

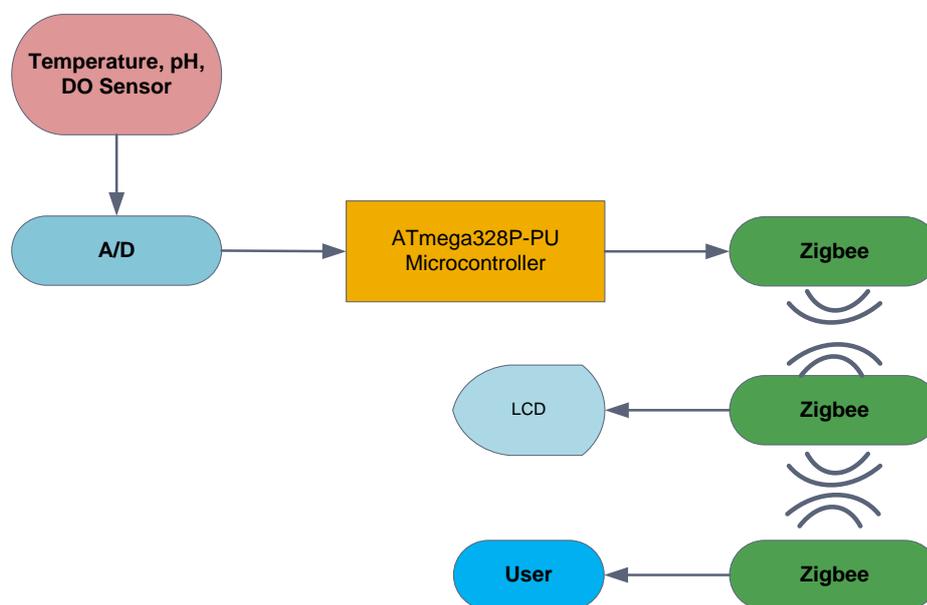
## บทที่ 4

### ผลการศึกษา

ผู้วิจัยได้ดำเนินการเพื่อวางกรอบและแนวทางการวิจัยตามแผนงาน เพื่อหาข้อมูล เอกสาร มาตรฐานและงานวิจัยที่เกี่ยวข้องเกี่ยวกับการศึกษาและพัฒนาระบบตรวจวัดคุณภาพของแหล่งน้ำ โดยการใช้เทคโนโลยีซิกบี. เพื่อรวบรวม วิเคราะห์ในการกำหนดกรอบในการพัฒนาระบบตรวจวัดคุณภาพของแหล่งน้ำ ซึ่งผลจากการศึกษาข้อมูล คณะผู้วิจัยจึงได้สร้างเครื่องวัดคุณภาพแหล่งน้ำ และผลการทดสอบการใช้เครื่องมือ ดังนี้

#### 1. การออกแบบระบบโดยรวม

ในขั้นตอนการพัฒนาได้ทำการออกแบบภาพรวมทั้งหมดประกอบไปด้วยอุปกรณ์เครือข่ายไร้สาย 2 ตัว sensor 3 ตัว และคอมพิวเตอร์จำนวน 1 เครื่อง ซึ่ง อุปกรณ์เครือข่ายไร้สาย 1 ตัว จะทำหน้าที่ในการเชื่อมต่อเข้ากับคอมพิวเตอร์ ส่วนตัวรับสัญญาณมีอุปกรณ์เครือข่ายไร้สายอีก 1 ตัว ติดอยู่กับ sensor เพื่อเป็นตัวส่งสัญญาณจากการอ่านค่าที่ sensor วัดได้ไปยังโหนดอ้างอิงที่ทำการเชื่อมต่อกับคอมพิวเตอร์เพื่อแสดงผลของตำแหน่ง ดังรูปที่ 4.1



รูปที่ 4.1 การออกแบบระบบ

#### 2. การติดตั้งชุดอุปกรณ์ไมโครคอนโทรลเลอร์ และอุปกรณ์เซ็นเซอร์ต่างๆ

จากรูป 4.2-4.3 เมื่อทำการออกแบบระบบเป็นที่เรียบร้อยแล้วก็ดำเนินการติดตั้งและเชื่อมต่ออุปกรณ์ต่างๆเพื่อทำการทดสอบระบบให้ตรงตามมาตรฐานการวัดของแต่ละเซ็นเซอร์



รูปที่ 4.2 การเตรียมกล่องติดตั้งอุปกรณ์และติดตั้งชุดอุปกรณ์คอนโทรลเลอร์

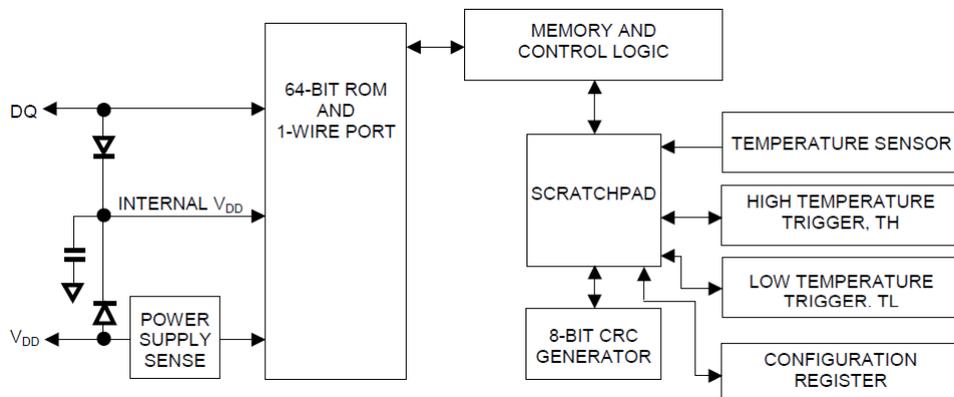


รูปที่ 4.3 ชุดอุปกรณ์เซ็นเซอร์ที่ติดต่อกับคอนโทรลเลอร์ชิป

### 3. การออกแบบด้านฮาร์ดแวร์

#### 3.1 เซ็นเซอร์วัดอุณหภูมิ

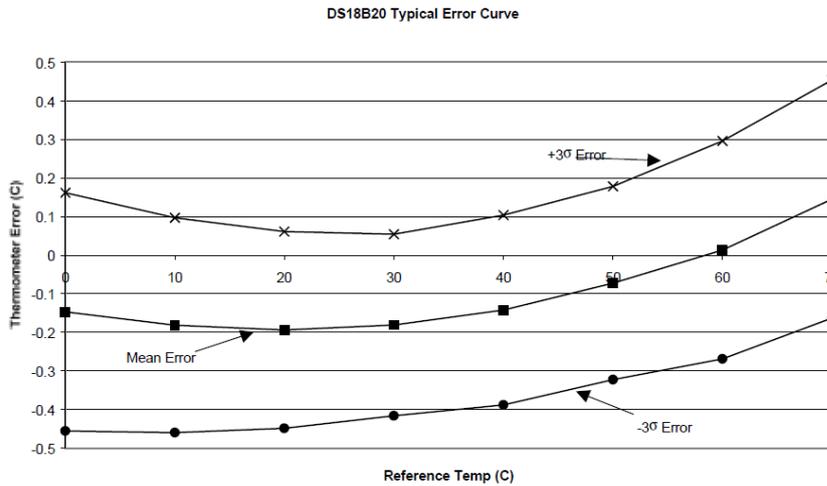
ได้ทำการออกแบบวงจรให้ไมโครคอนโทรลเลอร์รับค่าที่เป็นอนาล็อกจากเซ็นเซอร์อุณหภูมิผ่านทาง Analog Pin1 โดยในส่วนของ sensor อุณหภูมิได้เลือกใช้เซ็นเซอร์ประเภทตรวจจับอุณหภูมิ Temperature Sensor - Waterproof (DS18B20) เป็นหัววัดอุณหภูมิโดยใช้ Chip DS18B20 ที่ทำหัวเซ็นเซอร์ให้กันน้ำด้วย มีช่วงการวัดอยู่ที่  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ใช้งานติดต่อผ่านทาง 1 Wire interface โดยวงจรของอุปกรณ์เซ็นเซอร์วัดอุณหภูมิได้ถูกออกแบบตามรูปที่ 4.4



รูปที่ 4.4 วงจรเซ็นเซอร์วัดค่าอุณหภูมิ



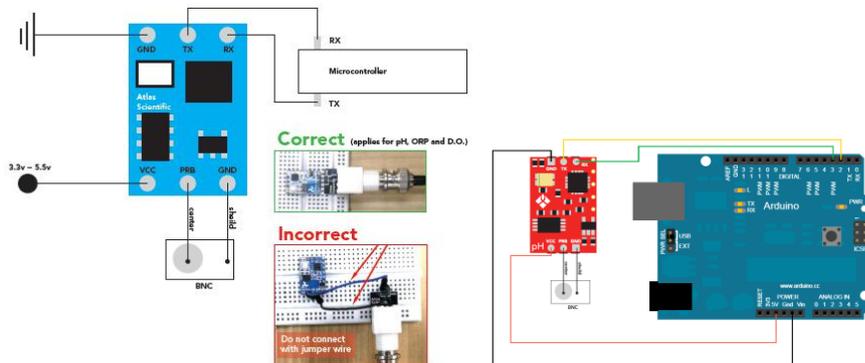
รูปที่ 4.5 การเชื่อมต่ออุปกรณ์เซ็นเซอร์ค่าอุณหภูมิ



รูปที่ 4.6 กราฟแสดงค่าความผิดพลาดของเซ็นเซอร์วัดอุณหภูมิ

### 3.2 เซ็นเซอร์วัดค่า PH

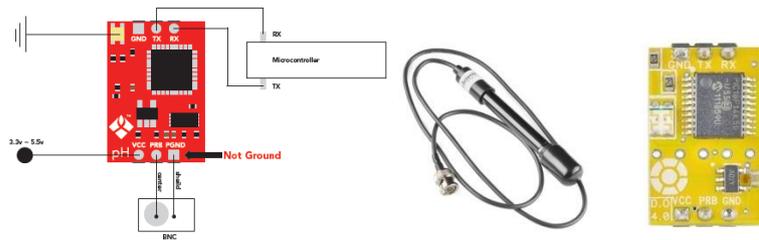
ได้ทำการออกแบบวงจรให้ไมโครคอนโทรลเลอร์รับค่าที่เป็นอนาล็อกจากเซ็นเซอร์วัดค่าความเป็นกรด เป็นด่าง ดังรูปที่ 4.7 โดยในส่วนของเซ็นเซอร์วัดค่า pH ได้เลือกใช้เซ็นเซอร์ประเภทที่ทำหน้าที่ตรวจจับสัญญาณ Analog ที่วัดได้จาก Sensor แล้วประมวลผลส่งข้อมูลทาง Serial Port (UART) เพียงส่งคำสั่งไปที่ ไมโคร Controller เพื่อดึงข้อมูลตามต้องการ โดยสามารถวัดค่า pH ได้ตั้งแต่ 0-14 offset +/- 0.20 pH อัตราการส่งข้อมูล 38400 bps ขนาด 8 บิต no parity



รูปที่ 4.7 การเชื่อมต่ออุปกรณ์เซ็นเซอร์ค่า PH

### 3.3 เซ็นเซอร์วัดค่า Do Oxygen

ได้ทำการออกแบบวงจรให้ไมโครคอนโทรลเลอร์รับค่าที่เป็นอนาล็อกจากเซ็นเซอร์ตรวจจับความเข้มข้นของออกซิเจนในน้ำ ดังรูปที่ 4.8 โดยในส่วนของเซ็นเซอร์วัดค่า DO ได้เลือกใช้เซ็นเซอร์ประเภทเป็นชุด Sensor ตรวจจับความเข้มข้นของออกซิเจน (Oxygen) ในน้ำว่ามีปริมาณมากหรือน้อย โดย Sensor จะตรวจวัดค่าแล้วมีตัวประมวลผล สามารถเก็บค่าที่วัดได้ไปประมวลผล โดย Interface กับโมดูลผ่านทาง Serial Port (UART) เพียงส่งคำสั่งไปที่ โมดูล ก็จะได้ Data ตามต้องการ โดยสามารถวัดค่าได้ระหว่าง 0-20 mg/L PSL สูงสุดที่ 690 kPa



รูปที่ 4.8 ทดสอบอุปกรณ์เซ็นเซอร์ค่า DO Oxygen

#### 4. ติดตั้งโปรแกรมติดต่อคอนโทรลเลอร์และอุปกรณ์ต่างๆด้วยโปรแกรม Arduino

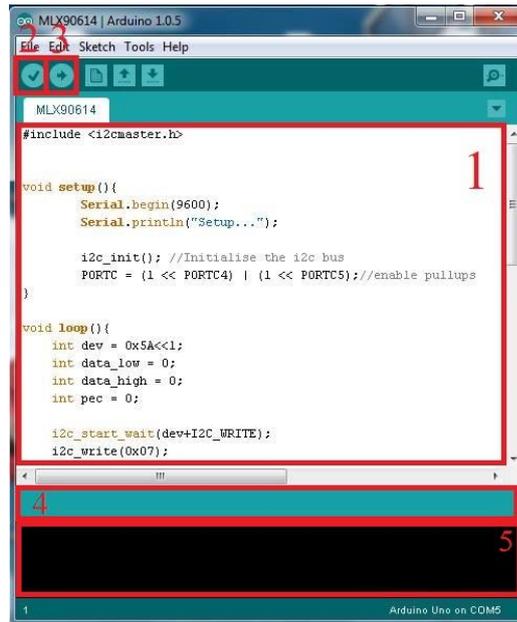
ดังรูปที่ 4.5 ใช้สำหรับการเขียนโปรแกรมควบคุม ไมโครคอนโทรลเลอร์ ซึ่งภาษาของ Arduino จะอ้างอิงตามภาษา C/C++ จึงอาจกล่าวได้ว่าการเขียนโปรแกรมสำหรับ Arduino ก็คือการเขียนภาษาซี โดยเรียกใช้ฟังก์ชันและไลบรารีที่ทาง Arduino ได้เตรียมไว้ให้แล้ว

โปรแกรมของ Arduino แบ่งได้เป็นสองส่วนคือ

Void setup()

Void loop()

โดยฟังก์ชัน setup() เมื่อโปรแกรมทำงานจะทำคำสั่งของฟังก์ชันนี้เพียงครั้งเดียว ใช้ในการกำหนดค่าเริ่มต้นของการทำงาน ส่วนฟังก์ชัน loop() เป็นส่วนทำงาน โปรแกรมจะทำคำสั่งในฟังก์ชันนี้ต่อเนื่องกันตลอดเวลา โดยปกติใช้กำหนดโหมดการทำงานของขาต่าง ๆ กำหนดการสื่อสารแบบอนุกรม ฯลฯ ส่วนของ loop() เป็นโค้ดโปรแกรมที่ทำงาน เช่น อ่านอินพุต ประมวลผล สั่งงานเอาต์พุต ฯลฯ โดยส่วนกำหนดค่าเริ่มต้น เช่น ตัวแปร จะต้องเขียนที่ส่วนหัวของโปรแกรม ก่อนถึงตัวฟังก์ชัน นอกจากนั้นยังต้องคำนึงถึงตัวพิมพ์เล็ก - พิมพ์ใหญ่ ของตัวแปรและชื่อฟังก์ชันให้ถูกต้อง



รูปที่ 4.6 หน้าต่างแรกของโปรแกรม Arduino

#### 4.1 การเขียนโปรแกรมติดต่อเซ็นเซอร์วัดค่าความเป็นกรด เป็นด่าง และเซ็นเซอร์วัดค่าอุณหภูมิ

```
#include <SoftwareSerial.h> //we have to include the SoftwareSerial library,
                             //or else we can't use it.
```

```
#include <LiquidCrystal.h>
```

```
#define rx 2 //define what pin rx is going to be.
```

```
#define tx 3 //define what pin Tx is going to be.
```

```
#define SERIESRESISTOR 10000
```

```
#define THERMISTORPIN A1 //define A1 to analog input Tempature to board
```

```
SoftwareSerial myserial(rx, tx); //define how the soft serial port is going to work.
```

(send signal to board of (2,3) รับ-ส่ง ค่าจากเซ็นเซอร์ ไปยังบอร์ดที่ 2,3

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

```
SoftwareSerial mySerZig(0, 1); //Board Arduino (0,1) (rx,tx) send to board Zigbee
```

บอร์ดอาร์ดูยโน้ ฮา 0,1 ไปยังบอร์ด ชิก็ปี

```
//String rrr;
```

```
char ph_data[20];
```

```
char computerdata[20];
```

```

byte received_from_computer=0;
byte received_from_sensor=0;
byte arduino_only=0;
byte startup=0;
float ph=0;
byte string_received=0;

void setup(){
  Serial.begin(38400);    //enable the hardware serial port 38400
  myserial.begin(38400); //enable the software serial port
  lcd.begin(16, 2);      // start the library
  //lcd.setCursor(0,0);
  //lcd.print("Push the buttons"); // print a simple message
}

void serialEvent(){
  if(arduino_only!=1){
    received_from_computer=Serial.readBytesUntil(13,computerdata,2
0);
    computerdata[received_from_computer]=0;
    myserial.print(computerdata);
    myserial.print('\r');
  }
}

void loop(){
  //Temperature
  float reading;
  float reading01;

  reading = analogRead(THERMISTORPIN);

  //Serial.print("Analog reading ");
  //Serial.println(reading);

  // convert the value to resistance
  reading = (1023 / reading) - 1;

```

```

reading = SERIESRESISTOR / reading;
//Serial.print("Thermistor resistance ");
// Serial.println(reading);

//convert to c
reading01 = (reading/10000) * 35;
//Serial.print("Celsius reading ");
//Serial.println(reading01);

//End Temperature

if(myserial.available() > 0){
  received_from_sensor=myserial.readBytesUntil(13,ph_data,20);
  ph_data[received_from_sensor]=0;
  string_received=1;
  Serial.println(ph_data);
  //myserial.print(ph_data);
  //Serial.println("7"); //for testing
  Serial.println(reading01);
}

if(arduino_only==1){Arduino_Control();}
  mySerZig.print(ph_data); //arduino send data to Zigbee of Temperature
  อาร์ทดุยน โส่งข้อมูล ไปที่ ชิก็ปี
  //mySerZig.print("7"); //for testing
  mySerZig.print(reading01);
  //mySerZig.print(rrr);
  lcd.setCursor(0,0); // move cursor to second line "1" and 9 spaces over
  //lcd.print("7");
  lcd.print("pH is "); // display seconds elapsed since power-up
  lcd.print(ph_data);
  lcd.setCursor(0,1);
  lcd.print("Temp is ");
  lcd.print(reading01);
  //delay(1000);
}

```

```

void Arduino_Control(){

    if(startup==0){
        myserial.print("e\r");    //take the pH Circuit out of continues mode.
        delay(50);                //on start up sometimes the first command is
missed.
        myserial.print("e\r");    //so, let's send it twice.
        delay(50);
        startup=1;
    }
    delay(800);
    myserial.print("R\r");        //send it the command to take a single reading.
    if(string_received==1){      //did we get data back from the pH Circuit?
        ph=atof(ph_data);
        if(ph>=7.5){Serial.println("high\r");}
        if(ph<7.5){Serial.println("low\r");}
        string_received=0;      //reset the string received flag.
    }

//here are some functions you might find useful
//these functions are not enabled

void s_cal(){                  //calibrate to a pH of 7
    myserial.print("s\r");    //send the "s" command to calibrate to a pH of 7.00

void f_cal(){                  //calibrate to a pH of 4
    myserial.print("f\r");    //send the "f" command to calibrate to a pH of 4.00

void t_cal(){                  //calibrate to a pH of 10.00
    myserial.print("t\r");    //send the "t" command to calibrate to a pH of 10.00

void pHFactoryDefault(){      //factory defaults the pH circuit
    myserial.print("X\r");    //send the "X" command to factory reset the device

void read_info(){             //get device info
    myserial.print("I\r");    //send the "I" command to query the information

```

```

void phSetLEDs(byte enabled) //turn the LEDs on or off
{
  if(enabled)           //if enabled is > 0
    myserial.print("L1\r"); //the LED's will turn ON
  else                   //if enabled is 0
    myserial.print("L0\r"); //the LED's will turn OFF
}

```

#### 4.2 การเขียนโปรแกรมติดต่อเซ็นเซอร์วัดค่าความเข้มข้นของออกซิเจนในน้ำ

```

#include <SoftwareSerial.h>           //add the soft
serial libray
#define rxpin 12                       //set the RX pin to
pin 2
#define txpin 13                       //set the TX pin to
pin 3

SoftwareSerial myserial(rxpin, txpin); //enable the soft serial port

String inputstring = "";
String sensorstring = "";
boolean input_stringcomplete = false;
boolean sensor_stringcomplete = false;

void setup(){                          //set up the hardware
  Serial.begin(38400);                  //set baud rate for
the hardware serial port to 38400
  myserial.begin(38400);                //set baud rate for
software serial port to 38400
  inputstring.reserve(5);               //set aside some
bytes for receiving data from the PC
  sensorstring.reserve(30);            //set aside some
bytes for receiving data from Atlas Scientific product
}

void serialEvent() {
  char inchar = (char)Serial.read();

```

```

        inputstring += inchar;                                //add it to the
inputString
        if(inchar == '\r') {input_stringcomplete = true;}
    }

void loop(){                                                //here we go...

    if (input_stringcomplete){
        myserial.print(inputstring);
        inputstring = "";                                    //clear the string:
        input_stringcomplete = false;
    }

    while (myserial.available()) {
        char inchar = (char)myserial.read();                //get the new
char
        sensorstring += inchar;
        if (inchar == '\r') {sensor_stringcomplete = true;}
    }

    if (sensor_stringcomplete){
        Serial.print(sensorstring);
        sensorstring = "";                                    //clear the string:
        sensor_stringcomplete = false;
    }
}

```

