คู่มือทางด้านเทคนิค (Technical Manual)

เครื่องมือวัดอัตราการไหลแบบพกพา (Portable Water Flow Meter Sensor)

จัดทำโดยนิสิตสาขาวิชาวิศวกรรมโยธา-ชลประทาน

1. นาย กรันต์ พรวุฒิพงศ์ 6120500880
2. นาย ศุภกิตติ์ ยิ้มพงศ์ 6120501045
3. นาย อานนท์ เหมลา 6120501126

เครื่องมือนี้เป็นผลงานจากวิชาปัญหาพิเศษ 02207489

ภาคปลาย ปีการศึกษา 2563

ภาควิชาวิศวกรรมชลประทาน คณะวิศวกรรมศาสตร์ กำแพงแสน

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

พุทธศักราช 2564

คำนำ

คู่มือเล่มนี้เป็นส่วนหนึ่งของเครื่องมือวัดอัตราการไหลแบบพกพา ในปัจจุบันการทำการวัดอัตราการไหลของน้ำเพื่อเป็นประโยชน์ในด้านการเกษตร การคำนวณการให้น้ำเป็นสิ่งที่มีความสำคัญมากเพราะอัตราผลผลิตขึ้นอยู่กับการให้น้ำที่เพียงพอถ้าให้น้ำไม่เพียงพอก็จะส่งผลให้ผลผลิตลดลงหรือเสียหายได้ดังนั้นการคำนวณการให้น้ำจึงเป็นสิ่งที่สำคัญอย่างยิ่งในด้านการปลูกพืชหรือทำการเกษตรต่าง ๆ เนื่องจากการให้น้ำในปัจจุบันส่วนใหญ่จะใช้การวัดจากมาตรวัดน้ำแบบดั้งเดิมซึ่งเราจะเสียเวลาไปอยู่หน้างานโดยการใช้มาตรวัดแบบระบบอนาล็อก(แบบใบพัด) ดังนั้นทางทีมงานของเราจึงเล็งเห็นความสำคัญในการให้น้ำเราจึงได้ประดิษฐ์เครื่องมือการวัดอัตราการไหลแบบพกพาขึ้นมาเพื่อความสะดวกสบายในกรณีที่เราไม่ได้อยู่ ณ สถานที่ให้น้ำแต่เราสามารถที่จะดูว่าเราให้น้ำในอัตราเท่าใดให้น้ำเพียงพอหรือไม่ จะได้นำข้อมูลอัตราการให้น้ำไปวิเคราะห์เพื่อเพิ่มผลผลิตและเพิ่มความสะดวกสะบายในการตรวจสอบการให้น้ำต่อไป

ข้อดี สะดวกสบาย ประยุกต์ใช้กับการให้น้ำอัตโนมัติได้ ไม่ต้องอยู่ ณ สถาณที่เพาะปลูกก็สามารถดูอัตราการให้น้ำแบบออนไลน์ได้สะดวกทุกที่ ทุกเวลา

ข้อจำกัด ค่าความละเอียดอาจจะมีความคลาดเคลื่อนได้ควรปรับเทียบค่าเดือนล่ะ 1 ครั้ง ระบบอาจจะมีการดูแลรักษายากกว่าแบบใบพัด ผู้ใช้ต้องมีความรู้เบื้องต้นในการดูแลและแก้ไขอุปกรณ์ที่ชำรุดได้

คณะผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คำนำ	ก
สารบัญ	ข
1. ขั้นตอนการออกแบบและการสร้าง	1
2. รายละเอียดเครื่องมือและอุปกรณ์	3
3. ภาพแสดงการประกอบ Sensor	7
4. โปรแกรมที่ใช้งานและCode	8
5. แหล่งซื้ออุปกรณ์และราคา	14
6. ข้อเสนอแนะในการพัฒนาต่อไป	15
บรรณานุกรม	16
ภาคผนวก	
- ภาคผนวก ก	17

1. ขั้นตอนการออกแบบและการสร้าง

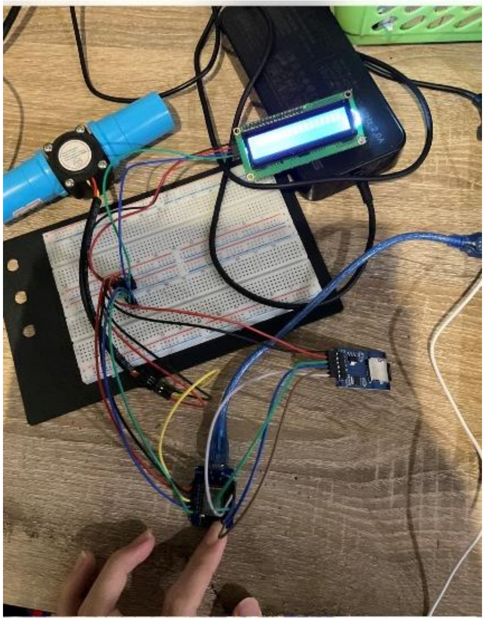
1. ตั้งปัญหาและศึกษาข้อมูลที่จะทำโครงงาน

2. จัดซื้ออุปกรณ์ต่างๆที่ใช้ในการออกแบบโครงงาน

3. ใช้ Arduino IDE ทำการเขียน Code และทำการ Compile ในเพื่อทดสอบ Sensor ทำงานได้ปกติหรือไม่

4. ทำการเชื่อมต่อ Sensor เข้ากับแอพพลิเคชัน Blynk ผ่านSmartphone

5. ทำการออกแบบอุปกรณ์และทำการปรับแก้ Code แล้วทำการติดตั้งอุปกรณ์

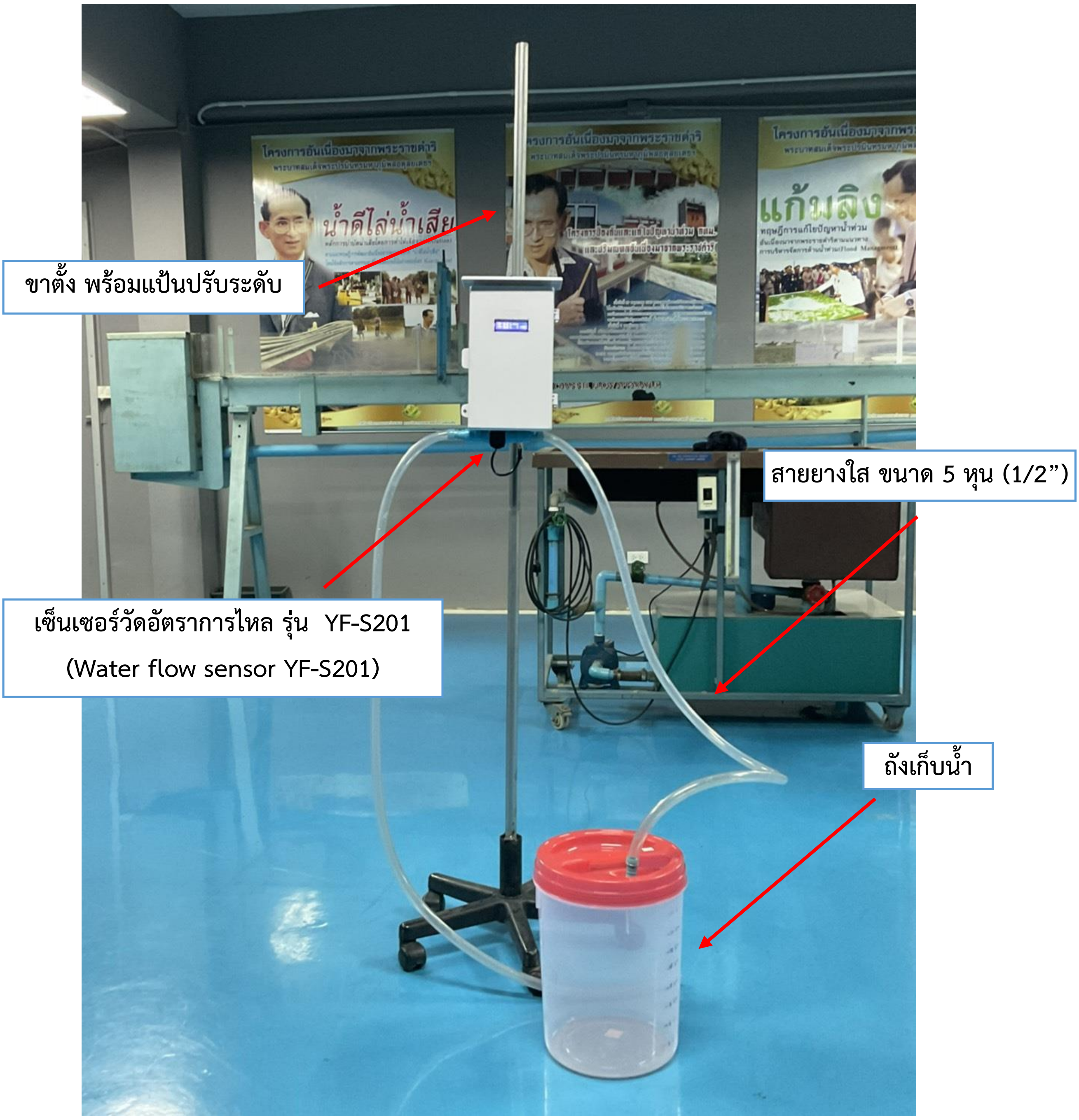


ภาพที่ 1 ปรับแก้ Code

ภาพที่ 2 อุปกรณ์ภายในกล่อง

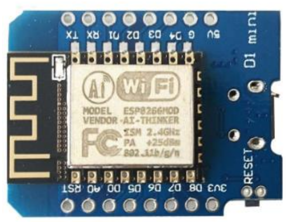
6. ทดสอบอุปกรณ์ ทำการปรับแก้ค่ากับอุปกรณ์ที่ได้มาตราฐาน

7.ทำการประกอบอุปกรณ์ทั้งหมดและนำอุปกรณ์ไปทดลองใช้งานจริง



ภาพที่ 3 รูปเครื่องมือวัดอัตราการไหลแบบพกพา

2. รายละเอียดเครื่องมือ

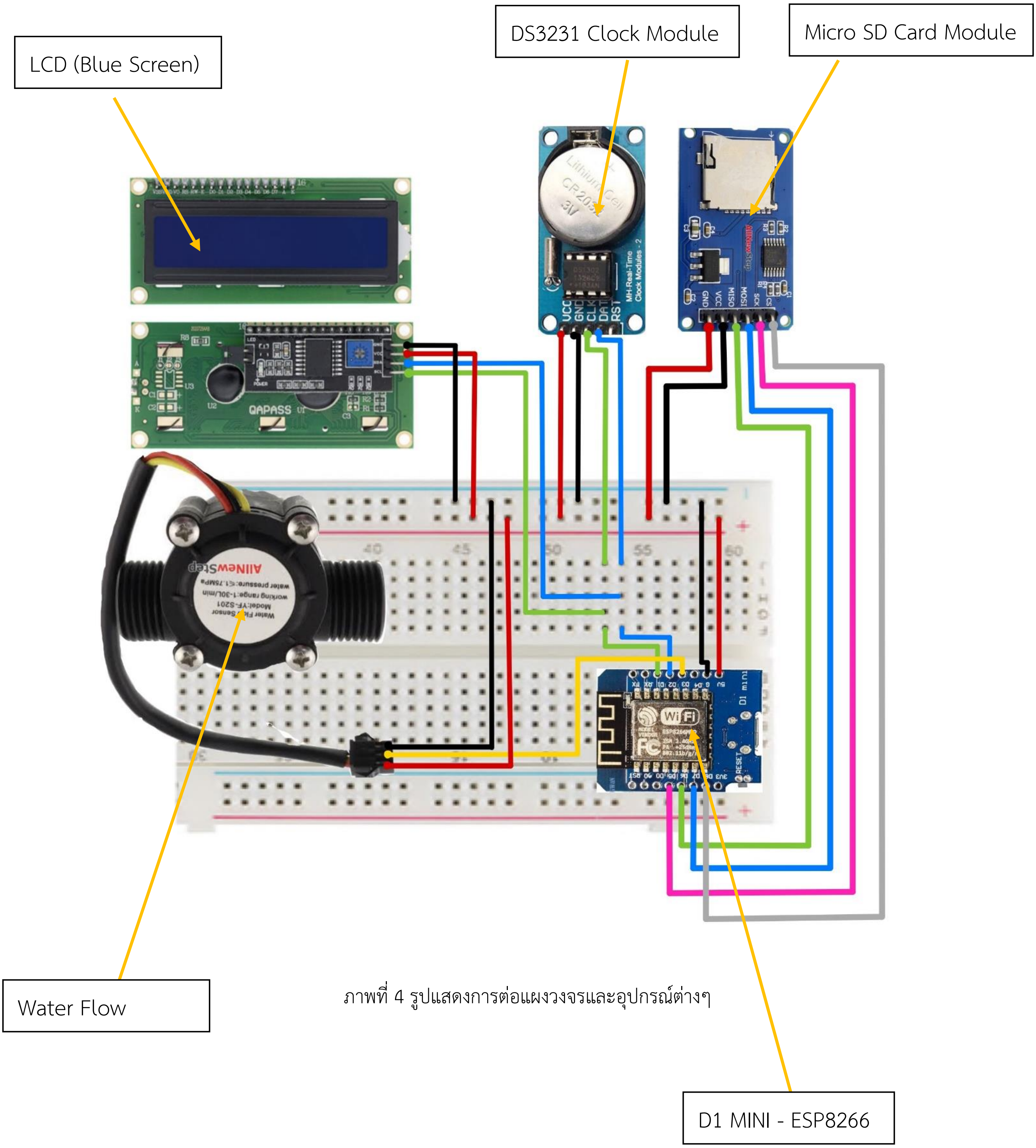
ชื่ออุปกรณ์	ข้อมูลเฉพาะ (Specifics data)	การทำงาน
1. D1 MINI - ESP8266 ESP12 NodeMcu Dev-Kit WiFi Modul Board WeMos for Arduino ภาพที่ 3 Wemos D1 Mini 	- Chip : ESP-8266EX - For Arduino compatible, programming with Arduino IDE - Compatible with Node MCU firmware - GPIO pins : 11 - 1 pin ADC (0V to 3.3V) - Max Input Voltage : 24V - Operating Frequency : 80/160 MHz - Flash Memory : 4MB - I2C interface - Size : 34.2 x 25.6 mm	ส่วนกลางควบคุมการทำงานของเซ็นเซอร์ต่างๆ แล้วสามารถส่งค่าผ่านไวไฟขึ้นไปยัง Blink server ได้
2. DS3231 Clock Module with Battery Real-Time Clock Module RTC for Arduino AVR ARM 	Clock operation by AM / PM indicator decided to use 24 or 12 hour format. DS1302 between the microcontroller can simply adopt the way synchronous serial communication port only used three lines: (1) RST Reset (2) I / O data lines (3) SCLK Serial Clock. Clock / RAM read / write data in a byte or up to 31 bytes of character set type communication 4.3x 2.3cm Package included: 1x DS1302 Module 1x CR2032 battery	เป็นอุปกรณ์ระบุเวลา ณ ปัจจุบัน

3. Micro SD Card Module 	รองรับ Micro SD Card และ High Speed Micro SD Card - ไฟเลี้ยง 4.5 - 5V ในโมดูลมีวงจรเหลือ 3.3V - is a standard communication interface SPI interface - M2 screw positioning holes for easy installation Control Interface: A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK is SPI bus, CS is the chip select signal pin; - ขนาด 24 x 42 mm	เก็บค่าข้อมูลลงบน SD Card
4. water flow sensor 	Working voltage: DC 5V-18V Fluid flow: 1-30L/min Maximum current: 15 mA (DC 5V) Operating temperature: -25 ~ + 80 ° Operating humidity: 35% ~ 90% RH (without icing) External thread: 1/2 " Output: 5V TTL The characteristics of the pulses: 450 pulses/liter or f (Hz) = 7,5* flow rate (l/min) Cable length: ~15 cm	เซนเซอร์วัดอัตราการไหล
5. 1602 LCD (Blue Screen) 16x2 LCD พร้อม บัดกรี I2C Adapter 	16 characters (5x8 dots) x 2 lines FSTN Positive Black White Transflective Dot Matrix LCD module. - Viewing Angle: 6 Oclock direction. - Driving scheme: 1/16 Duty, 1/5 bias. - HD44780 LCD Controller and Driver or equivalent. - 40-Channel Segment/Common Driver for Dot Matrix LCD or equivalent. - Connector: 16 pins ZIF SMD connector (CON-IL-402). - White LED05 backlight.	หน้าจอแสดงผล

6. micro sd card 	Form Factor: microSDHC, microSDXC - Capacity: 32GB - Performance: Up to 80MB/s read speed, write speed lower - Dimension: micoSDHC : 10.92 x 14.99 x 1.02 mm - Operation Temperature: -25 ℃ to 85 ℃ - Storage Temperature: -40 ℃ to 85 ℃	สำหรับเก็บข้อมูล
7. ข้อต่อตรงเกลียวใน ทองเหลือง 1/2"ตราช้าง scg 	ขนาด ½ นิ้วเกลียวทองเหลือง	สำหรับต่อกับ water flow sensor
8. สาย micro usb 	Micro USB Cable for arduino with DUE R3 D1 mini NodeMcu V3 TP4056 18650 Board 30CM	สำหรับส่งข้อมูลจากคอมพิวเตอร์เข้าบอร์ดควบคุม wemos mini d1
11. ตู้กันน้ำพลาสติกมีหลังคา Electric Enclosure with rain hood 	MATERIAL : ABS	11.ตู้กันน้ำพลาสติกมีหลังคา Electric Enclosure with rain hood

12. ข้อต่อสายยาง PVC 1/2" 	- ข้อต่อสายยางPVC แบบสวมกับท่อ 4หุน	สำหรับต่อกับข้อต่อตรงเกลียวใน
13. สายยาง 5 หุน (5/8") 	- มีความยืดหยุ่นตัวสูง ทนต่อสภาพ อากาศได้ดี - ไม่แข็งตัว และกรอบแตกง่าย - ทนต่อแรงกระแทก - สามารถม้วนเก็บได้ง่าย พกพา สะดวก - ขนาด 5/16 นิ้ว x 20 ม.	สำหรับต่อกับข้อต่อสายยาง PVC 1/2"
14. ถังน้ำพลาสติก 	- ถังพลาสติกความจุ 16 ลิตร - ขนาดเส้นผ่าศูนย์กลาง 28 * 37 cm	สำหรับเก็บน้ำ

3. ภาพแสดงการประกอบ Sensor



ภาพที่ 4 รูปแสดงการต่อแผงวงจรและอุปกรณ์ต่างๆ

4. โปรแกรมที่ใช้งานและCode

- โปรแกรมที่ใช้งาน Arduino IDE 1.8.13

- Code ในการเขียนโปรแกรม Arduino IDE

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
#include <SimpleTimer.h>
#include "RTClib.h"
// Libraries for SD card
#include "FS.h"
#include "SD.h"
#include <SPI.h>

const int sensorPin = D3;   // WeMos & NodeMCU D3

///// set AUTH,WIFI
char auth[] = "****************************";
char ssid[]  = "****************************";
char pass[] = "****************************";
```

```
// The hall-effect flow sensor outputs approximately 4.5 pulses per second per
litre/minute of flow.

float calibrationFactor = 1;//4.5, 7.5 ,1
volatile byte pulseCount;

float flowRate ;
float flowLitres ;
unsigned int flowMilliLitres ;
unsigned long totalMilliLitres ;
unsigned long delTime;

unsigned long currentTime ; //เวลาปัจจุบัน
unsigned long previousTime; //เวลาก่อนหน้า



float totalLitres = 0.0;
String dataMessage;
char buffer1[26];
char buffer2[26];

const int SD_CS = D8;  // เสียบช่อง D8 ถ้าจะเปลี่ยนช่องเปลี่ยนตรงนี้ด้วย
RTC_DS3231 rtc;
LiquidCrystal_I2C lcd(0x27,16,2);

//BlynkTimer timer;
SimpleTimer timer;

void setup(){

  Serial.begin(115200);
  Serial.println();
  pinMode(sensorPin, INPUT_PULLUP);
  digitalWrite(sensorPin, HIGH);
```

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

pulseCount        = 0;
flowRate          = 0.0;
flowMilliLitres   = 0;
totalMilliLitres  = 0;
previousTime      = 0;

////////////////////////////////////////////////////////
/* if (! rtc.begin()) {
  Serial.println("Couldn't find RTC");
  Serial.flush();
  abort();
}*/

if (rtc.lostPower()) {
  Serial.println("RTC lost power, let's set the time!");

  // When time needs to be set on a new device, or after a power loss, the
  // following line sets the RTC to the date & time this sketch was compiled
  //rtc.adjust(DateTime(__DATE__,__TIME__));
  // This line sets the RTC with an explicit date & time, for example to set
  // January 21, 2014 at 3am you would call:
 //rtc.adjust(DateTime(2021, 5, 25, 0, 33, 0));
}
lcd.setCursor(0, 0);lcd.print("Initial SD card ");
 // Initialize SD card
 if(!SD.begin(SD_CS)) {
  Serial.println("Card Mount Failed");
  return;
}
File dataFile = SD.open("datalog.txt", FILE_WRITE);
```

```
  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println("Date,Time,flowRate,totalLitres\r\n");
    dataFile.close();
 }

  lcd.setCursor(0, 0);lcd.print("Blynk Connecting");
  Blynk.begin(auth, ssid, pass,"pkdb10.thddns.net",8334);
  delay(4000);
   lcd.clear();
 // Configured to trigger on a FALLING state change (transition from HIGH state to LOW
state)
 attachInterrupt(digitalPinToInterrupt(sensorPin), pulseCounter, FALLING);

  timer.setInterval(1000L, showFlow);
}

void  showFlow() //average the flow over averageperiod
{
     detachInterrupt(digitalPinToInterrupt(sensorPin));
    currentTime = millis();

    delTime = currentTime-previousTime;
    previousTime = currentTime;

    flowRate = ((1000.0 * pulseCount)/delTime) / calibrationFactor ;   // L/min

    flowMilliLitres = (flowRate / 60.0) * 1000.0;                     // mL/s

    flowLitres = (flowRate / 60.0);                                   // L/s

    totalMilliLitres += flowMilliLitres;                              // mL
    totalLitres += flowLitres;                                        // L
```

```
    // Print the flow rate for this second in litres / minute
    Serial.print("Flow rate: ");
    Serial.print(flowRate, 1) ;       // Print the fractional part of the variable
    Serial.print("L/min");
    Serial.print(" Output Liquid Quantity: ");              // Output separator
    Serial.print(totalLitres);
    Serial.println("L");

    Blynk.virtualWrite(V1, flowRate);
    Blynk.virtualWrite(V2, totalLitres);

   ////////////Get date/time////////////
   DateTime now = rtc.now();
   sprintf(buffer1,"%02u-%02u-%04u",now.day(),now.month(),now.year());
   sprintf(buffer2,"%02u:%02u:%02u",now.hour(),now.minute(),now.second());
  // Serial.print(buffer1);Serial.println(buffer2);

dataMessage =  String(buffer1)+ "," + String(buffer2) + "," + String(flowRate) + "," +
String(totalLitres) + "\r\n";
  Serial.print("Save data: ");
  Serial.println(dataMessage);
  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  File dataFile = SD.open("datalog.txt", FILE_WRITE);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataMessage);
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataMessage);
  }
  // if the file isn't open, pop up an error:
  else {
```

```
    Serial.println("error opening datalog.txt");
  }
  //////////////print to LCD////////////
  lcd.setCursor(0, 0);lcd.print("Q=");
  lcd.setCursor(3, 0);lcd.print(flowRate);
  lcd.setCursor(7, 0);lcd.print(" ");
  lcd.setCursor(8, 0);lcd.print("L/min");

  lcd.setCursor(11, 1);lcd.print(now.hour(),DEC);
  lcd.setCursor(13, 1);lcd.print(":");
  lcd.setCursor(14, 1);lcd.print(now.minute(),DEC);

    lcd.setCursor(0, 1);lcd.print("V=");
  lcd.setCursor(3, 1);lcd.print(totalLitres);
  lcd.setCursor(7, 1);lcd.print(" ");
  lcd.setCursor(8, 1);lcd.print("L");

  pulseCount = 0;
  attachInterrupt(digitalPinToInterrupt(sensorPin), pulseCounter, FALLING);
}

/////////////////////////////////////
 ICACHE_RAM_ATTR void pulseCounter(){
  // Increment the pulse counter
  pulseCount++;
}

void loop(){
  Blynk.run();
  timer.run();
}
```

5. แหล่งที่ซื้ออุปกรณ์ และราคา

ชื่ออุปกรณ์	แหล่งที่ซื้อ	ราคา (บาท)
1. DS3231 Module (Real Time Clock)	[ส่งด่วน] DS3231 Module โมดูลนาฬิกาและเซนเซอร์วัดอุณหภูมิ DS3231M สำหรับ Arduino มีเก็บเงินปลายทาง พร้อมส่งทันที !!!!!! | Shopee Thailand	125
2. Micro SD Card Module	โมดูลบันทึกข้อมูลลง Micro SD Card Micro SD Card Module Micro SD Card module | Shopee Thailand	35
3. Water Flow sensor 1/2″ Water flow sensor,Flow meter สำหรับ Arduino	Water Flow Sensor วัดการไหลของน้ำ 1/2" | Shopee Thailand	160
4. LCD--Blue-Screen--16x2-LCD	จอ LCD 1602 +I2C 16x2 1602 Blue screen 16 Character 2 row LCD IIC/I2C Serial Module For Arduino NodeMCU Wemos | Shopee Thailand	100
5. Memory Card 32GB	ร้าน Advice กำแพงแสน	150
6.ข้อต่อตรงเกลียวในทองเหลือง 1/2"ตราช้าง scg	ร้านรุ่งเรืองศรีค้าวัสดุก่อสร้าง	20
7.กล่องกันน้ำ	⚡🏠⚡ ตู้กันน้ำพลาสติก มีฝาเปิดปิด Electric Enclosure Water Proof Box NANO-11W NANO-22W NANO-101W NANO-102W NANO-103W | Shopee Thailand	180
8.เสาเหล็ก6เมตรรวมขาตั้ง	ร้านขายวัสดุก่อสร้าง	400
9. Power Bank	ร้านค้าหน้ามหาวิทยาลัย	
10. อุปกรณ์เบ็ดเตล็ด เทปพันสายไฟ,ปืนกาว,เทปพันเกลียว ตราท่อน้ำไทย,ท่อ pvc class 8.5 ขนาด ½ inch,ข้อต่อต่างๆ,กาวทาท่อน้ำ	ร้านรุ่งเรืองศรีค้าวัสดุก่อสร้าง	260
	รวม	1430

6. ข้อเสนอแนะในการพัฒนาต่อไป

1. ควรเลือกใช้Sensorที่สามารถรับอัตราการไหลให้มากขึ้นกว่าเดิมและมีความแข็งแรงมากขึ้น
2. สามารถเพิ่มSensorวัดอัตราการไหลของแต่ละรุ่นเพื่อตรวจสอบและเปรียบเทียบความแม่นยำของอัตราการไหล
3. สามรถนำไปประยุกต์ใช้กับเครื่องมือการให้น้ำแบบอัตโนมัติในการเกษตร
4. สามรถนำไปประยุกต์ใช้ในท่อที่มีขนาดต่างกันเพื่อที่จะดูอัตราการไหลของแต่ละขนาด

บรรณานุกรม

1. My Arduino.(2563). สอนใช้งาน NodeMCU ESP8266 DS3231 Module โมดูลนาฬิกา ตั้งเวลาใหม่, สืบค้นเมื่อ 10 พฤศภาคม 2564.จาก.แหล่งที่มา: https://www.myarduino.net/article/183/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-nodemcu-esp8266-ds3231-module

2. My Arduino.(2563). สอนใช้งาน สอนใช้งาน NodeMCU ESP8266 บันทึกข้อมูลลง SD Card และ อ่านข้อมูลจาก SD card,สืบค้นเมื่อ 10 พฤศภาคม 2564.จาก.แหล่งที่มา: https://www.myarduino.net/article/180/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-nodemcu-esp8266

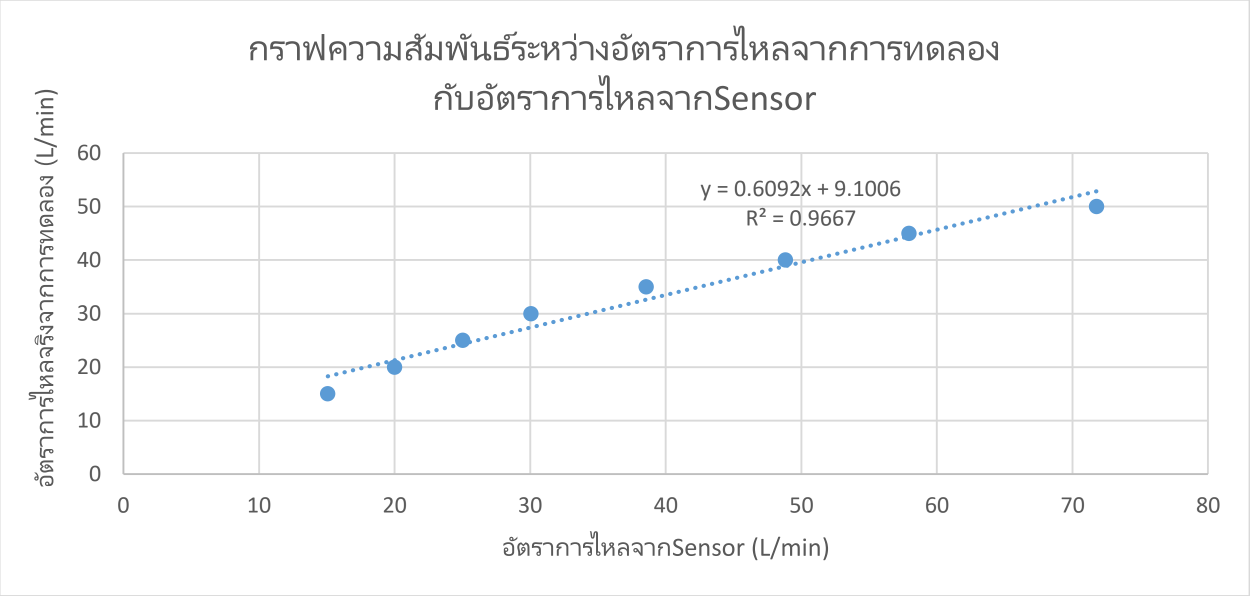
3. AB-Maker.(19 พ.ค. 2562). การใช้งาน Code Water flow Sensor กับ Arduino วัดน้ำไหลในท่อ, สืบค้นเมื่อ 10 พฤศภาคม 2564.จาก.แหล่งที่มา: https://www.ab.in.th/article/8/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-code-water-flow-sensor

ภาคผนวก ก. ผลการเปรียบเทียบ

ผลการเปรียบเทียบเครื่องมือวัดอัตราการไหลแบบพกพากับอุปกรณ์วัดอัตราการไหลภายในท่อ ณ ห้องปฏิบัติการกลศาสตร์ของของไหล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน

ตารางแสดงผลการทดลอง

อัตราการไหลจริงจากการทดลอง (L/min)	อัตราการไหลจากSensor (L/min)				เวลา (นาที)	ปริมาตรจาก Sensor (L.)				ค่าความคลาดเคลื่อนของอัตราไหล	ค่าความคลาดเคลื่อนของอัตราไหล (%)
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย		ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย		
15	15.1	15.1	15	15	1	14	15	15	15	0.00	0
20	20	20	20	20	1	20	19	20	20	0.00	0
25	25	25	25.1	25	1	25	25	25	25	0.00	0
30	30.2	30	30	30	1	30	31	30	30	0.00	0
35	38.4	38.8	38.5	39	1	38	39	38	38	0.10	10
40	48.7	48.8	49	49	1	48	49	49	49	0.22	22
45	58	57.8	58	58	1	58	59	59	59	0.29	29
50	72.7	71	71.6	72	1	72.7	71	71	72	0.44	44



กราฟที่ 1 แสดงการเปรียบเทียบระหว่างค่าอัตราการไหลจากการทดลองกับค่าอัตราการไหลจากSensor