AN-NAJAH NATIONAL UNIVERSITY

Computer Engineering Department

Hardware Graduation Project

Livesmart

Students
Dania Nazzal and Marah Dmaide

Under the supervision of
Dr.Suleiman Abu Kharmeh

A report submitted in partial fulfillment of the requirements for
bachelor's degree in computer engineering in the Faculty of Engineering
Information Technology - Hardware Project

May 2023

Acknowledgements

We would like to express our gratitude to everyone who contributed to this initiative. It would not have been possible if it weren't for their assistance and support. Many thanks to Dr. Sulieman Abu Kharmeh for his leadership, tenacious oversight, and assistance in seeing the project through to completion. Our families' support during this trying and stressful period, our coworkers' moral and technical support, and the people who volunteered their time are all greatly appreciated.

Disclaimer

The authors of this research are Dania Nazzal and Marah Dmaide from the Faculty of Engineering's Department of Computer Engineering at An-Najah National University. Other than editorial corrections, there have been no alterations made to the document to enhance its language or content. The thoughts expressed and any recommendations made in it are not supported by Marah Dmaide or Dania Nazzal. When a report is used for a purpose other than the one for which it was commissioned, An-Najah National University disclaims all liability and responsibility for the results.

Contents

List of Figures

Abstract

The world is witnessing a dramatic acceleration in the development and adoption of new technologies, the internet is also changing. People used to use the internet to connect to each other. But with the changes in recent years, the internet is starting to be used more to connect devices to each other (IoT).

The Internet of Things (IoT) is one of the foremost buzzwords in computer science and information technology. It describes the fact that the Internet will be used more and more to connect devices to each other. In today's connected world, the primary purpose of technology is to make our lives more comfortable, giving us more time for leisure activities. And this is what we did in our project, which is a weather station, so that the technology was harnessed up for our service and for a more comfortable life.

As we mentioned earlier, the project is weather station that monitors weather fluctuations by sensing the external environment. Based on the values read, some events will be performed, for example turning off or on the irrigation system based on the soil moisture level. It is also possible that this station will help us to live smart.

Also, there is an uninterruptible power source that provides emergency power when the input power source fails. Also, we added a renewable energy source which is a solar technology the converts sunlight into electrical energy, and this is one of the most important steps in order to preserve the environment from pollution.

There is actually an application for this idea already, but it still needs many improvements in order to reach the required level of efficiency.

1 Introduction

If we consider that technology should only be adopted to solve a problem, we will missed many opportunities that can facilitate and make our lives more convenient and smarter. Even at a business level, technology helps create opportunities for its growth.

Recently, smart home technology has become more and more widespread. There are many reasons that contributed to this, Where we can control home appliances and systems from anywhere, or even according to changes in the surrounding environment, energy efficiency, provide home management insights so we can monitor our activities and behaviors at home, and it can save up to 30-40% energy usage compared to non-smart homes as studies show[1].
Our main objective with this project is to control the house and its garden are by environmental changes, so that sensors are used to read the temperature, rain, soil moisture, etc., and based on these values, the house systems will be controlled, such as the irrigation system, the umbrella, and others, wirelessly.

All these changes will be viewed to the users by a mobile application connected with the client side.

1 SMART HOME ENERGY SAVINGS. RITHUM HOME. (2023, MARCH 22). HTTPS://RITHUMHOME.COM/SMART-HOME-ENERGY-SAVINGS-TAKE-CONTROL-OF-YOUR-BILLS/

2 Constraints, Standards/ Codes and Earlier course work

2.1 Constraints

- Our lack of experience in mechanical parts makes us face some problems, starting from finding the store or workshops that sell the required parts in an appropriate price
- The two ESPs were so sensitive for hand prints so we had to deal carefully with them.
- Time limit, some sensors were damaged, so finishing on time was challenging.
- The Israeli occupation was putting obstacles and preventing us from reaching the university and Nablus in particular, which led to the completion of the project on time as a kind of challenge.

.

2.2 Earlier Courses

- **Micro-Controllers:** This course taught us the fundamentals of how to program hardware components.
- **Micro-Controllers Lab:** Taught us how to use Arduino Uno**.**
- **CPU Lap:** Assist with the hardware's cabling, welding, and troubleshooting.
- **Critical Thinking and Scientific Research:** make finding information much easier and more professional.
- **Electronic circuits:** made us much familiar with the project components and the correct circuits' design of each.
- **Networks:** Let us know how to deal with protocols and how to make needed http requests with responses.
- **Wireless:** Helped us to learn how to use access points to make request and response between the two ESPs (server & client) to implement (IOT).

3 Literature Review

Smart home technology has become very popular in recent years, but progress wasn't that fast when it first appeared on the scene. In the beginning, a smart home was associated increasingly with tech-savvy homeowners who are open to changes and are quick to adopt them. The first smart home appeared in 1975 with the release of the X10, a home automation platform that used radio frequency pulses to a home's electrical wiring to send digital information so that users could remotely control appliances in their home using command consoles and modules [2].

With time, there has been fierce competition between huge companies such as Google and Amazon. In this paper, we will discuss how our project helped to control systems without human intervention, by using sensors and wireless networks. All the stages that we went through in order to reach the final architecture of the project will also clarified. All modules circuits and their relationship to each other and the codes for all parts will be clarified.

2 STANLEY, J. (2023, JANUARY 27). THE HISTORY OF SMART HOME TECHNOLOGY. FAMILY HANDYMAN. HTTPS://WWW.FAMILYHANDYMAN.COM/ARTICLE/THE-HISTORY-OF-SMART-HOME-TECHNOLOGY/

4 Design and Implementation

We designed a wooden box to fit the house with its ward so we could implement Livesmart in different stages starting from using the first ESP32 for reading needed information from used sensors , then send these data wirelessly to the second ESP32 which is responsible about controlling in some features that are exist in the house . And the final step is a water pump in the ward used to irrigate the plants after making sure it need water by checking the humidity sensor for its soil.

4.1 The design

Figure 4.1.1: The design

4.2 Hardware and Software Specifications

4.2.1 Hardware Components

Hardware components for Livesmart:

For this part, we have used two different ESP-WROOM**,** Arduino Uno**,** External Temperature sensor (DS18B20), Soil moisture sensor, Rain sensor, Solar panel with chargeable batteries, DC-GEAR motor,12V Relay module, L298N Dual H-Bridge motor driver module and water pump.

- **ESP32-WROOM**

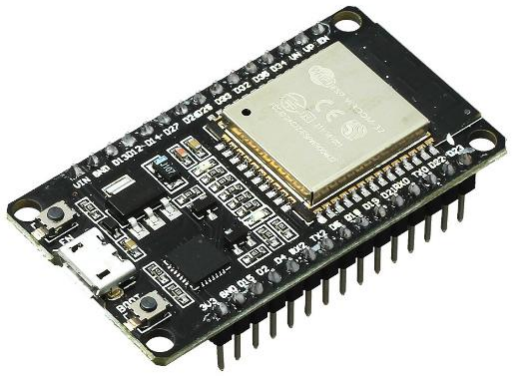

Figure 4.2.1: ESP32-WROOM

ESP32-WROOM-32 is a powerful, generic Wi-Fi module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks. Here, we used the first ESP32 in reading the data from the temperature and humidity sensors by connecting them and debug there code to install it on the ESP32, then send this redden data wirelessly to the second ESP32 that have to control things according the redden data.

- **Arduino Uno**

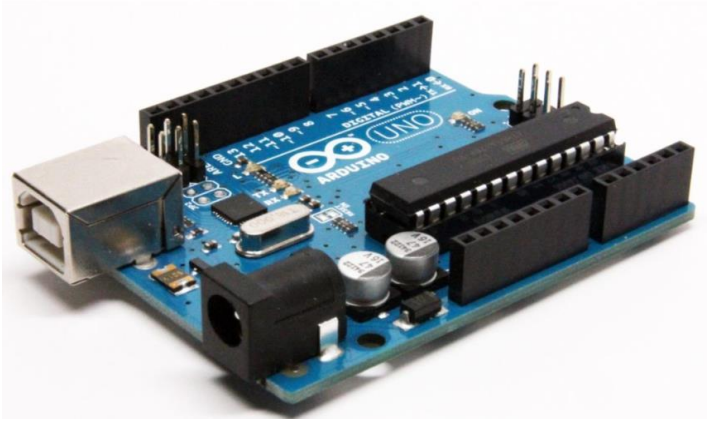

Figure 4.2.2: Arduino Uno

Arduino UNO is a microcontroller board based on the ATmega328P.It has 14 digital input/output pins, it contains everything needed to support the microcontroller. We used it with the DC-Gear motor to control the umbrella.

- **External Temperature Sensor(DS18B20):**

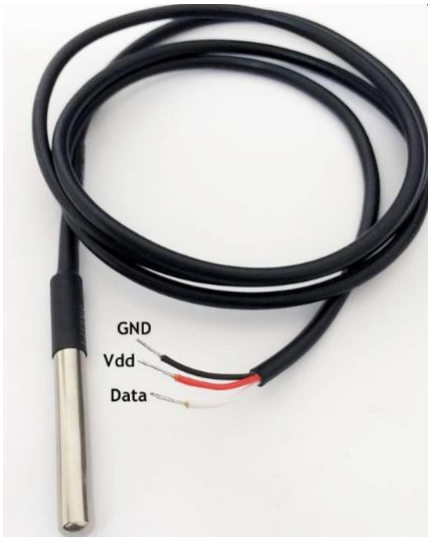

Figure 4.2.3: DS18B20

An External Temperature Sensor is sensor work with supply voltage 3~5.5v containing 3pins as shown in figure 4.2.2 , its sensing local temperature -55 °C ~ 125 °C with sensor of type digital, local . Here, we used this sensor to sense the temperature and save the data with the esp to control the situation then.

- **Soil moisture sensor**

Figure 4.2.4: Soil moisture sensor

The Soil Moisture Sensor utilizes capacitance to quantify the water substance of soil. Here, we used it to test the water of the plants soil, if it is rare and they want water the connected pump will be turned on and irrigate the plants.

- **Rain Sensor**

Figure 4.2.5: Rain Sensor

This is a detector of rain with a wide detection area on both sides and control electronics, Here, we used it to detect presence of rain, if so, data will be sent to the first ESP (server) and it will send the needed instructions to the second ESP (client) to make the umbrella opened.

- **L298N Dual H-Bridge**

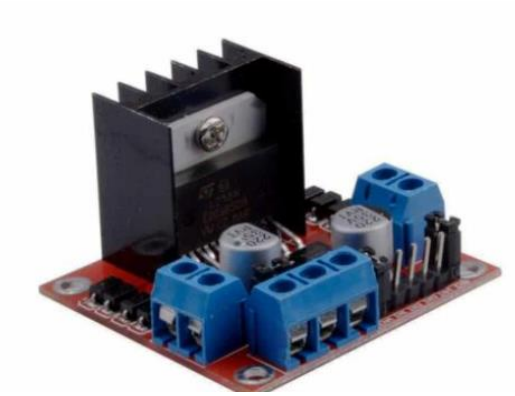

Figure 4.2.6: L298N

The L298N Dual Motor Controller H-Bridge Driver Module contains a dual channel full H-Bridge driver which allows for up to 2 DC motors to be independently controlled in both forward and reverse directions as well as speed. We used it with the DC-Gear motor to control opening and closing the umbrella with presence of rain.

- **12V Relay Module**

Figure 4.2.7: Relay module

It's a single channel relay module with 12V by using it we could set the trigger mode: active on low or high. We used two of this relay , one with the water pump and the second with th DC-Gear motor to control high voltage circuits for both.

- **Water Pump**

Figure 4.2.8: water pump

The water pump is connected with the soil moisture sensor to irrigate the plants when needed

- **Solar Panel System**

Figure 4.2.9: Solar Panel

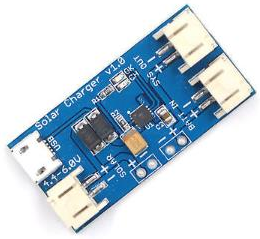

Figure 4.2.10: Solar Charger Board

Figure 4.2.11: Li-rechargeable battery

A solar panel is a device that converts sunlight into electricity by using photovoltaic (PV) cells system so the charger board is used to save the converted electricity into lithium rechargeable battery so we could use this electricity to light a connected led for the house.

- **DC-GEAR Motor**

Figure 4.2.12: DC-GEAR Motor

A dc gear motor is an electric motor and a power reducer combined into a single unit that reduces the number of revolutions but increases the torque of the operating shaft .Here, we used it to pull up the house umbrella due to the second ESP32 instructions, the umbrella will be pulled if the weather is very sunny or if it is raining.

4.3 Arduino Code

```
int enA = 9;
int in1 = 8;
int in2 = 7;

void setup()

{
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(2, INPUT);
  pinMode(4, INPUT);

}

void controlGear()

{
  if(digitalRead(4)==HIGH && digitalRead(2)==HIGH)
  {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 255);

  }

 else if(digitalRead(4)==LOW && digitalRead(2)==HIGH)
 {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    analogWrite(enA, 255);
 }

 else
 {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
 }

}

void loop()
{

  controlGear();
  delay(10000);
}
```

4.4 ESP32 Code (server):

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include <SPIFFS.h>
#include <DS18B20.h>


#define rainDigital 2
#define moisAnalog 36 //SP


const char* ssid = "Ahmad Nazzal";
const char* password = "0598209314";

DS18B20 ds(4);
AsyncWebServer server(80);



String readTemp()
{
  int value;
  if(ds.selectNext())
  {
    value = ds.getTempC();
    Serial.println(value);
    return String(value);
  }
  else return  readTemp();
}

String readMois()
{
  int value=analogRead(moisAnalog);
  Serial.println(value);

  if(value>1500)
  {
    return String("1");//Dry --> operate the system
  }

  else
  {
    return String("0");//Wet --> stop the system
  }

}
```

```
String readRain()
{
  int value=digitalRead(rainDigital);
  Serial.println(value);

  if(!value &&digitalRead(26) == LOW && digitalRead(27) == HIGH) //check Limit switch
  {
    return String("1"); // Raining --> open umbrella
  }

  else if (if(!value &&digitalRead(12) == LOW && digitalRead(14) == HIGH) //check Limit switch)
  {
    return String("2"); //Raining && the umbrella already closed --> Do nothing
  }

  else if (if(!value &&digitalRead(12) == LOW && digitalRead(14) == HIGH) //check Limit switch)
  {
    return String("0"); //Not raining --> stop the system
  }
}


void setup()
{
  Serial.begin(115200);
  Serial.println();
  WiFi.begin(ssid, password);
  SPIFFS.begin();

  if (WiFi.waitForConnectResult() != WL_CONNECTED)
  {
    Serial.printf("WiFi Failed!\n");
    return;
  }
  Serial.print("IP Address: "); Serial.println(WiFi.localIP());
  server.serveStatic("/", SPIFFS, "/").setDefaultFile("index.htm");

  server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request)
  { request->send(200, "plain/text", readTemp());});

  server.on("/moisture", HTTP_GET, [](AsyncWebServerRequest *request)
  { request->send(200, "plain/text", readMois()); });

  server.on("/rain", HTTP_GET, [](AsyncWebServerRequest *request)
  { request->send(200, "plain/text", readRain());});

  server.begin();
}

void loop(){
}
```

(Client):

```
String httpGETRequest(const char* serverName)
{
  WiFiClient client;
  HTTPClient http;

  http.begin(client, serverName);

  // Send HTTP POST request
  int httpResponseCode = http.GET();

  String payload = "--";

  if (httpResponseCode>0)
  {
    payload = http.getString();
  }

  else
  {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }

  http.end();
  return payload;
}
```

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "Ahmad Nazzal";
const char* password = "0598209314";

//Your IP address or domain name with URL path
const char* serverNameTemp = "http://192.168.1.63/temperature";
const char* serverNameMois = "http://192.168.1.63/moisture";
const char* serverNameRain = "http://192.168.1.63/rain";


#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define RELAY_PIN 17

String temperature;
String moisture;
String rain;

unsigned long previousMillis = 0;
const long interval = 5000;
```

```
26  void setup() {
27    Serial.begin(115200);
28    pinMode(RELAY_PIN, OUTPUT);
29
30    WiFi.begin(ssid, password);
31    Serial.println("Connecting");
32
33    while(WiFi.status() != WL_CONNECTED)
34    {
35      delay(500);
36      Serial.print(".");
37    }
38
39    Serial.println("");
40    Serial.print("Connected to WiFi network with IP Address: ");
41    Serial.println(WiFi.localIP());
42
43  }
44
45  void irrigationSystem(String M,String R)
46  {
47    if (M.equals("1")&&R.equals("0")) // operate the system
48    {
49      Serial.println("open pumb");
50      digitalWrite(RELAY_PIN, HIGH);
51    }
52
53    else //stop the system
54    {
55      Serial.println("cloes pumb");
56      digitalWrite(RELAY_PIN, LOW);
57    }
58  }
59
60  void umbrellaSystem(String R)
61  {
62    if (R.equals("1")) // operate the system
63    {
64      Serial.println("open umbrella");
65    }
66
67    else //stop the system
68    {
69      Serial.println("cloes umbrella");
70    }
71  }
72
```

```
void loop() {
  unsigned long currentMillis = millis();

  delay(5000);
  if(currentMillis - previousMillis >= interval)
  {
    if(WiFi.status()== WL_CONNECTED ){
      temperature = httpGETRequest(serverNameTemp);
      moisture = httpGETRequest(serverNameMois);
      rain = httpGETRequest(serverNameRain);

      Serial.println("Temperature: " + temperature + " *C");
      irrigationSystem(moisture,rain);
      umbrellaSystem(rain);

      // save the last HTTP GET Request
      previousMillis = currentMillis;
    }

    else
    {
      Serial.println("WiFi Disconnected");
    }

  }
}

```

5 Results and Analysis

Prior to moving on to the next stage, we compiled the source code for each step and verified that everything was functioning properly. This greatly assisted us in identifying the cause of the issues.
Our project ultimately met all of the standards associated with it like sensing the rain, temperature and the sun to deal with lights and the umbrella. Also, check the soil for irrigating the plants.

6 Conclusions and Recommendation

6.1 Conclusions and recommendations

In our project we created a smart home with its farm as an implementation of the (IOT) in addition to a mobile application to notify the owner about the situation and what is going to happen according to it.

This helped us to improve our skills with using Arduino and ESP, also knew how to deal with different sensors with correct way to but all of them together.

6.2 Future Work

We are planning to improve our project by making it all working with the solar panel in addition to make the water pump as a well and controlling its filling with water from the rain.

7 References

[1] Smart Home Energy Savings. Rithum Home. (2023, March 22). https://rithumhome.com/smart-home-energy-savings-take-control-of-your-bills/

[2] Stanley, J. (2023, January 27). The history of smart home technology. Family Handyman. https://www.familyhandyman.com/article/the-history-of-smart-home-technology/