**รายงาน Mini Project 5**

**หัวข้อ** การใช้จอ LCD กับ Arduino ในการแสดงผลต่างๆ

**โจทย์ (เลขคู่)**

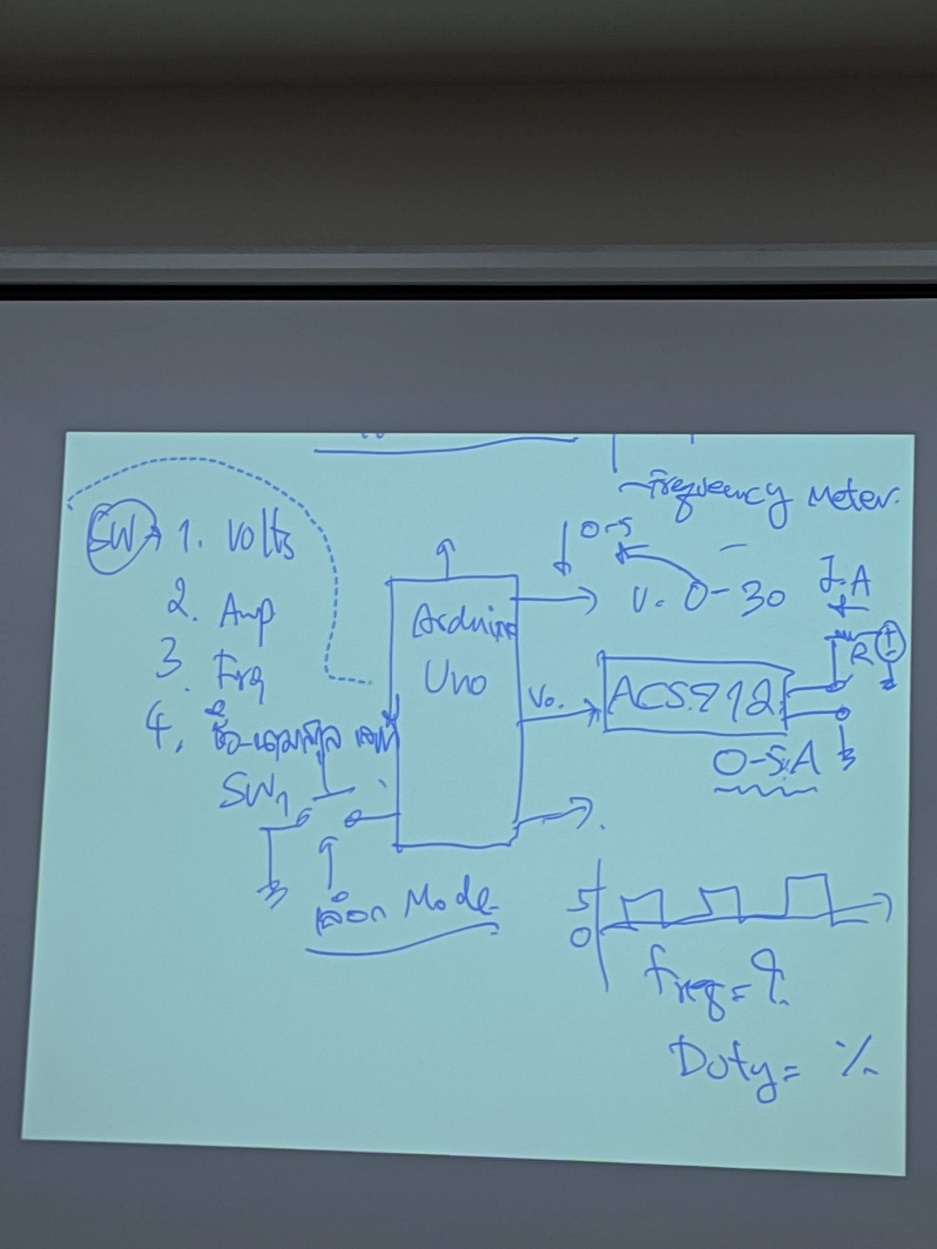
ทำการกดสวิตช์ 1 ครั้งเพื่อเปลี่ยนโหมดระหว่าง

1.Voltmeter (0-30 V)

2.Ampmeter (0 – 3 A)

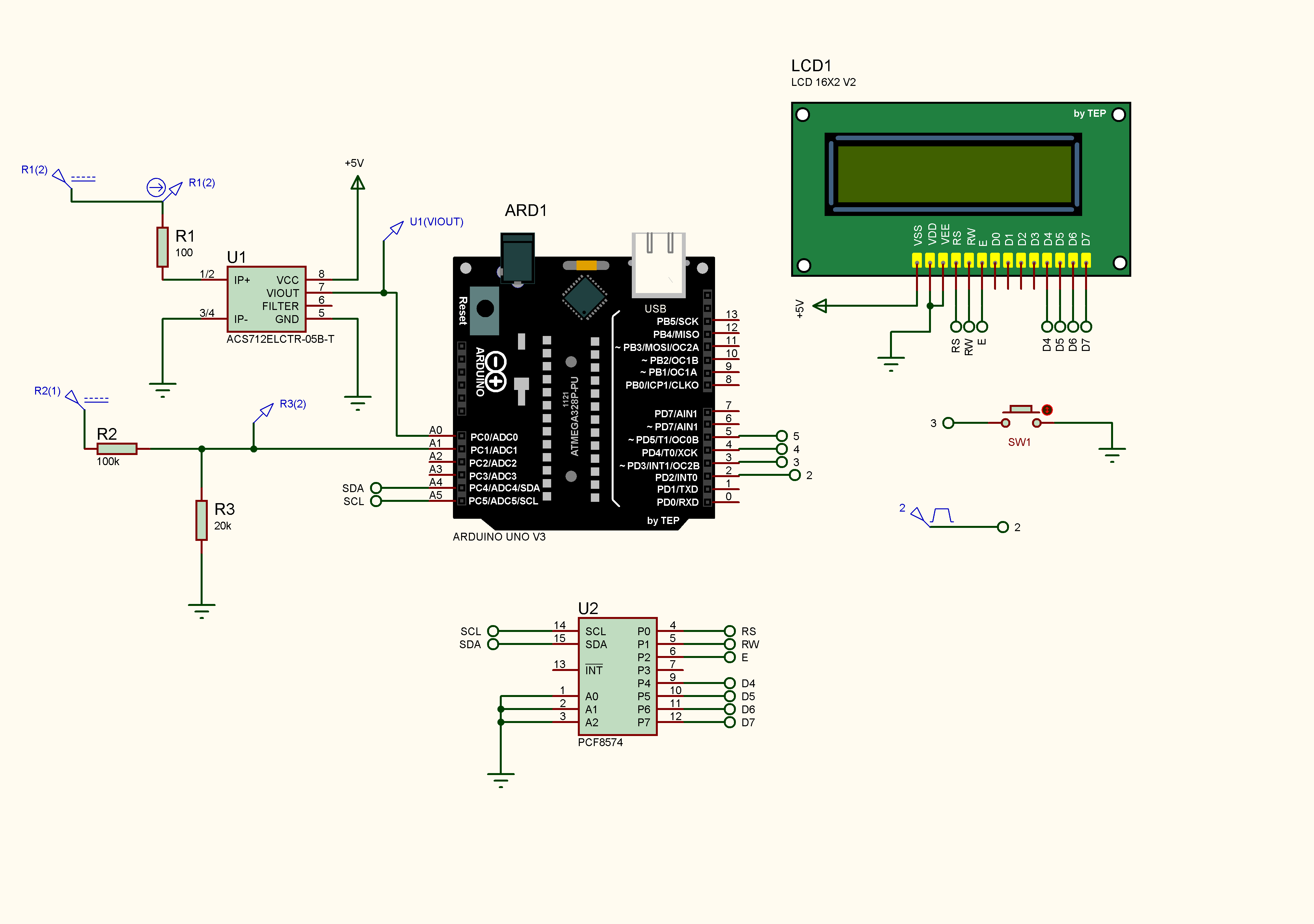
3.Frequency measurement (1 – 10 kHz) & PWM Duty cycle

4.แสดงชื่อตัวเอง



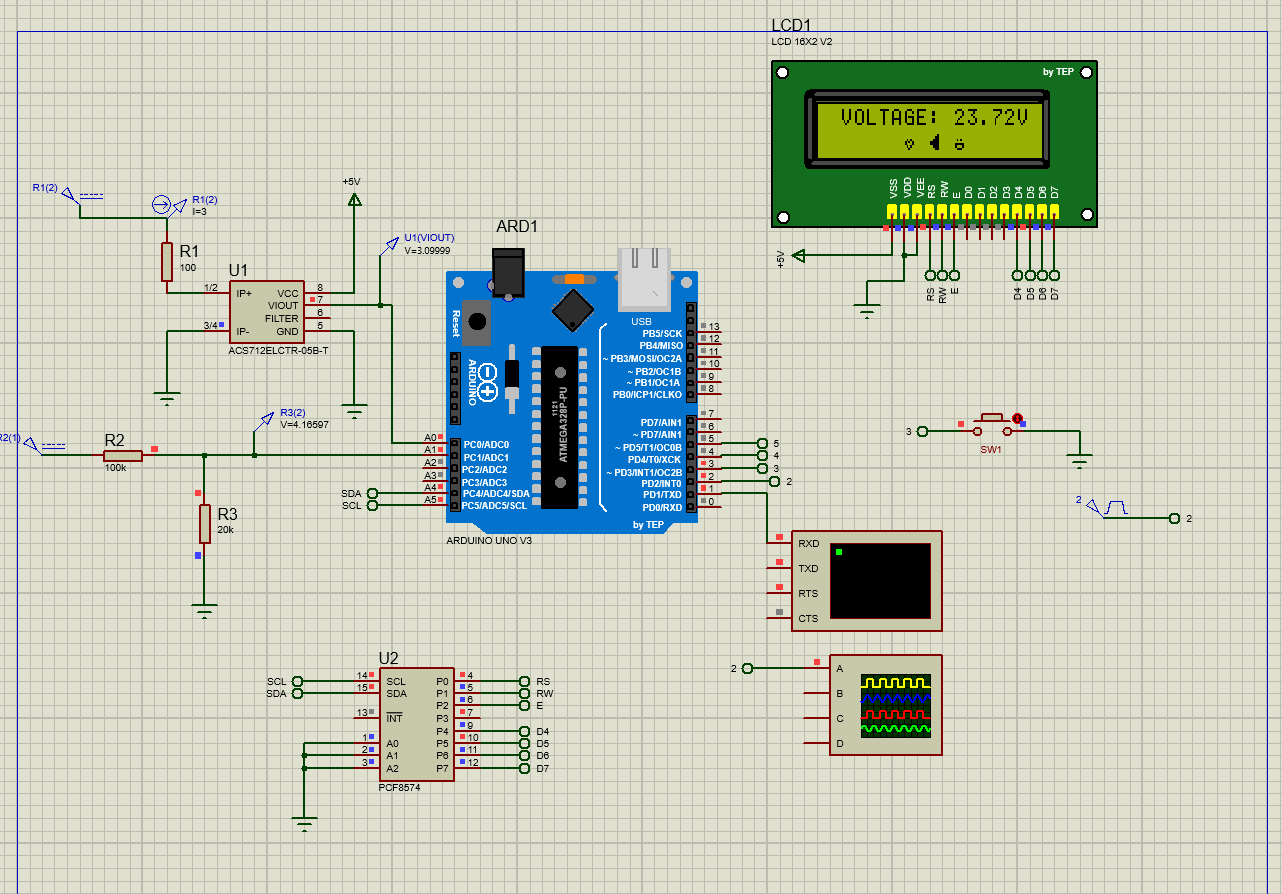
**อุปกรณ์ที่ใช้**

1. Arduino UNO R3
2. LCD 16x2
3. PCF8574
4. Resistor
5. Signal generator
6. Oscilloscope
7. DC power supply



รูปวงจรที่ใช้

**ผลการจำลองในโปรแกรม Proteus**

****

**A computer circuit board with many wires

Description automatically generated with medium confidence**

**A computer circuit board with many wires

Description automatically generated**

**A computer circuit board with wires and a display

Description automatically generated with medium confidence**

**โค้ดโปรแกรม Arduino ที่ใช้ควบคุม**

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x20, 16, 2); // 0x20 for Proteus 0x27 for Real LCD

uint8\_t HeartChar[] = { 0x00, 0x00, 0x0a, 0x15, 0x11, 0x0a, 0x04, 0x00 };

uint8\_t SpeakerChar[] = { 0x01, 0x03, 0x07, 0x1f, 0x1f, 0x07, 0x03, 0x01 };

uint8\_t SmileyFaceChar[] = { 0x00, 0x00, 0x0a, 0x00, 0x1f, 0x11, 0x0e, 0x00 };

uint8\_t BellChar[] = { 0x04, 0x0e, 0x0a, 0x0a, 0x0a, 0x1f, 0x00, 0x04 };

uint8\_t Battery1Char[] = { 0x0e, 0x1b, 0x11, 0x11, 0x11, 0x11, 0x11, 0x1f };

uint8\_t Battery2Char[] = { 0x0e, 0x1b, 0x11, 0x11, 0x11, 0x11, 0x1f, 0x1f };

uint8\_t Battery3Char[] = { 0x0e, 0x1b, 0x11, 0x11, 0x11, 0x1f, 0x1f, 0x1f };

uint8\_t Battery4Char[] = { 0x0e, 0x1b, 0x11, 0x1f, 0x1f, 0x1f, 0x1f, 0x1f };

const int voltagePin = A1;

const int currentPin = A0;

const int frequencyPin = 2;

const int sw1 = 3;

const int inputPin = 2;

int mode = 0;

volatile unsigned long pulseCount = 0;

unsigned long previousMillis = 0;

unsigned long interval = 500;

int ontime, offtime, duty;

float freq, period;

void setup() {

lcd.begin(16, 2);

lcd.backlight();

lcd.createChar(0, HeartChar);

lcd.createChar(1, SpeakerChar);

lcd.createChar(2, SmileyFaceChar);

lcd.createChar(3, BellChar);

lcd.createChar(4, Battery1Char);

lcd.createChar(5, Battery2Char);

lcd.createChar(6, Battery3Char);

lcd.createChar(7, Battery4Char);

lcd.setCursor(4, 0);

lcd.print("BOOTING!");

lcd.setCursor(0, 1);

lcd.write(byte(0));

lcd.setCursor(2, 1);

lcd.write(byte(1));

lcd.setCursor(4, 1);

lcd.write(byte(2));

lcd.setCursor(6, 1);

lcd.write(byte(3));

lcd.setCursor(8, 1);

lcd.write(byte(4));

lcd.setCursor(10, 1);

lcd.write(byte(5));

lcd.setCursor(12, 1);

lcd.write(byte(6));

lcd.setCursor(14, 1);

lcd.write(byte(7));

delay(1000);

lcd.clear();

pinMode(voltagePin, INPUT);

pinMode(currentPin, INPUT);

pinMode(frequencyPin, INPUT);

pinMode(sw1, INPUT\_PULLUP);

Serial.begin(9600);

}

void loop() {

if (digitalRead(sw1) == LOW) {

mode += 1; // mode change

delay(200); // Debounce delay

}

switch (mode) {

case 0:

measureVoltage();

break;

case 1:

measureCurrent();

break;

case 2:

measureFrequency();

break;

case 3:

showName();

break;

case 4:

resetMode();

break;

}

}

void measureVoltage() {

int voltageValue = analogRead(voltagePin);

float voltageOut = voltageValue \* (5.0 / 1023.0); // Assuming 5V Arduino

float voltageIn = voltageOut \* 5.69;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("VOLTAGE: ");

lcd.print(voltageIn, 2); // Display two decimal places

lcd.print("V");

lcd.setCursor(6, 1);

lcd.write(byte(0));

lcd.setCursor(8, 1);

lcd.write(byte(1));

lcd.setCursor(10, 1);

lcd.write(byte(2));

delay(400);

}

void measureCurrent() {

// Calibration parameters

float offsetVoltage = 2.5; // Offset voltage when no current is flowing

float sensitivity = 0.185; // Sensitivity, in volts per ampere

float calibrationFactor = 0.775; // Calibration factor to adjust the reading

// Read the raw ADC value

int currentRawValue = analogRead(currentPin);

// Map the raw ADC value to voltage (0V to 5V)

float voltage = currentRawValue \* (5.0 / 1023.0);

// Subtract the offset voltage (2.5V when 0A)

voltage -= offsetVoltage;

// Convert voltage to current based on the sensitivity

float current = (voltage / sensitivity) \* calibrationFactor;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("CURRENT: ");

lcd.print(current, 2); // Display current with two decimal places

lcd.print("A");

lcd.setCursor(6, 1);

lcd.write(byte(3));

lcd.setCursor(8, 1);

lcd.write(byte(5));

lcd.setCursor(10, 1);

lcd.write(byte(6));

delay(400);

}

void measureFrequency() {

unsigned long onTime = pulseIn(frequencyPin, HIGH);

unsigned long offTime = pulseIn(frequencyPin, LOW);

unsigned long period = onTime + offTime;

float frequency = 0.0;

float dutyCycle = 0.0;

if (period > 0) {

frequency = 1000000.0 / period;

dutyCycle = (onTime / (float)period) \* 100.0;

}

lcd.clear();

lcd.setCursor(2, 0);

lcd.print("FREQ: ");

if (frequency > 1000) {

lcd.print(((float)frequency / 1000.0), 1);

}

else {

lcd.print((int)frequency);

}

lcd.print(" kHz");

lcd.setCursor(2, 1);

lcd.print("DUTY: ");

lcd.print(dutyCycle, 2);

lcd.print("%");

delay(400);

}

void showName() {

lcd.clear();

lcd.setCursor(4, 0);

lcd.print("65010386");

lcd.setCursor(4, 1);

lcd.print("THONGCHAI");

lcd.setCursor(2, 0);

lcd.write(byte(0));

lcd.setCursor(2, 1);

lcd.write(byte(1));

lcd.setCursor(14, 0);

lcd.write(byte(2));

lcd.setCursor(14, 1);

lcd.write(byte(7));

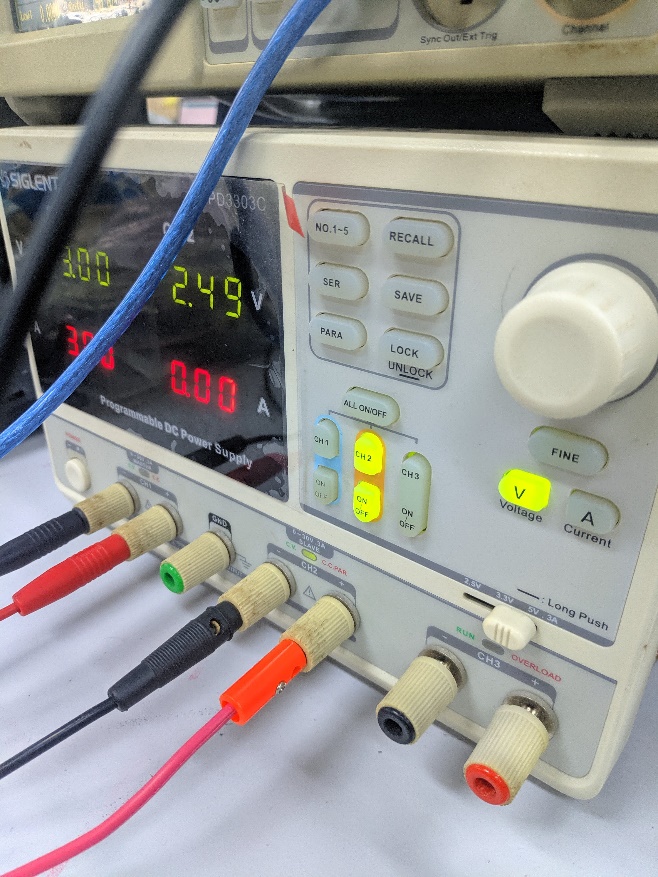
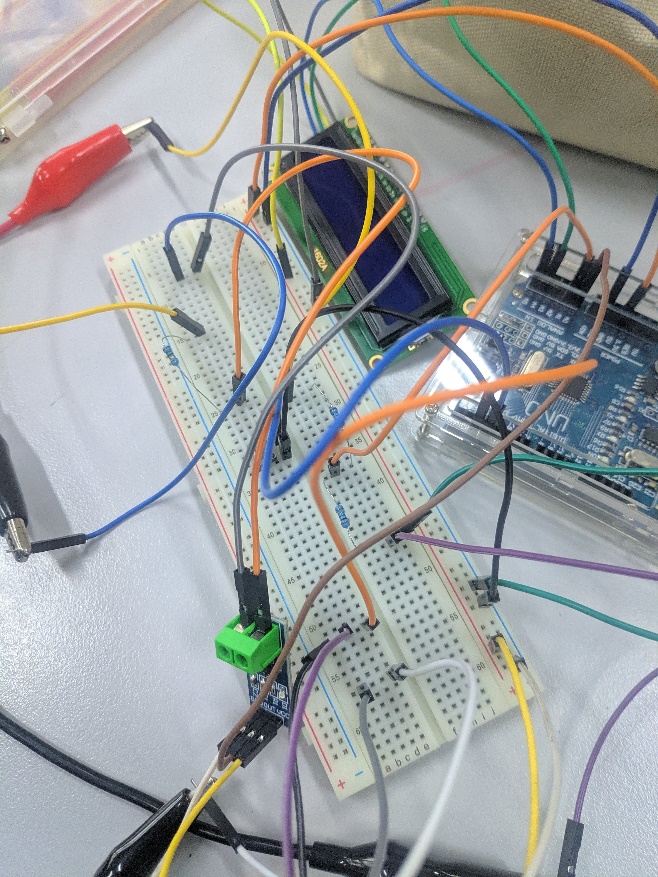
delay(400);

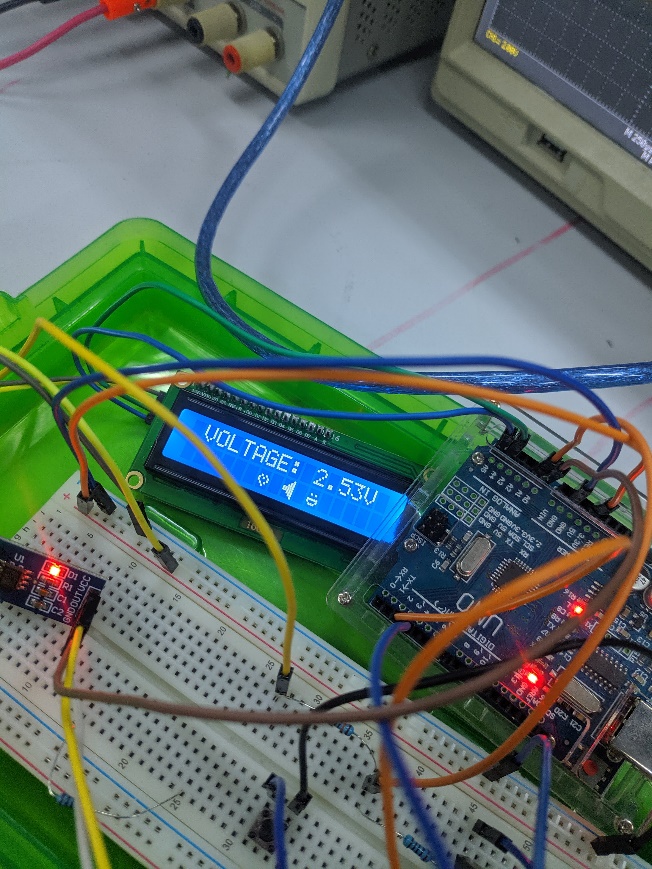
}

void resetMode() {

mode = 0;

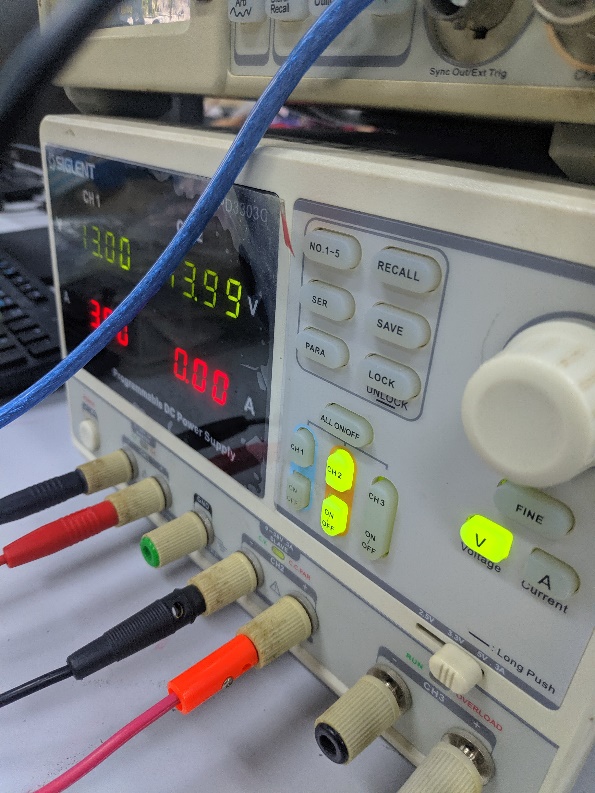
}

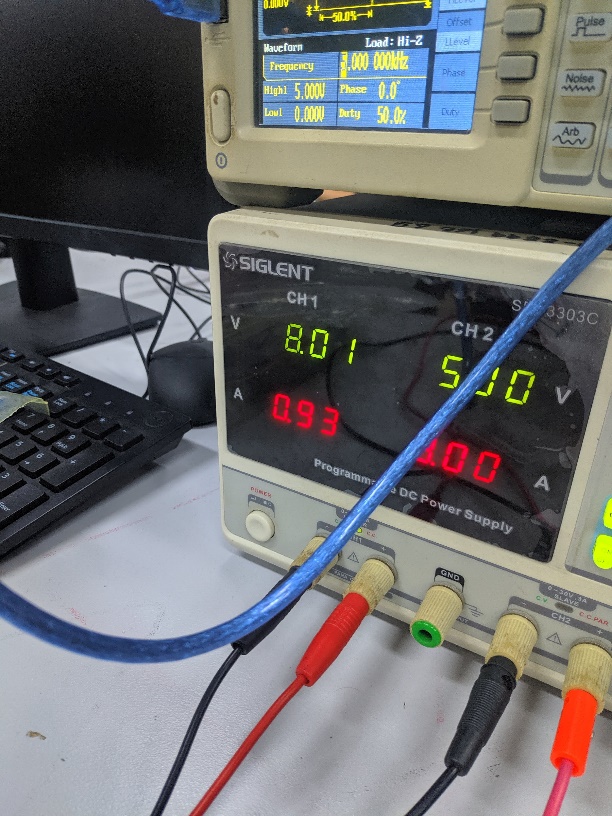
**ผลการต่อจริง**

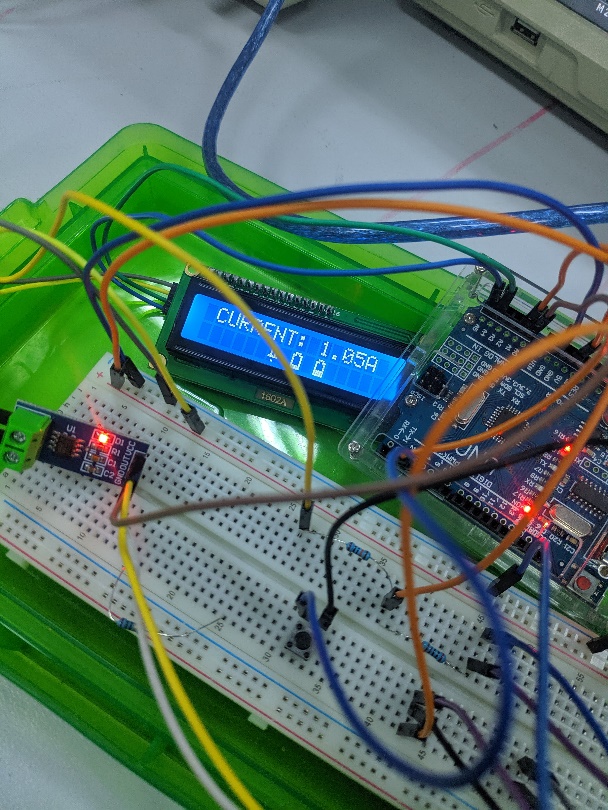


ผลที่ได้เมื่อจ่ายไฟ 2.5V เข้าไปให้ Arduino ผ่าน Voltage divider โดยวัดไฟได้ 2.53V แล้วแสดงออกมาทาง LCD

A circuit board with wires

Description automatically generated

จ่ายไฟ 14V อ่านค่าออกมาได้ 13.57V



นำ load มาต่อให้กินกระแส 1A วัดได้ 1.05A

A close-up of a device

Description automatically generatedA close up of a device

Description automatically generated

A circuit board with wires

Description automatically generated

จ่ายสัญญาณ PWM 5V Duty cycle 50% เข้าไป

อ่านค่าได้ 5 kHz Duty cycle 50%

A close-up of a monitor

Description automatically generatedA close up of a device

Description automatically generated

A green electronic device with wires

Description automatically generated

จ่ายสัญญาณ PWM 1 kHz duty cycle 80%

วัดได้ 1 kHz duty cycle 80%

A circuit board with wires

Description automatically generated

แสดงชื่อตัวเอง