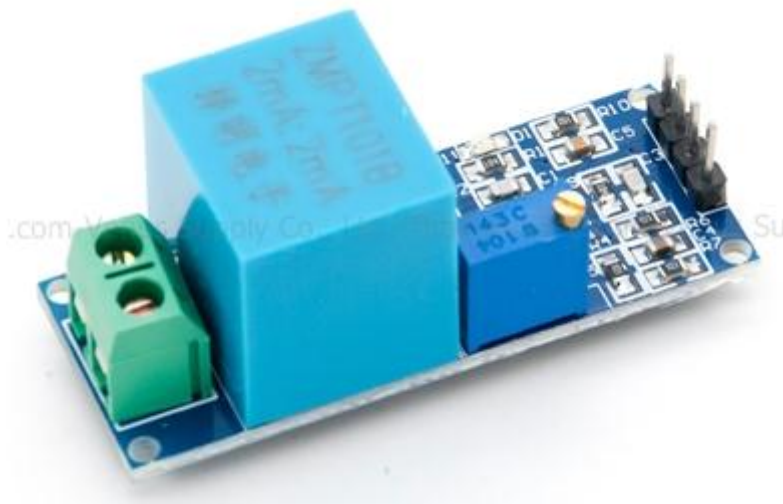
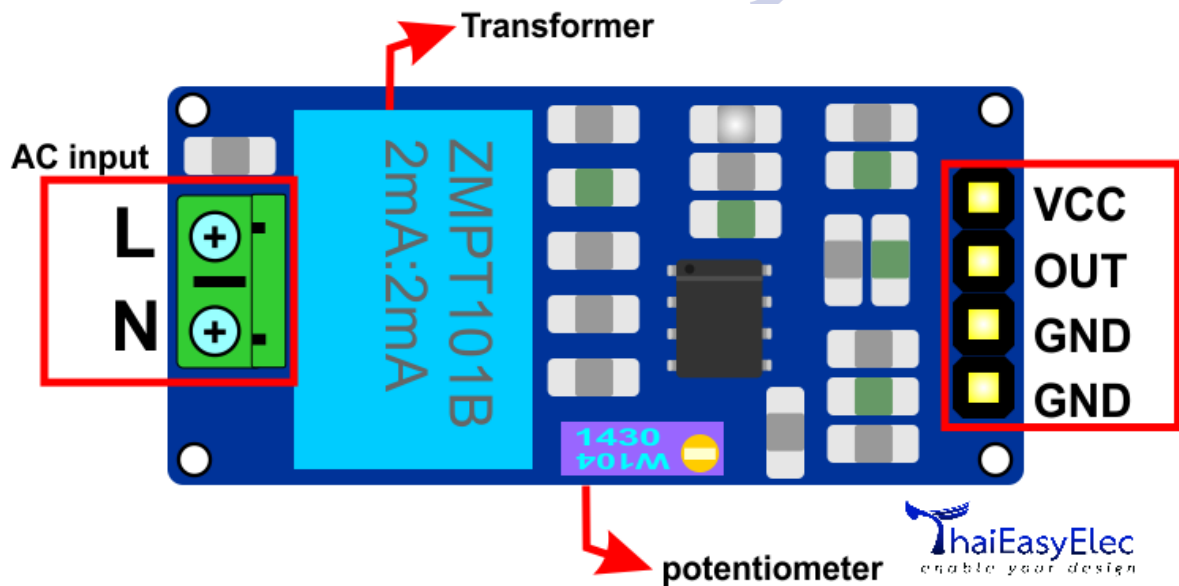


Single phase voltage sensor

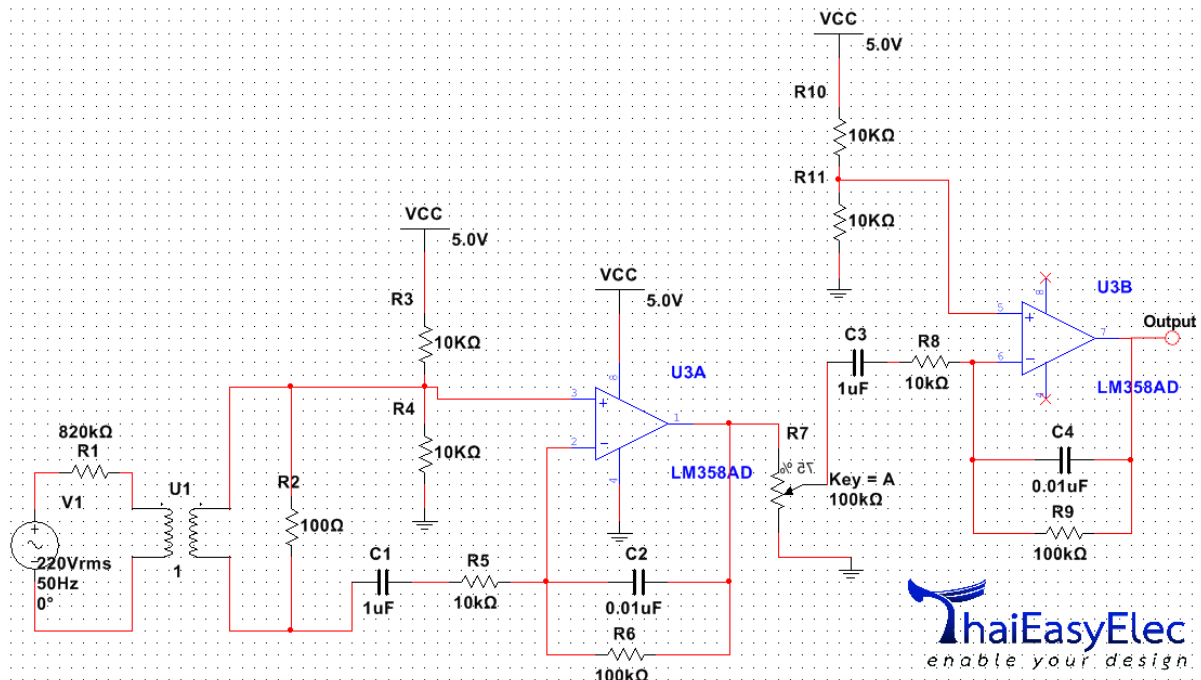


โมดูลเซ็นเซอร์วัดแรงดันไฟ AC สูงสุดที่ 250VAC สัญญาณที่ออกจากโมดูลเป็นสัญญาณอนาล็อก สามารถนำไปต่อเข้ากับขา ADC ของไมโครคอนโทรลเลอร์ที่ใช้ Vref +5V ได้ทันทีเช่น Arduino Uno, Mega, Leonardo มีวงจรขยายสัญญาณ สามารถปรับขนาดแอมพลิจูดของสัญญาณเอาต์พุตได้ จากการปรับตัวต้านทานปรับค่าแบบจูน

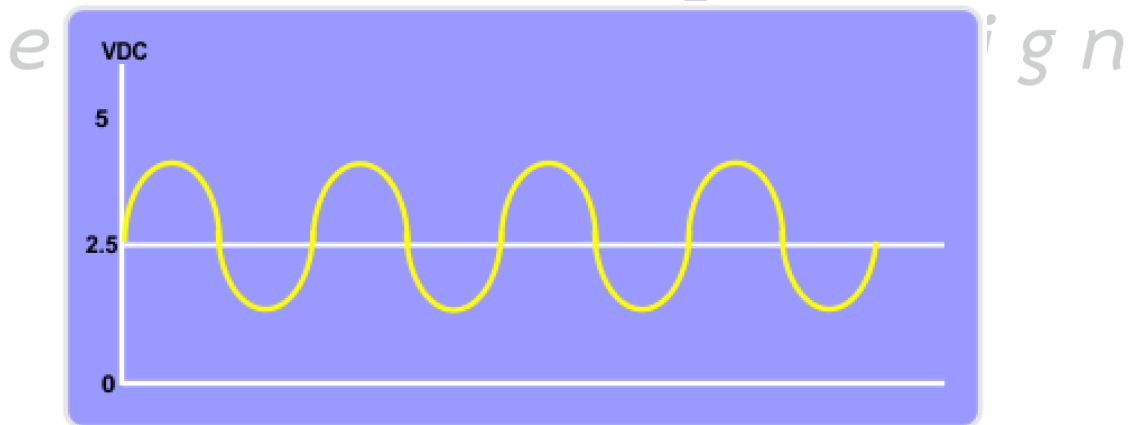
Hardware



วงจรภายใน Module (Unofficial)

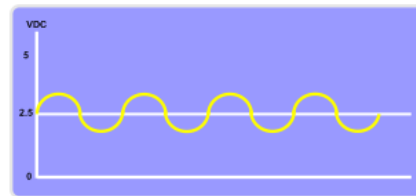
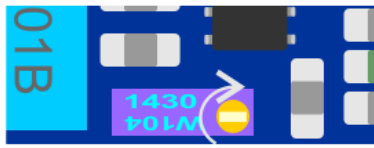


รูปแบบสัญญาณที่ได้จาก single phase voltage sensor

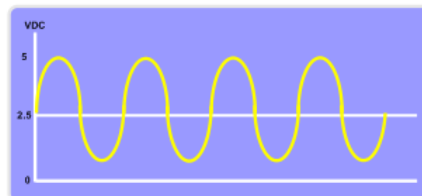


รูปแบบสัญญาณที่ได้จาก Output ของ single phase voltage sensor เมื่อนำไปวัดไฟ AC 220V ที่ใช้ตามบ้าน แล้วใช้ Oscilloscope จับสัญญาณจะเห็นรูปแบบสัญญาณ sin ที่ขี้อยู่บนไฟ DC ประมาณ 2.5 VDC ซึ่งเราสามารถปรับค่า Amplitude ของสัญญาณ sin ได้โดยการปรับ potentiometer

- หมุน potentiometer ในทิศทางตามเข็มนาฬิกา จะทำให้ Amplitude ลดลง



- หมุน potentiometer ในทิศทางทวนเข็มนาฬิกา จะทำให้ Amplitude เพิ่มขึ้น

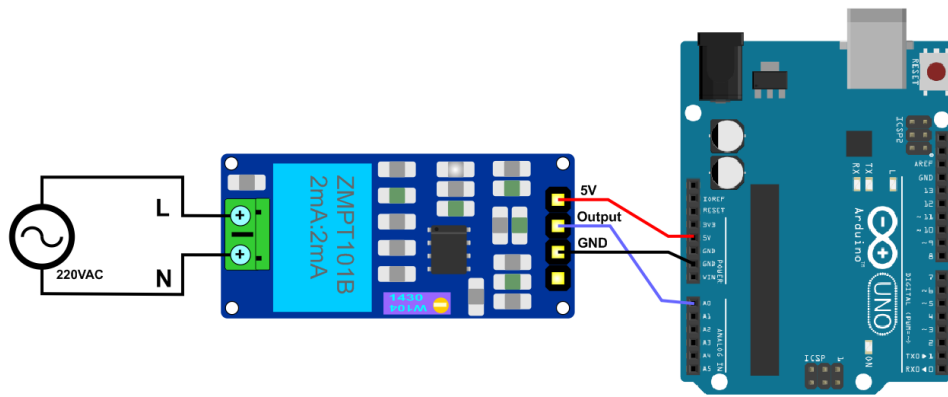


ขั้นตอนการปรับค่า Amplitude ให้กับ single phase voltage sensor

การใช้งาน single phase voltage sensor จำเป็นต้องมี Oscilloscope จับสัญญาณจะเห็นรูปสัญญาณเพื่อปรับ Amplitude ให้อยู่ในค่าที่เหมาะสมเพื่อนำไปใช้คำนวณ เพื่อหาค่า Voltage จริงต่อไป แต่เนื่องจากสัญญาณที่ได้จาก output ของ single phase voltage sensor เป็นสัญญาณที่ไม่ได้มีความถี่สูงมากนัก เราจึงเขียนโปรแกรม “ThaiEasyElec Single Phase Voltage sensor” มาเพื่อแสดงผล การปรับ Amplitude แทน Oscilloscope โดยใช้ Arduino อ่านค่า ADC จากสัญญาณ Output ของ Single Phase Voltage sensor แล้วส่งค่า ADC ที่ได้มาแสดงผลบนโปรแกรม



- วิธีการต่อ single phase voltage sensor กับ Arduino UNO R3



โปรแกรม Code สำหรับอ่าน ADC จาก single phase voltage sensor

```
#define BUFFER 700
void setup()
{
  Serial.begin(115200);
}
void loop()
{
  if(Serial.available())
  {
    String req = Serial.readStringUntil("\n");
    if(req.indexOf(F("start")) != -1)
    {
      unsigned char flag=0;
      while(1)
      {
        int x = analogRead(A0);

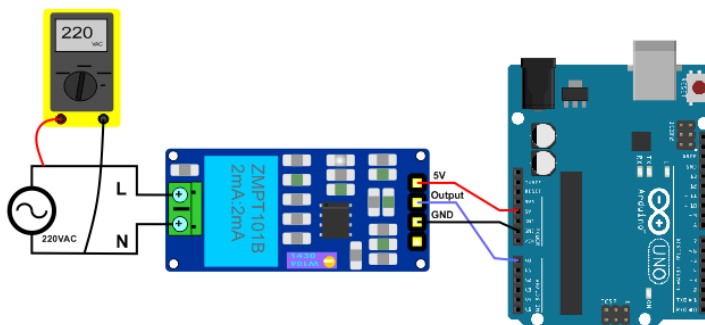
        if(x<512)
        {
          flag=1;
        }
        if((x>=512)&&(flag==1))
        {
          break;
        }
      }
      for(int i=0;i<BUFFER;i++)
      {
        int data = analogRead(A0);
        Serial.write(data>>8);
        Serial.write(data&0xFF);
      }
      Serial.print("end");
    }
  }
}
```

- เปิดโปรแกรม “ThaiEasyElec Single Phase Voltage sensor” ขึ้นมาเลือก Com port ไปที่ช่อง Com port ของ Arduino
- คลิก Button Open เพื่อเปิด Com port
- คลิก Button Start เพื่อเริ่มอ่านค่า ADC มาแสดงผล
- ปรับค่า potentiometer ให้ได้ Amplitude ที่เหมาะสม



ตัวอย่างการใช้งาน Detect ไฟ AC

- วัดค่า AC Voltage Input โดยใช้ MultiMeter และ จดบันทึกค่าเอาไว้ (ในที่นี้ขอสมมติว่าเป็น 233.5VAC)



- ทำการปรับค่า Amplitude ตามหัวข้อ ขั้นตอนการปรับค่า Amplitude ให้กับ single phase voltage sensor ให้ได้ค่า Amplitude ที่เหมาะสม (ในที่นี้ได้กำหนดให้ Amplitude = 2V หรือ 411 ใน ADC 10 Bits)



- เขียนโปรแกรมดังนี้

```
#define SAMPLING 300 //กำหนดจำนวนการสุ่มค่า
#define VOFFSET 512 // 2.5V จาก 10Bit
#define ADC_PIN A0 //กำหนดรียค่า Analog จาก Pin A0
#define AMPLITUDE 411.00 //กำหนดค่า Amplitude ที่ได้จากแอมแกรม
#define REAL_VAC 233.5 //กำหนดค่า Vinput ที่อ่านได้จริงจาก Multimeter

int adc_max,adc_min;
int adc_vpp;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  read_VAC(); //เรียกใช้งาน Function อ่านค่า V
  float V = map(adc_vpp,0,AMPLITUDE,0,REAL_VAC*100)/100.00; // แปลงค่าที่อ่านได้เป็น
  //VAC

  String data;
  data = "Volt = "+String(V,1)+" VAC";
  Serial.println(data);
  delay(200);
}
```

```

void read_VAC()
{
    int cnt;
    adc_max = 0;
    adc_min = 1024;

    for(cnt=0;cnt<SAMPLING;cnt++)    //วน loop อ่านค่า ADC
    {
        int adc = analogRead(ADC_PIN); //อ่านค่า ADC
        if(adc > adc_max)                //หาค่า max
        {
            adc_max = adc;
        }
        if(adc < adc_min)                //หาค่า min
        {
            adc_min = adc;
        }
    }
    adc_vpp = adc_max-adc_min;          //หาผลต่างของ input (Vpp)
}

```

ทดลอง Run Program

The screenshot shows the Arduino IDE interface. The main window displays the code for 'Ex_SinglePhaseVoltageSensor'. The code defines constants for sampling rate, offset, pin, and amplitude, and implements a loop that reads the ADC value and calculates the peak-to-peak voltage (Vpp). A serial monitor window titled 'COM31' is open, showing the output of the program. The output consists of multiple lines of 'Volt = ' followed by a numerical value and 'VAC'. The values fluctuate between approximately 232.9 and 234.1 VAC. The serial monitor settings are set to 9600 baud and Carriage return.

```

Ex_SinglePhaseVoltageSensor | Arduino 1.6.5
File Edit Sketch Tools Help
Ex_SinglePhaseVoltageSensor$
1 #define SAMPLING 300
2 #define VOFFSET 512 // 2.5V จาก 10Bit
3 #define ADC_PIN A0
4 #define AMPLITUDE 411.00
5 #define REAL_VAC 233.5
6 int adc_max,adc_min;
7 int adc_vpp;
8 void setup()
9 {
10     Serial.begin(9600);
11 }
12
13 void loop()
14 {
15     read_VAC();
16     float V = map(adc_vpp,0,AMPLITUDE,0,REAL_VAC);
17     String data;
18     data = "Volt = "+String(V,1)+" VAC";
19     Serial.println(data);
20 }
Done uploading.
Global variables use 234 bytes (11%) of dynamic memory, leaving
1,814 bytes for local variables. Maximum is 2,048 bytes.
COM31
Volt = 233.5 VAC
Volt = 234.1 VAC
Volt = 233.5 VAC
Volt = 232.9 VAC
Volt = 233.5 VAC
Volt = 234.1 VAC
Volt = 233.5 VAC
Volt = 233.5 VAC
Volt = 233.5 VAC
Volt = 234.1 VAC
Volt = 233.5 VAC
Volt = 233.5 VAC
Volt = 233.5 VAC
Volt = 234.1 VAC
Volt = 233.5 VAC
Volt = 232.9 VAC
Volt = 233.5 VAC
Autoscroll Carriage return 9600 baud

```