Farm Automation System (Light Bulb Control) with Temperature and Humidity Detection System with IOT Data Logging Function

BCA144 – B183
Introduction to IoT

Neil Christian P. Balayon
Mark Joceph A. Apao
Jyresa Mae M. Amboang
Matthew Karl M. Asidillo

May 2022

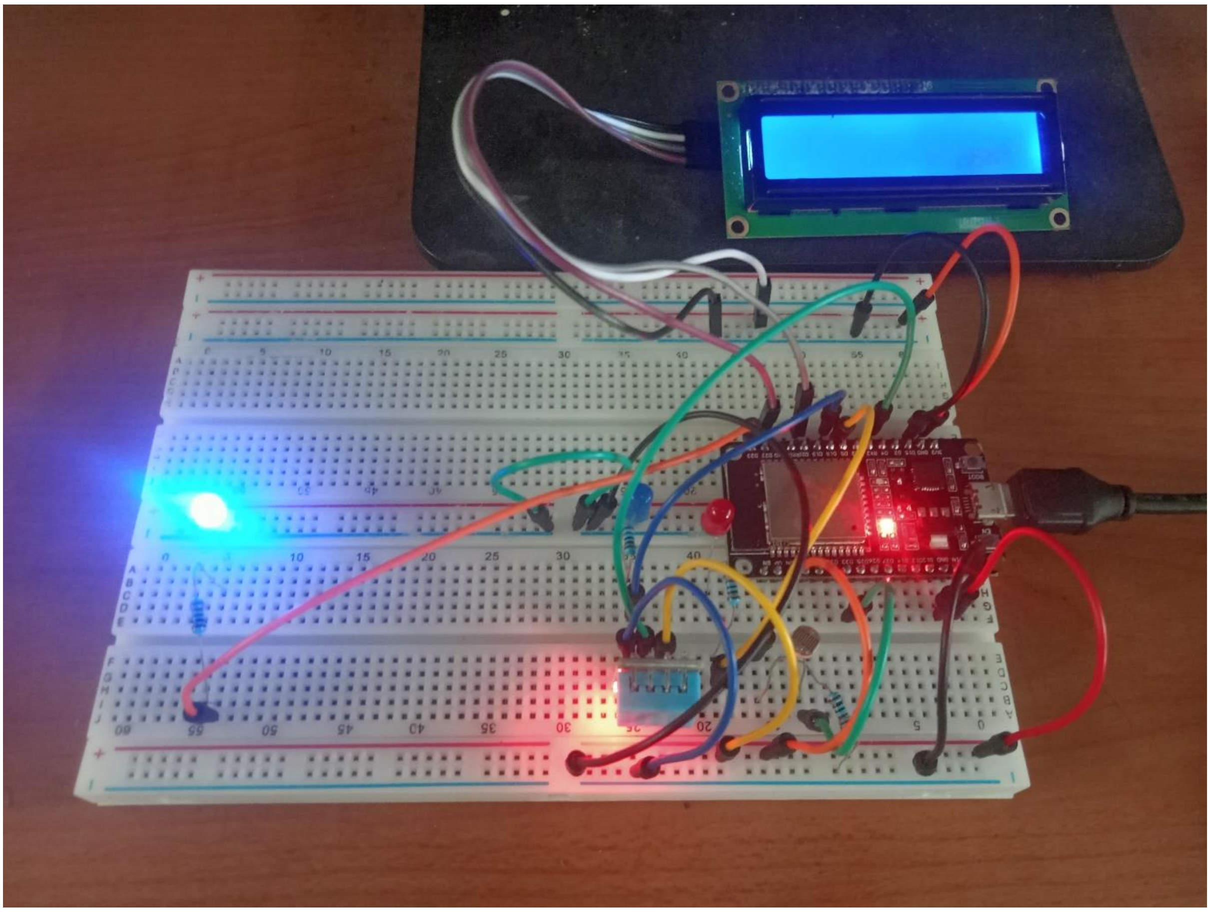


Table of Contents

CHAPTER I

Introduction

Agriculture faces new major problems. Our Farm Automation system may be regarded as a goal-setting exercise with the primary purpose of increasing agricultural and farm resource output.

This project focuses on the creation of a computational system for farm applications. A prototype of a farm automation system was constructed for this purpose, and embedded system implementation was produced utilizing sensors capable of suitable air condition for plant, animals, and other circumstances in a farm.

We are using a cloud service of Adafruit IO from this monitored environment, and records of the gathered data are made available in real-time with computer access over the Internet. As a result, the goal of this research is to describe the construction of a system model based on IoT communication that allows access to temperature, humidity, and luminosity records.

1. Scope and Significance of the Study

This project emphasizes that agriculture must connect with new technologies in order to meet these expanding demands. Agriculture will benefit from new smart farming applications powered by IoT. This will help farmers to overcome the immense challenges they confront. To apply automation to the farm environment process, our prototype was developed which has light management, temperature sensor, and humidity sensor. These sensors are used to monitor and manage the factors that have the greatest impact on the growth of a vegetal or animal species' air condition and also it can provide farmers with a wealth of information and data on current patterns and innovation.

2. Theoretical Framework

Figure 1.1 presents the different blocks on how data flows in our circuitry.

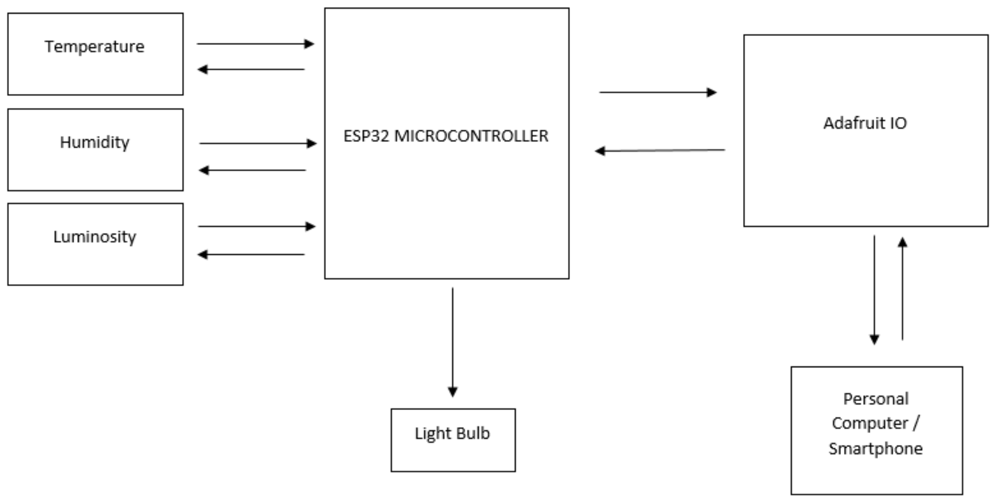


Figure 1.1 Theoretical Framework of Farm Automation System

CHAPTER II

METHODOLOGY

This chapter discusses the different components that the researchers have used in their study and the methodology we employed.

1. Sensors

We use DHT11 which detects the temperature and humidity. Also, we use a photoresistor which detects the luminosity. We consider sensor operating voltage, accuracy, operating current, and measurement range. The table 1 describe the sensors features that was chosen for this prototype.

Table 1. Characteristics of the sensors.

Variable	**Sensor**	**Operating Voltage**	**Accuracy**	**Operating Current**	**Measurement range**
Temperature	**DHT11**	**3.3 V to 6 V**	**+/- 0.2º C**	**1 - 1.5 mA**	**-40ºC to 125ºC**
Humidity	**DHT11**	**3.3 V to 6 V**	**+/- 2 %**	**1 - 1.5 mA**	**0 to 100 %**
Luminosity	**Photoresistor**	**0 V to 1.8 V**			

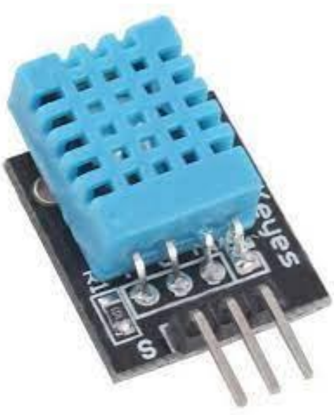


Fig.2 DHT11

Fig.3 Photoresistor

These sensors are connected to the ESP32 microcontroller and will provide the data needed for the automation. The data collected by the sensor will be uploaded to the Adafruit IO platform through the microcontroller which will read, display, and analyze the real-time readings of the sensors.

2. ESP32 Microcontroller

The ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces. Figure 4 shows how ESP32 upload data readings to Adafruit IO through Wi-Fi.

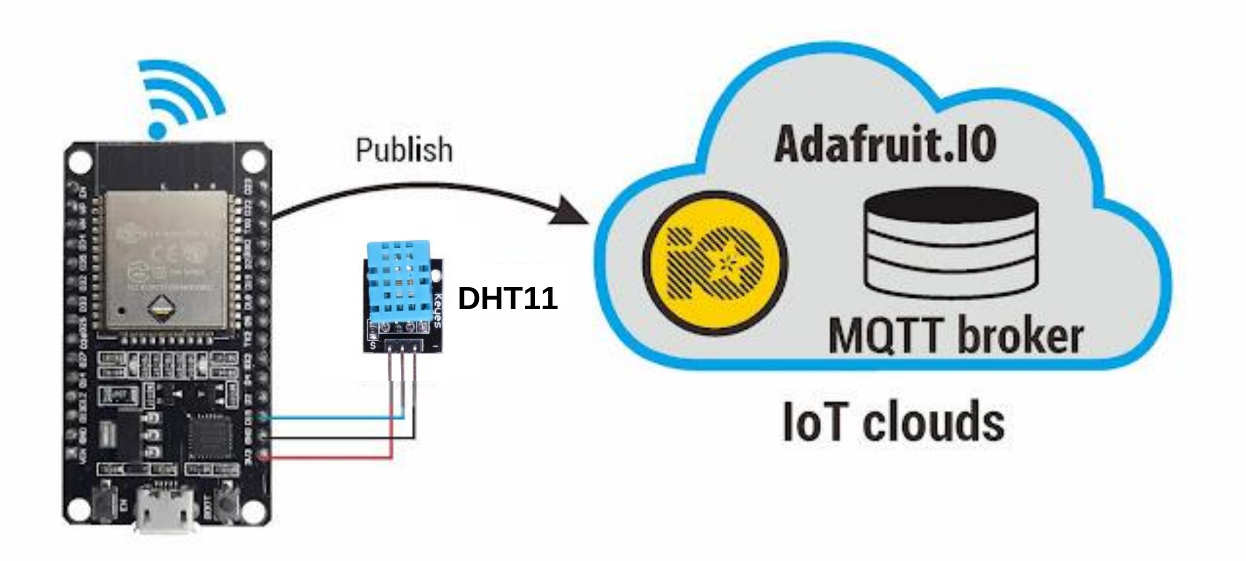


Fig.4 How ESP32 upload data readings to Adafruit IO

3. Adafruit IO

Adafruit.io is a cloud service that just means they run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data. As shown in figure 4 how data is uploaded from the microcontroller to the Adafruit platform. In figure 5 is the dashboard of the actual data readings from the sensors transmitted to the ESP32 then was uploaded to Adafruit.

Fig.5 Adafruit Dashboard

4. Personal Computer/Smartphone

The computer or smartphone device that is connected to the Adafruit account where the microcontroller is linked will received the real-time readings which will display the data as shown in figure 5. The information received can be publicize by letting other accounts to view the readings. In this way there is no need to log in the account where the microcontroller for the farm automation system is connected.

5. LED (Assumed Light Bulb)

The LED is the output that will manifest the desired output depends of the data readings and with the specific range that is set for bulb light auto activation. This process will repeat infinitely.

6. Light Indicators

Using 2 light LEDs as indicator for the person who is present in the farm to know if the environment is either warm or cold. The LEDs are red and blue where red will indicate if the environment inside the farm is warm and the blue when it's cold. The LED indicator outputs depends on what the reading is receive from the sensors and with the specific range this output indicators will give LOW or HIGH output as shown in figure 6.

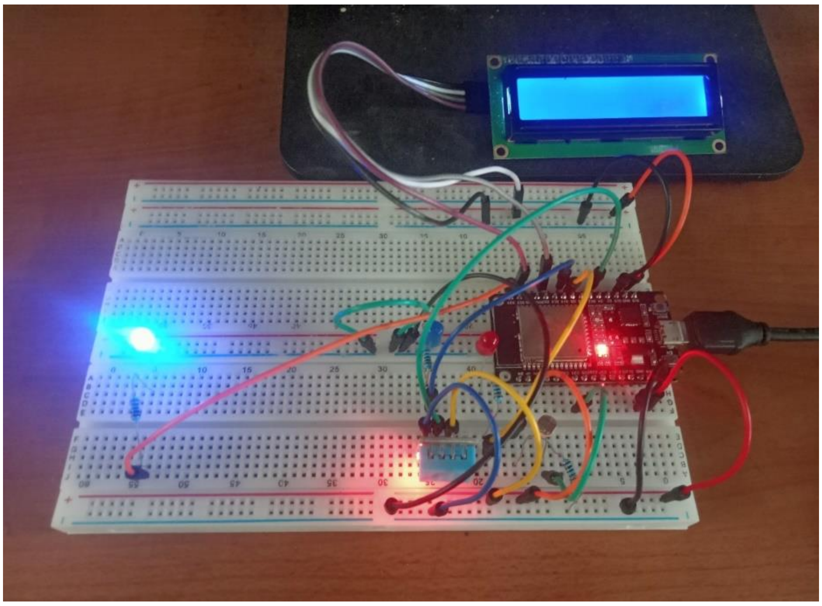


Fig.6 Actual Device

CHAPTER III

CONCLUSION AND RECOMMENDATION

1. Conclusion

Our Farm Automation system can be thought of as a goal-setting activity aimed at increasing agricultural and farm resource output. Also, this system is also ideal for remote weather stations and home environmental management systems.

2. Recommendation

Based on the findings and conclusions, the researchers presented the following recommendations and suggestions to:

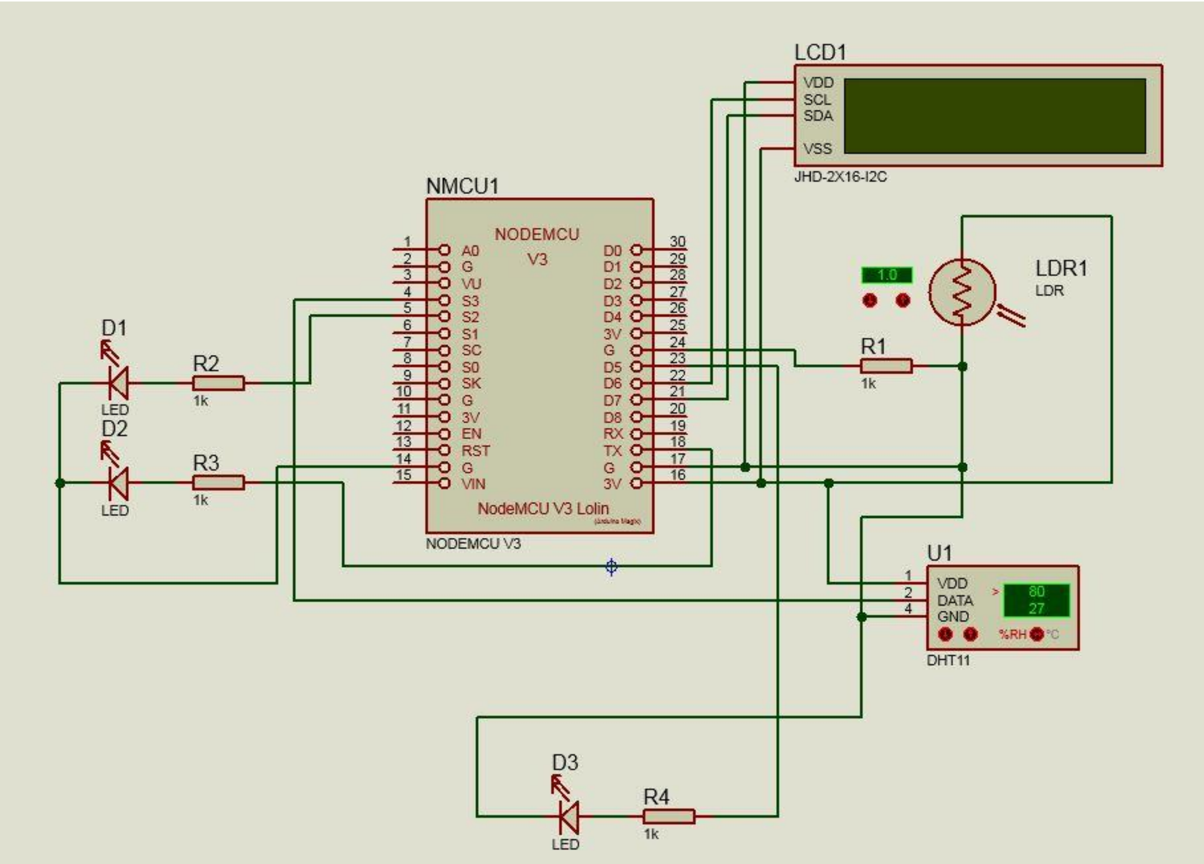
- Must add a function which automatically or manually sprinkles water into a certain area.
- Must add a fire alarm triggering system in case of fire.
- Automatic feeding system for farm animals.
- Improved dashboard and notification system.
- Improved performance.

Google drive folder:
https://drive.google.com/drive/folders/1lBsh3a9hKc4eurDUWx0D8RvqQYSCjCw

APPENDICES

Appendix A

Schematic Diagram



Appendix B

Arduino Code

```
#define IO_USERNAME "ncbalayon"
#define IO_KEY "aio_KcSm62U1LqygqArtTpP54udksHVD"
#define WIFI_SSID "Balayon_Family"
#define WIFI_PASS "Balayon12300.."

#include "AdafruitIO_WiFi.h"

#if defined(USE_AIRLIFT) || defined(ADAFRUIT_METRO_M4_AIRLIFT_LITE) ||
\
    defined(ADAFRUIT_PYPORTAL)

#if !defined(SPIWIFI_SS)

#define SPIWIFI SPI
#define SPIWIFI_SS 10
#define NINA_ACK 9
#define NINA_RESETN 6
#define NINA_GPIO0 -1
#endif
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS,
SPIWIFI_SS,
                   NINA_ACK, NINA_RESETN, NINA_GPIO0, &SPIWIFI);
#else
AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);
#endif

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

// pin connected to DH22 data line
#define DATA_PIN 4

// create DHT22 instance
DHT_Unified dht(DATA_PIN, DHT11);

AdafruitIO_Feed *temperature = io.feed("temperature");
AdafruitIO_Feed *humidity = io.feed("humidity");
AdafruitIO_Feed *luminance = io.feed("luminance");
AdafruitIO_Feed *led = io.feed("led");
AdafruitIO_Feed *automatic = io.feed("automatic");

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

#define redLED 5
#define blueLED 18
#define bulb 23

#define PRPIN 24
```

```
int x, manual;

//////////////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////

void tests(){
   // connect to io.adafruit.com
  lcd.clear();
  lcd.print("Connecting");
  io.connect();

  led->onMessage(handleMessage);
  automatic->onMessage(handleMessage2);

  //Initialize Light Bulb


  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    lcd.print(".");
         int luxx = analogRead(PRPIN);

         if(luxx > 100){
            digitalWrite(bulb, LOW);
          }else{
            digitalWrite(bulb, HIGH);
          }
    delay(500);
  }

  // we are connected
  lcd.setCursor(0, 1);
  lcd.print("CONNECTED!");

  led->get();
  automatic->get();

  delay(2000);

  lcd.clear();
}

void setup() {

  //delay(5000);

  dht.begin();

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

  pinMode(redLED, OUTPUT);
  pinMode(bulb, OUTPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(PRPIN, INPUT);

  tests();
```

```
}

void loop() {

   if(io.status() != AIO_CONNECTED){

      lcd.clear();
      lcd.print("DISCONNECTED!");
      delay(3000);

      tests();

    }

  io.run();

  sensors_event_t event;
  dht.temperature().getEvent(&event);

  int celsius = event.temperature;

  temperature->save(celsius);

  dht.humidity().getEvent(&event);

  int humid = event.relative_humidity;

  humidity->save(event.relative_humidity);

  int lux = analogRead(PRPIN);

  lcd.setCursor(0, 0);

  lcd.print("T: ");
  lcd.print(celsius);
  lcd.print("*");
  lcd.print("C");
  lcd.print("  ");
  lcd.print("H: ");
  lcd.print(humid);
  lcd.print("%");

  lcd.setCursor(0, 1);
  lcd.print("Bulb Stat: ");

  if(celsius > 35 && humid < 35){
    digitalWrite(redLED, HIGH);
    //digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, LOW);
  }else if(celsius < 30 && humid > 60){
    digitalWrite(redLED, LOW);
   // digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, HIGH);
  }else if(celsius >= 30 && celsius <= 35 && humid >= 35 && humid <=
60){
    digitalWrite(redLED, LOW);
   // digitalWrite(greenLED, HIGH);
```

```
    digitalWrite(blueLED, LOW);
  }else{
    digitalWrite(redLED, LOW);
    //digitalWrite(greenLED, HIGH);
    digitalWrite(blueLED, LOW);
  }

  lcd.setCursor(12, 1);

  if(manual == 1){
    if(lux < 100 && x == 1){
       digitalWrite(bulb, HIGH);
       lcd.print("ON ");
    }else if(lux > 100 && x == 1){
       digitalWrite(bulb, HIGH);
       lcd.print("ON ");
    }else if(lux < 100 && x == 0){
       digitalWrite(bulb, LOW);
       lcd.print("OFF");
    }else if(lux > 100 && x == 0){
       digitalWrite(bulb, LOW);
       lcd.print("OFF");
    }
  }else if(manual == 0){
    if(lux > 100){
      digitalWrite(bulb, LOW);
      lcd.print("OFF");
    }else{
      digitalWrite(bulb, HIGH);
      lcd.print("ON ");
    }
  }
  /*int PRVALUE = analogRead(PRPIN);

  if(PRVALUE <= 100){
    digitalWrite(bulb, HIGH);
  }else{
    digitalWrite(bulb, LOW);
  }*/

  luminance->save(lux);

  delay(10000);

}

//------------------------------------------------------------

void handleMessage(AdafruitIO_Data *data) {
  //Serial.print("received <- ");

  if(data->toPinLevel() == HIGH)
    x = 1;
  else
    x = 0;

  //digitalWrite(bulb, data->toPinLevel());
}
```

```
void handleMessage2(AdafruitIO_Data *data) {
  //Serial.print("received <- ");

  if(data->toPinLevel() == HIGH)
    manual = 1;
  else
    manual = 0;

  //digitalWrite(bulb, data->toPinLevel());
}
```