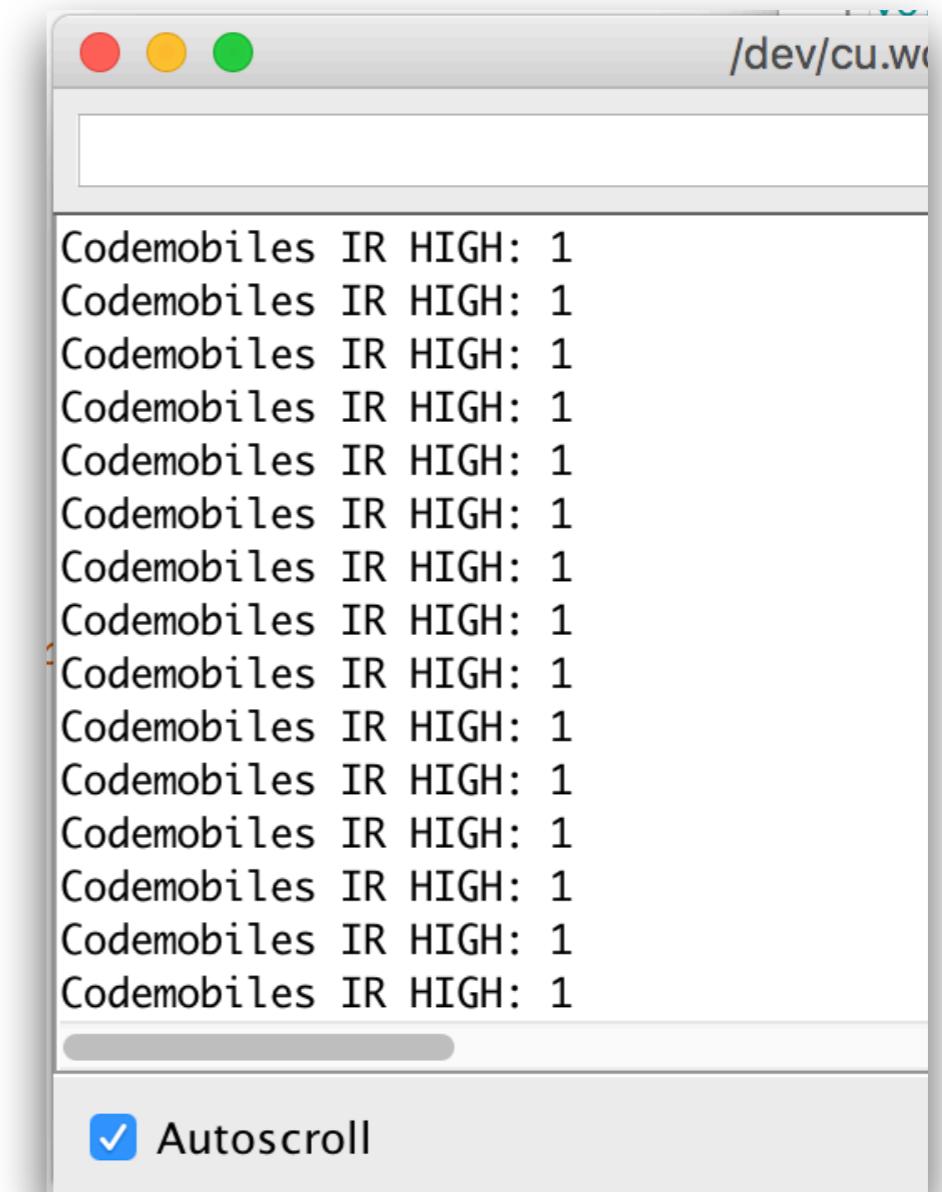
Infrared Obstacle Sensor เป็นเซนเซอร์วัดระยะที่ใช้หลักการสะท้อนของคลื่นอินฟราเรด สามารถกำหนดระยะในการทำงานได้โดยปรับค่าที่ Potentiometer Output เป็นแบบ Digital ใช้ไฟเลี้ยง 5V ระยะทำงานไม่เกิน 5 CM

สถานะเริ่มต้นของ Infrared จะเป็น HIGH เมื่อตรวจจับวัตถุ ในระยะไม่เกิน 5 CM ได้ สถานะ จะเปลี่ยนเป็น LOW



ตัวอย่างที่ 1 โค้ดโปรแกรม IR แสดงผลผ่าน Serial Monitor

```
int val = 0;  
int inputPin = 2;  
  
void setup(){  
    Serial.begin(115200);  
    pinMode(inputPin, INPUT);  
}  
  
void loop(){  
    val = digitalRead(inputPin); //การค่า IR  
    if (val == HIGH) {  
        Serial.print("Codemobiles IR HIGH: ");  
        Serial.println(val);  
    }  
    else{  
        Serial.print("Codemobiles IR LOW: ");  
        Serial.println(val);  
    }  
}
```



ตัวอย่างที่ 2 โค้ดโปรแกรม ดักการเปลี่ยนแปลงค่า IR ผ่าน Interrupted Function

```
#define IRsensor 2
int counter = 0;
int last_count = 0;
volatile long lastStatusOccured = 0;
boolean found = false;

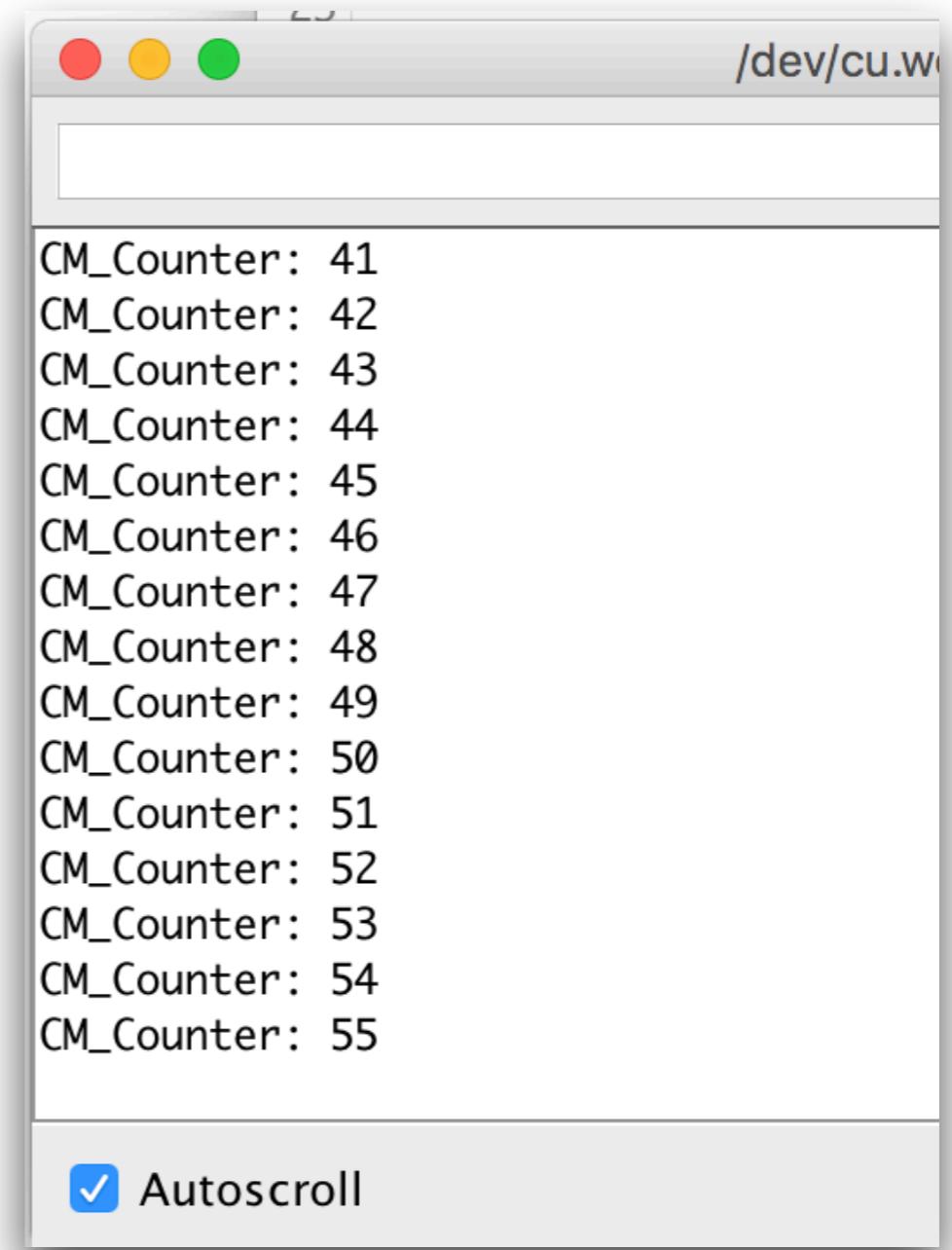
void setup(){
    initialization();
}

void loop(){
}

void initialization(){
    Serial.begin(115200);
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(IRsensor, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(IRsensor),interruptFunc,CHANGE);
}
```

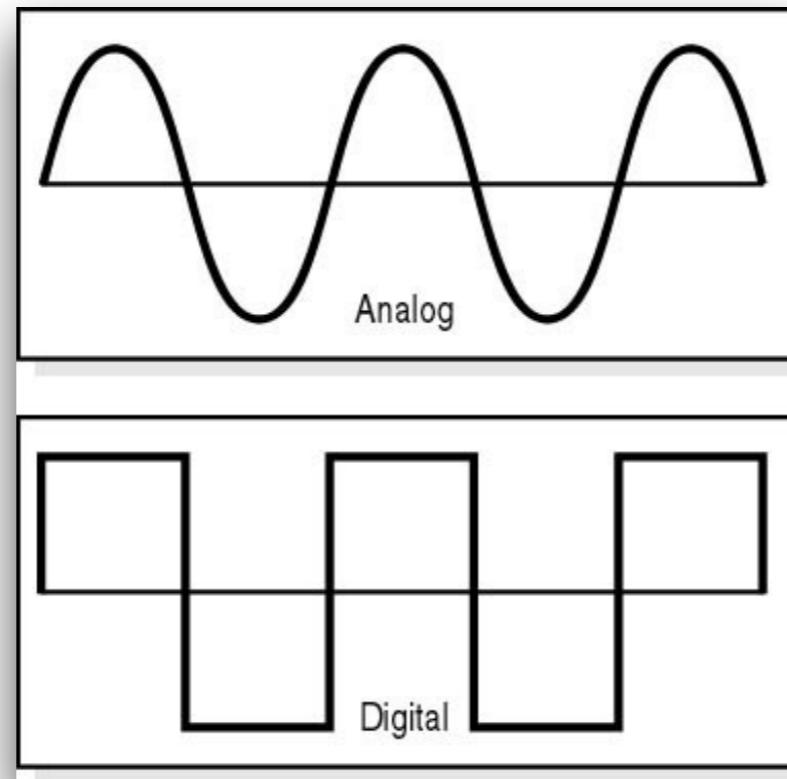
ตัวอย่างที่ 2 โค้ดโปรแกรม ดักการเปลี่ยนแปลงค่า IR ผ่าน Interrupted Function (ต่อ)

```
void interruptFunc(){
    long fin = millis();
    if(fin - lastStatusOccured > 50)
    {
        found = !found;
        if(found == true){
            counter++;
            if (last_count != counter){
                Serial.print("CM_Counter: ");
                Serial.println(counter);
            }
            last_count = counter;
        }
    }
    lastStatusOccured = fin;
}
```



การแปลงค่าแรงดันแบบ Analog เป็น Digital





สัญญาณ Analog เป็นสัญญาณที่มีความต่อเนื่องขนาดของสัญญาณเปลี่ยนแปลงตามเวลา ต่างจากสัญญาณแบบ digital ที่มีแค่ 0 กับ 1 เท่านั้น ใน NodeMCU ของเรา มี PIN สำหรับอ่านค่าสัญญาณ Analog อยู่ทั้งหมด 6 pin ครับ โดยการอ่านค่า analog ของ Arduino Nano จะแปลงค่าที่อ่านได้เป็นค่า digital แบบ 10 บิต หรือมีค่าเป็นจำนวนเต็มคือ 0 ถึง 1023

ตัวอย่างที่ 1 โค้ดโปรแกรม Analog to Digital แปลงค่าอะนาล็อก

แปลงค่าอะนาล็อกจาก 0 - 1023 ไปเป็น Voltage 0 - 5V

```
#define ANALOG_PIN A0

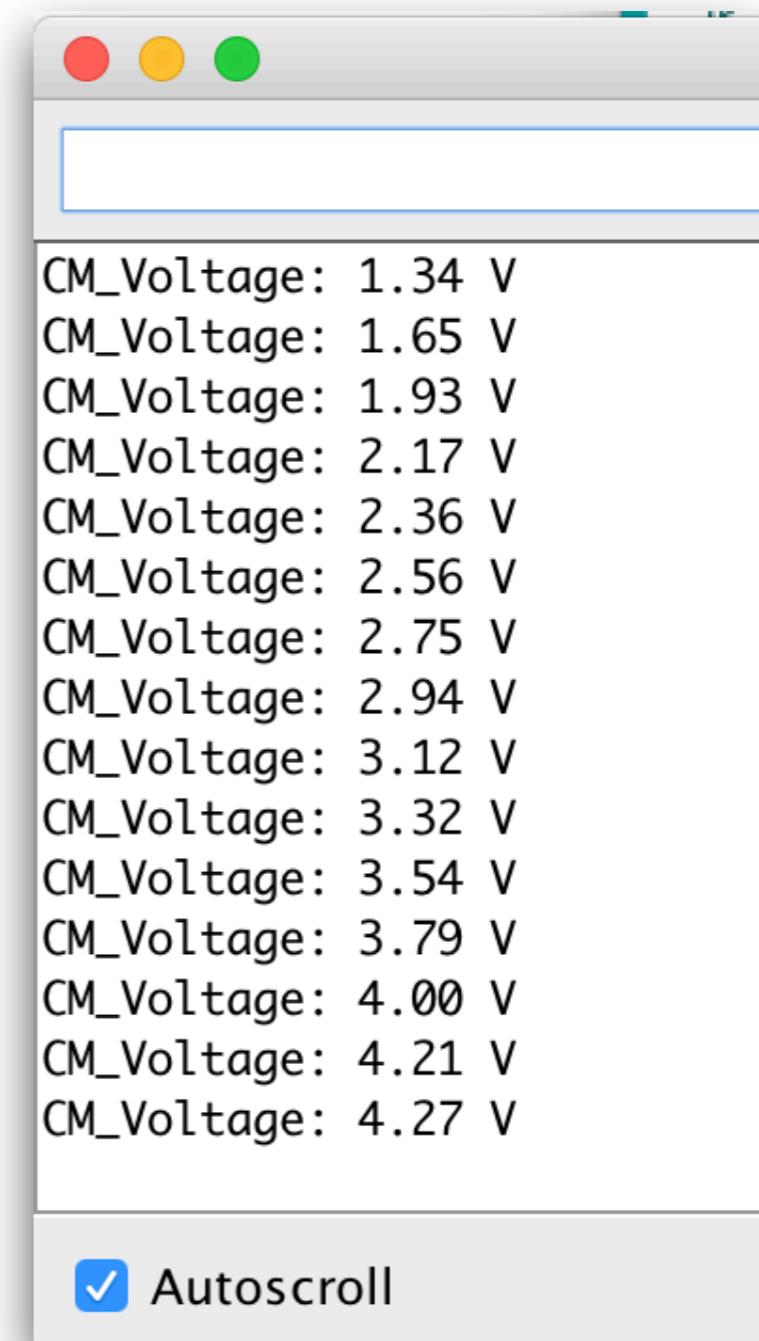
float voltage = 0;
void setup(){
    Serial.begin(115200);
}

void loop(){
    readAnalogVoltage();
}
```

ตัวอย่างที่ 1 โค้ดโปรแกรม Analog to Digital แปลงค่าอะนาล็อก (ต่อ)

void readAnalogVoltage(){

```
int sensorValue = analogRead(ANALOG_PIN);
voltage = sensorValue * (5.0 / 1023.0);
// print out the value you read:
Serial.print("CM_Voltage: ");
Serial.print(voltage);
Serial.println(" V");
delay(300);
}
```



ตัวอย่างที่ 2 โค้ดโปรแกรม Analog to Digital แสดงจำนวนนับ

ทำการตรวจสอบว่ามี R มีการเปลี่ยนแปลงค่ามากกว่าค่าปัจจุบันหรือไม่

- ถ้าใช่ให้เพิ่มจำนวน Counter
- หรือถ้าน้อยกว่าค่าปัจจุบันให้ลดจำนวน Counter

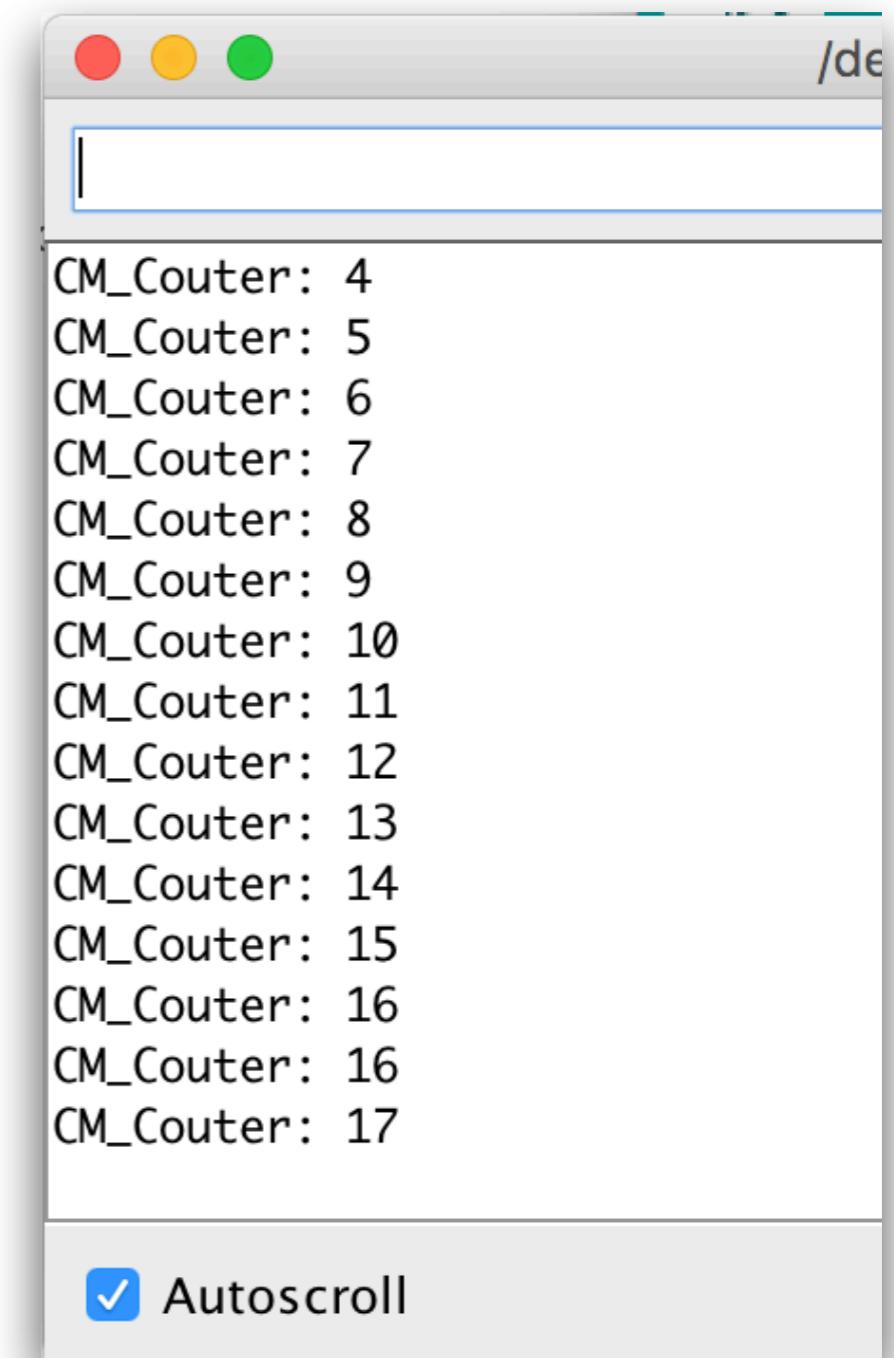
```
#define ANALOG_PIN A0
int analogValue = 0;
int value = 0;
int counter = 0;

void setup(){
    Serial.begin(115200);
}

void loop(){
    readAnalogValue();
}
```

ตัวอย่างที่ 2 โค้ดโปรแกรม Analog to Digital แสดงจำนวนนับ (ต่อ)

```
void readAnalogValue(){
    //อ่านค่า Analog
    analogValue = analogRead(ANALOG_PIN);
    if(analogValue>value+5 ){
        counter++;
    }else
        if(analogValue<value-5){
            if(counter>0){
                counter--;
            }
        }
    value = analogValue;
    Serial.print("CM_Couter: ");
    Serial.println(counter);
    delay(300);
}
```



Full Workshop

การทำเครื่องนับจำนวนด้วย Infrared และ แสดงผลบน LCD และสามารถปรับจำนวน ด้วยตัว R ปรับค่าได้

คำอธิบาย โค้ด โปรแกรม

การเรียกใช้ LCD Display จำเป็นต้องเรียกใช้งาน Library **LiquidCrystal_I2C** ในบรรทัดที่ 2

และประกาศตัวแปรเพื่อกำหนดการใช้งาน LCD Display ในบรรทัดที่ 5

```
1
2 #include <LiquidCrystal_I2C.h>
3 #define ANALOG_PIN A0
4 #define IRsensor 2
5 LiquidCrystal_I2C lcd(0x27, 16, 2);
6
7
8 int analogValue = 0;
9 int value = 0;
10 int counter = 0;
11 int last_count = 0;
12 volatile long lastStatusOccured = 0;
13 boolean found = false;
14
```

ในส่วนของ **void setup()** จะถูกเรียก ใช้งานเพียงครั้งเดียวอัตโนมัติ เพื่อกำหนดค่าเริ่มต้น เช่น

- กำหนดค่าเริ่มขา INPUT ขา OUTPUT
- กำหนดแสดงข้อความเริ่มต้น

```
13
14 void setup(){
15     initialization();
16     initialDisplay();
17     setTextDisplay();
18     setCounterDisplay();
19 }
20
```

ในส่วนของฟังชั่น **interruptFunc()** จะถูกเรียก ใช้งานเมื่อ **CHANGE** เกิด
สถานะการเปลี่ยนแปลง

```
61
62 void interruptFunc()
63 {
64     long fin = millis();
65     if(fin - lastStatusOccured > 50)
66     {
67         found = !found;
68         if(found == true){
69             counter++;
70
71             if (last_count != counter){
72                 Serial.print("CM_Counter: ");
73                 Serial.println(counter);
74             }
75             last_count = counter;
76         }
77     }
78     lastStatusOccured = fin;
79 }
```

คำสั่ง **void loop()** จะทำงานแบบวนลูปจนกว่าไม่ได้

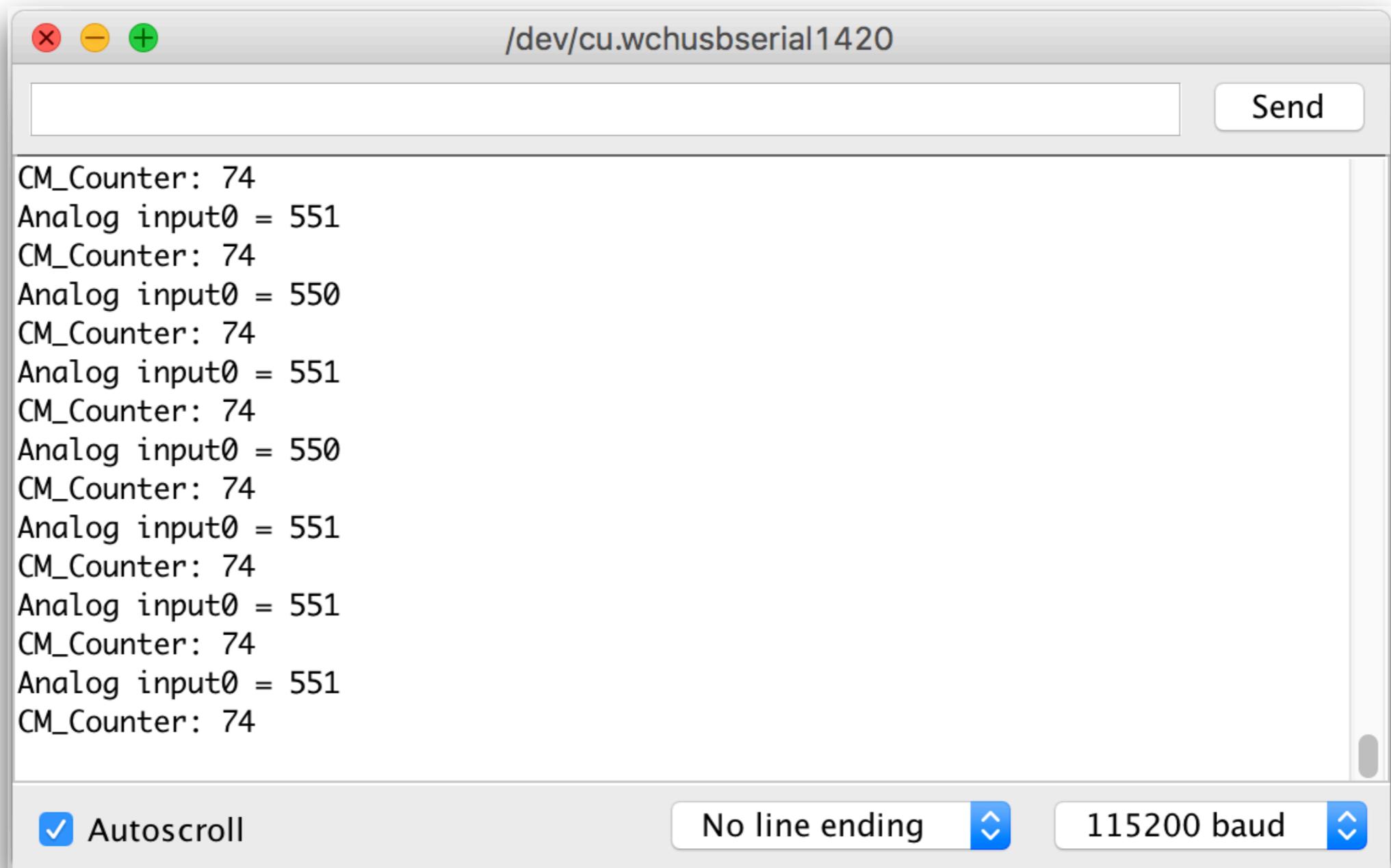
ในบรรทัดที่ 55 `analogValue = analogRead(ANALOG_PIN);`

คือการอ่านค่าตัวต้านทานปรับค่าได้

```
21 void loop(){
22     readAnalogValue();
23     setValueLCDAndSerialMonitor();
24 }
25 |
```

```
54 void readAnalogValue(){
55     analogValue = analogRead(ANALOG_PIN);
56     if(analogValue>value+5 ){
57         counter++;
58     }else
59     if(analogValue<value-5){
60         if(counter>0){
61             counter--;
62         }
63     }
64 }
```

ผลลัพธ์จาก Serial Monitor



The screenshot shows a Mac OS X-style application window titled "Serial Monitor". The title bar includes standard window controls (red, yellow, green) and the port name "/dev/cu.wchusbserial1420". The main pane displays the following text output:

```
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
Analog input0 = 550
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
Analog input0 = 550
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
Analog input0 = 551
CM_Counter: 74
```

At the bottom of the window, there are three configuration options: "Autoscroll" (checked), "No line ending" (selected), and "115200 baud".