People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA - Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Project Report Presented in Partial Fulfilment for
The Requirements of the Degree of
'LICENCE'
In Electrical and Electronic Engineering

Title:

Modern Smart House Automation using IoT Technology

Supervisor: Dr F. Mouhouche

Presented by:

. Naci Mouhamed Aiman

. Benzerga Abderraouf

Promotion :2020

Contents

List of FIGURES:

Liste of abbreviation:

IOT: Internet Of Things.

LDR: Light Dependent Resistors.

RadioRA: whole home lighting controlling system.

X10: communication protocol.

Zigbee: high-level communication protocols used to create personal area networks with small, low-power digital radios.

Zwave: a wireless communications protocol used primarily for residential and commercial building automation.

AR: augmented reality.

VR: Virtual reality.

Thermostat: temperature regulating device.

LE MCU: wireless microcontrollers.

Ethernet: family of wires.

UART: Universal Asynchronous Receiver Transmitter.

SPI: Serial Peripheral Interface.

I2S: Integrated Interchip Sound.

I2C: Inter-Integrated Circuit.

ATMEGA 328: high-performance Microchip.

ICSP header: In-Circuit Serial Programming.

FTID: Future Technology Devices International Limited.

IR: Infrared port.

RPM: evolution per minute.

JSON: JavaScript Object Notation.

Acknowledgement:

I would like to express my sincere gratitude to my project supervisor Dr F. Mouhouche for their invaluable guidance, support, and expertise throughout the duration of this project. Their insightful feedback, constructive criticism, and encouragement have been instrumental in helping us to complete this project to the best of our abilities. Their dedication and commitment to ensuring our success have been truly inspiring and we are grateful for their time and efforts. we also want to thank her for sharing their wealth of knowledge and experience with us, and for constantly pushing us to exceed our own expectations. Thank you, Dr F. Mouhouche, for everything you have done for us.

Abstract

This work presents the implementation of a smart house automation system using Internet of Things (IoT) technology. The project aimed to create a connected ecosystem of IoT devices and technologies to enhance the functionality of a traditional household. Key objectives included remote control and monitoring of home appliances, energy management, enhanced security, and personalized user experiences. The system integrated various IoT devices, utilized a centralized IoT hub, and employed cloud-based platforms for data processing and analytics. The project demonstrated the potential of IoT in transforming homes into smart and connected living spaces, contributing to the field of smart home automation.

Introduction:

The advent of the Internet of Things (IoT) has revolutionized the way we interact with our surroundings, leading to the emergence of smart homes. Smart home automation systems leverage IoT technologies to connect and control various devices, appliances, and systems within a household, offering improved convenience, efficiency, and security to homeowners. This report presents the implementation and findings of a project that aimed to develop a comprehensive smart house automation system using IoT.

The objective of this project was to create an intelligent ecosystem that seamlessly integrates different IoT devices and technologies to enhance the functionality of a traditional household. By connecting appliances, sensors, and actuators, the system enables remote control and monitoring of home functions, optimizes energy consumption, enhances security measures, and provides personalized user experiences through intelligent applications.

The project utilized a combination of IoT devices, including smart thermostats, lighting systems, door locks, and motion sensors, among others. These devices were interconnected through a centralized IoT hub or gateway, which served as the control centre of the system. Data collected from these devices was transmitted to and processed in the cloud, leveraging cloud-based platforms for advanced analytics, data storage, and communication.

Energy management was a primary focus of the project. Real-time energy consumption data from connected devices allowed for intelligent automation and optimization of energy usage. By analysing usage patterns, the system could automatically adjust temperature settings, turn off lights in unoccupied rooms, and manage power consumption of appliances, thus promoting energy efficiency and conservation.

Security was another crucial aspect addressed in the project. Integration of, motion sensors, and smart locks enabled remote monitoring and control of the home's security. Users received instant notifications on their connected devices in case of any suspicious activities, unauthorized access attempts, or emergencies, ensuring a heightened sense of safety and peace of mind.

The outcomes of this project contribute to the growing field of smart home automation and highlight the potential of IoT technologies in transforming traditional homes into connected and intelligent living spaces. The findings provide insights into the feasibility and benefits of implementing smart house automation systems, paving the way for further research and development in this area.

In the following sections, we will detail the implementation process, discuss the key features and functionalities of the system, present the results and observations, and conclude with an evaluation of the project's success in achieving its objectives.

Chapter 1: Smart House History, Benefits and Obstacles.

1.Smart Home Technology History - From Dreams to Reality:

1.1. The Conceptual Origins of Smart Homes:

The idea of a smart home can be traced back to the early 20th century, when science fiction writers and futurists began to envision homes with various automated features. In the 1930s, technological advancements such as electricity and modern appliances began to revolutionize the way people lived. In the 1950s, the advent of mainframe computers sparked new ideas about the possibilities of home automation. In 1966, Jim Sutherland created the Electronic Computing Home Operator (ECHO), a mainframe computer system designed to control various household appliances. In the 1970s and 1980s, microprocessors and personal computers further fueled the imagination of tech enthusiasts and futurists. In the decades that followed, innovations in computing, networking, and wireless technologies would pave the way for the modern smart homes we know today. [1]

1.2. Early Home Automation Systems:

Early home automation systems, such as X10, emerged in the mid-1970s and allowed homeowners to control appliances, lighting, and other devices through a wired network.

In the 1980s, other systems, such as the BSR System X-100 and the Lutron HomeWorks, expanded on the capabilities of X10 by offering more advanced features and improved reliability. In the 1990s, wireless home automation systems, such as the RadioRA system by Lutron, provided increased flexibility and ease of installation, paving the way for modern smart homes. [2]

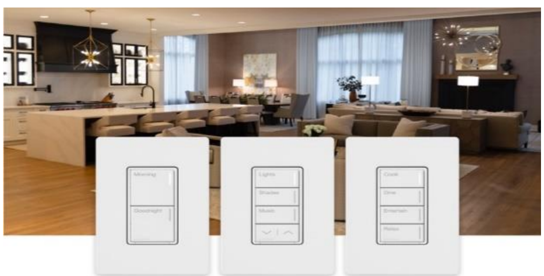


Figure 3 Lutron HomeWorks

Figure 1 X10 system

1.3. The Birth of the Internet and Its Impact on Smart Home:

The advent of the internet in the late 20th century has had a profound influence on the evolution of smart home technology. It enabled seamless connectivity, data exchange, and remote access, leading to the development of the Internet of Things (IoT). Home automation systems such as X10 allowed homeowners to control appliances, lighting, and other devices through a wired network. The IoT revolutionized the smart home industry, enabling devices to communicate with each other, learn from user behavior, and respond to changing conditions autonomously. Smartphones and tablets have further enhanced the impact of the internet on smart home technology. [3]

1.4. The Rise of Wireless Technologies and IoT:

The development of wireless technologies in the late 20th and early 21st centuries has had a major impact on the evolution of smart home systems. Wi-Fi, Zigbee, Z-Wave, and Bluetooth were key milestones in the history of wireless technologies, allowing devices to communicate and exchange data without physical connections. The IoT revolutionized the smart home landscape by allowing devices to communicate with each other, learn from user behavior, and respond to changing conditions autonomously. As wireless technologies and IoT advanced, a new generation of smart home devices emerged, offering greater ease of use, interoperability, and personalization. From smart thermostats and lighting systems to voice-controlled assistants and advanced security solutions, the rise of wireless technologies and IoT has laid the groundwork for the intelligent, interconnected homes of today and tomorrow. [4]

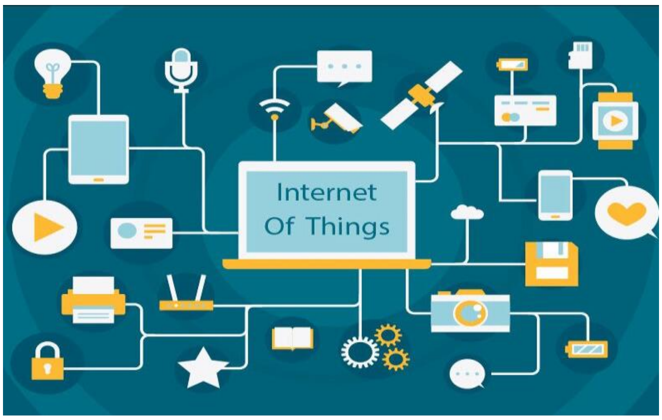


Figure 4:internet of things.

Figure 5:wireless technologies.

1.5. Voice Assistants and Their Influence on Smart Home Adoption:

Voice assistants in the early 21st century marked a significant milestone in the evolution of smart home technology. Apple's Siri in 2011 marked the beginning of a new era in voice-controlled technology, while Amazon's Echo and Google Home in 2016 both offered voice-activated features and served as a central hub for smart home devices. Voice assistants have also led to increased collaboration between smart home manufacturers, making it easier for users to mix and match devices from different manufacturers while maintaining a seamless control experience. As a result, voice assistants have played an instrumental role in driving the widespread adoption of smart home systems and shaping the intelligent, interconnected homes of the present and future. [4]

Figure 6:voice assistants.

1.6. The Future of Smart Homes:

Smart home technology is expected to evolve in the future, with AI and machine learning integration, greater interoperability and standardization, enhanced security and privacy, and energy efficiency and sustainability. These trends and predictions will shape the future of smart homes, making them more comfortable, efficient, and secure. The future of smart homes is expected to be exciting and dynamic, with new technologies and applications that will further enhance our everyday lives. These technologies include energy management systems, AR and VR[7], and applications for healthcare, education, and entertainment. This will lead to a new era of convenience, efficiency, and sustainability. [4]

2.Benefits:

The benefits of a modern smart house using IoT are numerous and can have a significant impact on the homeowner's quality of life. Here are some of the key benefits of a modern smart house using IoT:

1. Convenience: One of the most significant benefits of a smart home using IoT is the convenience it offers. Homeowners can control various aspects of their home remotely, making it easier to manage their home even when they are not physically present. For example, they can turn off lights or adjust the temperature from their smartphone or voice assistant, without having to get up from their couch.
2. Energy Efficiency: Smart home devices can help homeowners save energy and reduce their utility bills. For example, a smart thermostat can learn the homeowner's preferences and adjust the temperature automatically, which can save energy and money. Similarly, smart lighting systems can be programmed to turn off when no one is in the room, reducing energy consumption.
3. Improved Security: Smart home security systems can provide added peace of mind by allowing homeowners to monitor their home remotely and receive alerts in case of unusual activity. Some systems also come with built-in cameras and motion sensors for added security.
4. Customization: A smart home using IoT allows homeowners to customize their living space to their unique needs and preferences. They can program the lighting, temperature, and other aspects of their home to match their schedule and lifestyle.
5. Accessibility: Smart home devices can make it easier for people with disabilities or limited mobility to live independently. For example, they can control their home using voice commands, making it easier to manage various aspects of their home without relying on physical movement.
6. Improved Home Value: A smart home using IoT can increase the value of the homeowner's property. Potential homebuyers are increasingly interested in properties that have smart home features, making it easier to sell the home in the future.

Overall, a modern smart house using IoT offers numerous benefits for homeowners. It provides added convenience, energy efficiency, security, customization, accessibility, and can even increase the value of the homeowner's property. As technology continues to advance, it is likely that the benefits of a smart home using IoT will only increase, making it an even more appealing option for homeowners. [5]

3.Obstacles to generalize smart house idea:

While smart homes offer many benefits such as improved energy efficiency, comfort, and convenience, they also come with their own set of challenges and potential problems. Some of the key issues and concerns that arise with smart homes include:

1. Security and privacy risks: Smart homes are vulnerable to security breaches and hacking, which can compromise the privacy and safety of homeowners. This is especially concerning for devices that collect personal data, such as cameras, voice assistants, and smart locks.
2. Compatibility issues: With so many different types of smart devices available on the market, compatibility can be a challenge. Devices from different manufacturers may not work well together, or require complicated setups and configurations.
3. Complexity: Smart home systems can be complex and difficult to set up and use, requiring a high degree of technical expertise and knowledge.
4. Dependence on technology: Smart homes rely heavily on technology, which can be a problem when there are glitches, outages, or other technical issues that disrupt the system.
5. Cost: Smart home devices can be expensive, and the cost of upgrading an entire home to a smart system can be prohibitive for many homeowners.
6. Maintenance: Smart home systems require regular maintenance and updates to ensure optimal performance, which can be time-consuming and costly.
7. Ethical concerns: There are ethical concerns surrounding the collection and use of personal data by smart home devices, as well as the potential for these devices to perpetuate existing biases and inequalities.

Overall, while smart homes offer many benefits, it is important for homeowners and manufacturers to be aware of the potential problems and challenges associated with these systems. Addressing these issues through improved security measures, standardization, and better education and support for users can help ensure that smart homes are safe, effective, and sustainable over the long term. [5]

Chapter2: hardware overview and systems used.

1.Arduino platform:

Arduino is an open-source electronics platform that is designed to enable users, from beginners to experienced engineers, to create interactive projects with a range of electronic components. It consists of a hardware platform that includes a microcontroller, and a software platform that provides a programming environment for creating code that runs on the microcontroller.

The hardware platform of the Arduino consists of a printed circuit board (PCB) that features a microcontroller, a set of input and output pins, and a USB connector for power and communication. The microcontroller is the heart of the system, and it provides the processing power and memory required to run code and control the electronic components connected to the board.

The software platform of the Arduino consists of a programming environment that is based on the Processing language and the Wiring framework. This environment provides a simple, easy-to-use interface for writing code, uploading it to the board, and controlling the electronic components connected to the board.

One of the key features of the Arduino platform is its versatility. The platform is designed to work with a wide range of electronic components, including sensors, motors, lights, and displays. This makes it possible to create a wide variety of projects, ranging from simple LED blinkers to complex robotics projects. [6]

2.Arduino IoT Cloud:

The Arduino IoT Cloud lets you effortlessly create, develop, and manage connected Arduino-based projects. To get started, link a physical development board to your Arduino account and register it in the IoT cloud web application. Then, you can create a “thing” that represents your project. Such a thing contains the managed variables, the network credentials, and the source code of your project. The IoT Cloud application allows you to define as many variables as you like. The Arduino then synchronizes the values of these variables with the cloud, and powerful data-aggregation and display tools let you visualize the variable values and the overall state of your project without having to write a single line of code. For that purpose, you can create custom dashboards that display data and let you interact with your connected devices to change settings from anywhere you are. [7]

Figure 7:dash board.

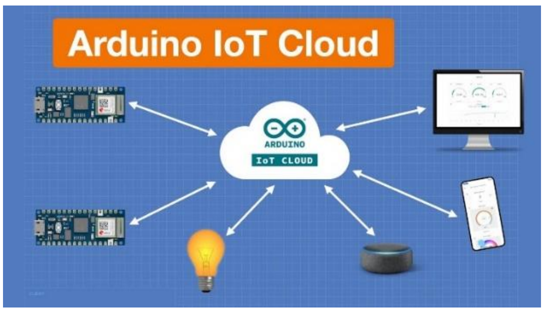


Figure 8:arduino iot cloud.

3.Microcontrollers:

3.1ESP32-WROOM-32:

ESP32-WROOM-32 is a powerful, generic Wi-Fi + Bluetooth® + Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz's The chip also has a low-power coprocessor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S, and I2C. [8]

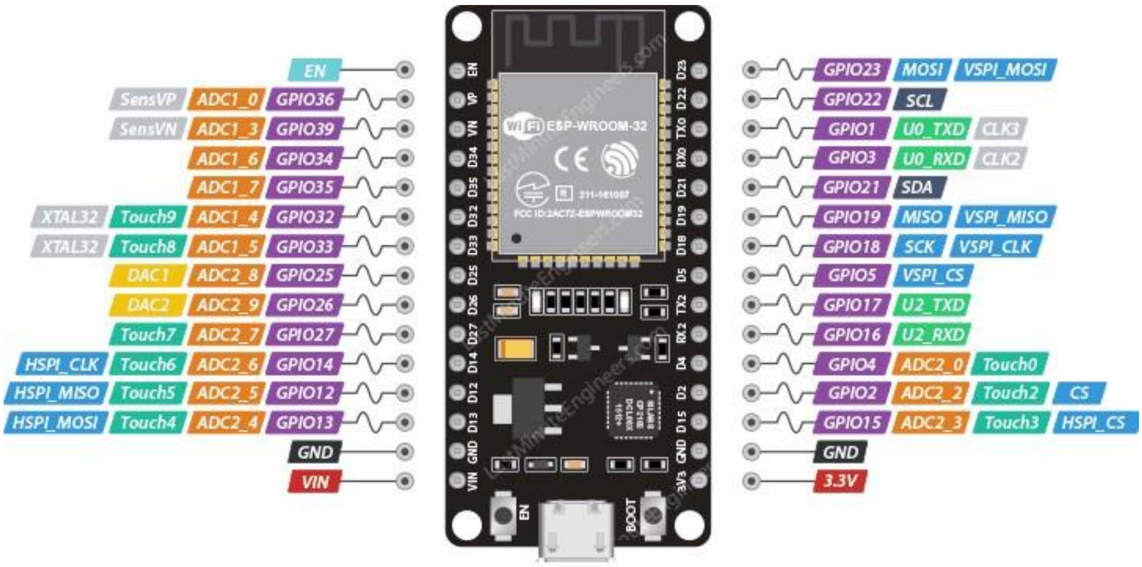


Figure 9:ESP32 pinout.

3.2Arduino Uno:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. [9]

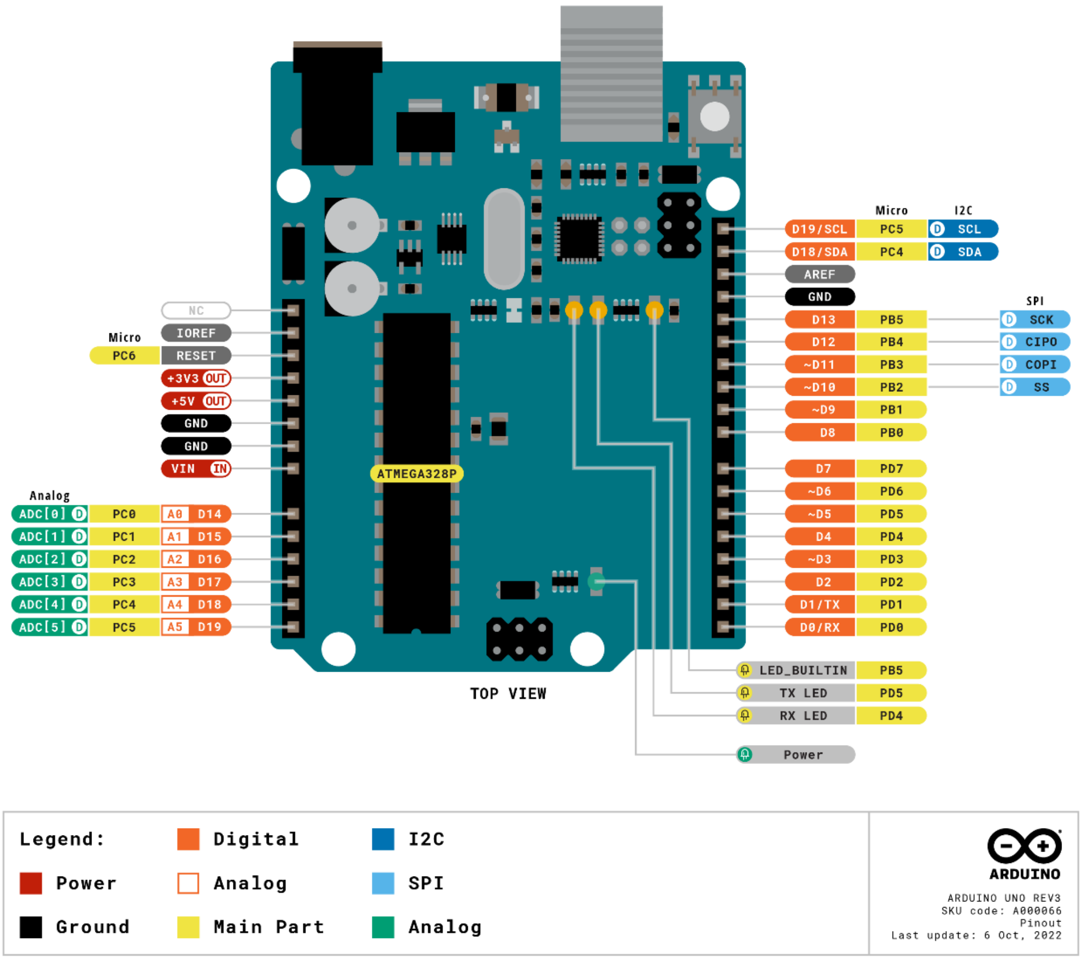


Figure 10:arduino uno pinout.

4.Sensors:

4.1 Light Dependent Resistor (LDR):

Photoresistors, also known as light dependent resistors (LDR), are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. In the dark, their resistance is very high, sometimes up to 1 MΩ, but when the LDR sensor is exposed to light, the resistance drops dramatically, even down to a few ohms, depending on the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices. They are used in many applications, but this light sensing function is often performed by other devices such as photodiodes and phototransistors. Some countries have banned LDRs made of lead or cadmium over environmental safety concerns. [10]

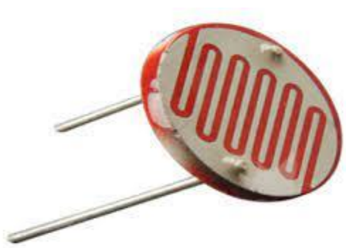


Figure 12:LDR.

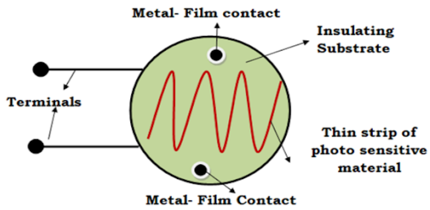


Figure 11:LDR construction.

4.2 Gaz sensor (MQ2):

The gas sensing material used in the MQ-2 gas sensor is tin dioxide (SnO_2), which has low conductivity in clean air. When there is flammable gas in the environment where the sensor is located, the conductivity of the sensor increases with the increase of the flammable gas concentration in the air. The MQ-2 gas sensor has a high sensitivity to liquefied gas, propane, and hydrogen, and is also ideal for the detection of natural gas and other combustible vapors. This sensor can detect a variety of flammable gases and is a low-cost sensor suitable for a variety of applications. [10]

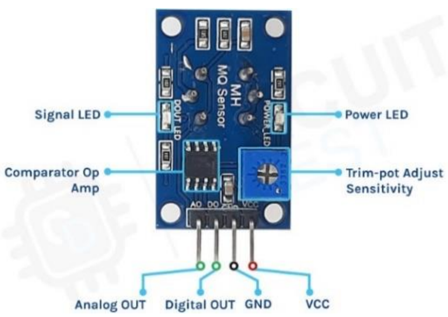


Figure 14:gaz sensor pins.

Figure 13:gaz sensor.

4.3 Flame sensor:

A Flame Sensor module or Fire Sensor module is a small size electronics device that can detect a fire source or any other bright light sources. This sensor basically detects IR (Infrared) light wavelength between 760 nm – 1100 nm that is emitted from the fire flame or light source. The flame sensor comes with a YG1006 Phototransistor sensor which is a high speed and high sensitivity. Two types of IR Infrared Flame Sensor Module available in the market one having three pins (D0, Gnd, Vcc) and another one having four pins (A0, D0, Gnd, Vcc) both are can be easily used with Arduino and other microcontroller boards. [10]

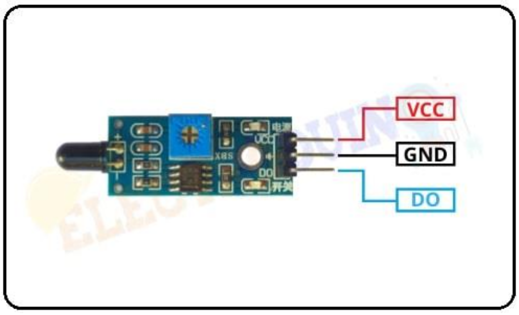


Figure 15:flame sensor pins.

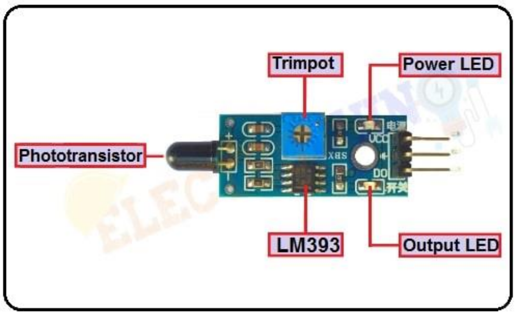


Figure 16:flame sensor components.

4.4 Digital Humidity and temperature sensor:

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. [10]

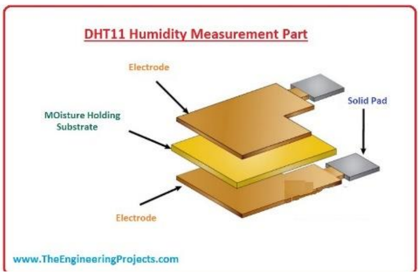


Figure 18:DHT sensor measurement part.

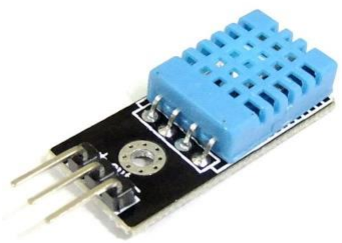


Figure 17:DHT sensor.

4.5 Water level sensor:

The water level sensor / leak detection sensor is a 3-pin module that outputs an analogue signal (generally 0 to 500) that indicates the approximate depth of water submersion. When used in conjunction with a pull up resistor, it can be used as a digital device to indicate the presence or water. The sensor has a series of ten exposed copper traces, five of which are power traces and five are sense traces. These traces are interlaced so that there is one sense trace between every two power traces. Usually, these traces are not connected but are bridged by water when submerged. The series of exposed parallel conductors, together acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water level. The change in resistance corresponds to the distance from the top of the sensor to the surface of the water. The resistance is inversely proportional to the height of the water. [10]

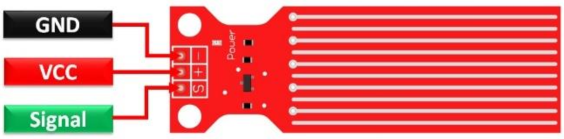


Figure 19:water level sensor pins

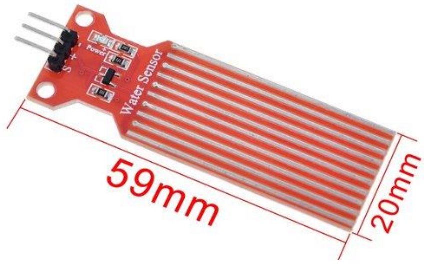


Figure 20:water level sensor.

4.6 Ultrasonic sensor:

The HC-SR04 Ultrasonic Distance Sensor is a sensor used for detecting the distance to an object using sonar. It's ideal for any robotics projects you have which require you to avoid objects, by detecting how close they are you can steer away from them. The HC-SR04 uses non-contact ultrasound sonar to measure the distance to an object, and consists of two ultrasonic transmitters (basically speakers), a receiver, and a control circuit. The transmitters emit a high frequency ultrasonic sound, which bounce off any nearby solid objects, and the receiver listens for any return echo. That echo is then processed by the control circuit to calculate the time difference between the signal being transmitted and received. This time can subsequently be used, along with some clever math, to calculate the distance between the sensor and the reflecting object. [10]

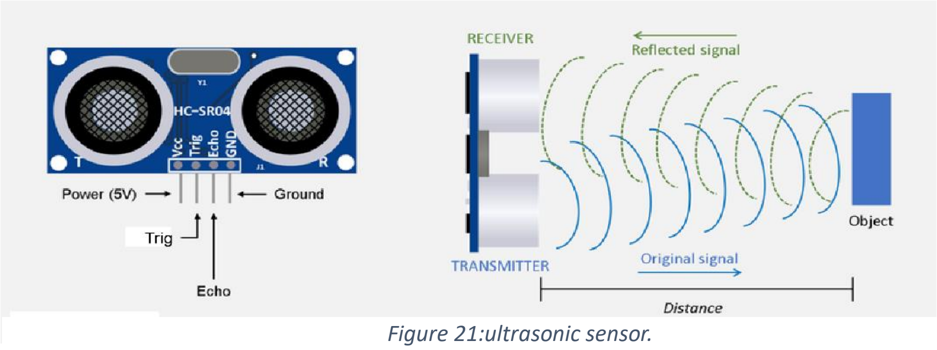


Figure 21:ultrasonic sensor.

4.7 RFID:

MF RC522 is a highly integrated read and write card chip applied to the 13.56MHz contactless communication. Launched by the NXP Company, it is a low-voltage, low-cost, and small-sized non-contact card chip, a best choice for intelligent instrument and portable handheld devices. The MF RC522 uses advanced modulation and demodulation concept which fully presented in all types of 13.56MHz passive contactless communication methods and protocols. In addition, it supports rapid CRYPTO1 encryption algorithm to verify MIFARE products. MFRC522 also supports MIFARE series of high-speed non-contact communication, with a two-way data transmission rate of up to 424kbit/s. As a new member of the 13.56MHz highly integrated reader card series, MF RC522 is much similar to the existing MF RC500 and MF RC530 when there are also great differences. It communicates with the host machine via the serial manner which needs less wiring. You can choose between SPI, I2C and serial UART

mode (similar to RS232), which helps reduce the connection, save PCB board space (smaller size), and reduce cost. [10]

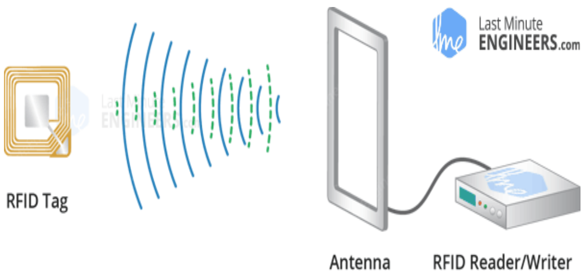


Figure 23:RFID concept.

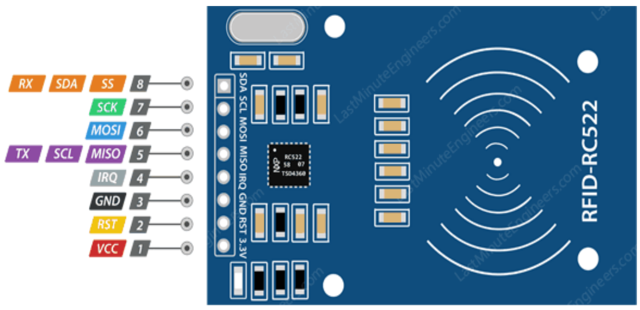


Figure 22:RFID pins.

5. Relay Module:

Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not.

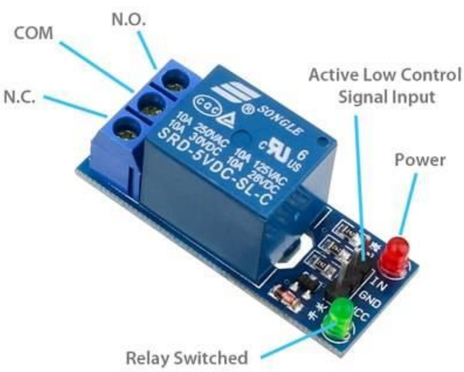


Figure 24:relay module pins.

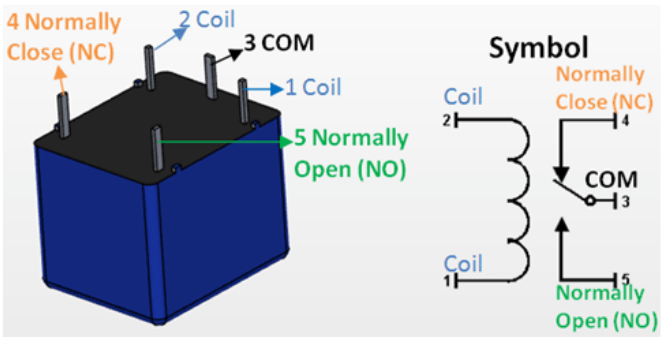


Figure 25:relay module construction.

Single-Channel Relay Module Specifications:

Supply voltage – 3.75V to 6V

Quiescent current: 2mA

Current when the relay is active: ~70mA

Relay maximum contact voltage – 250VAC or 30VDC

Relay maximum current – 10A

6. Servo motor (SG90):

A Servo motor is a type of motor that is powered by a DC source, either from an external supply or by a controller. A small and lightweight servo motor with high output power is called a micro servo motor sg90. This means that the sg90 micro servo motor will only work as hard as is required to complete the task at hand. A wide range of applications for servo motors exists, including cameras, telescopes, antennas, industrial automation, and robots. A motor rotates from 0 to 180 degrees at each position of 90 degrees so that names it SG90. Servo motors have a gear that reduces the rotational speed of the motor by reducing its RPM and increasing the torque.

Figure 26:servo motor pins.

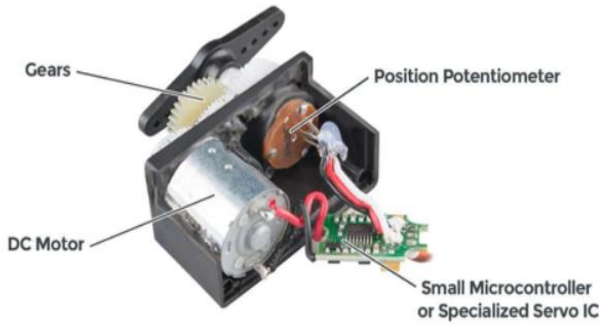


Figure 27:servo motor construction.

Chapter 3: Design & Development of the system

Circuits Diagrams connections of the project:

ESP32 circuit:

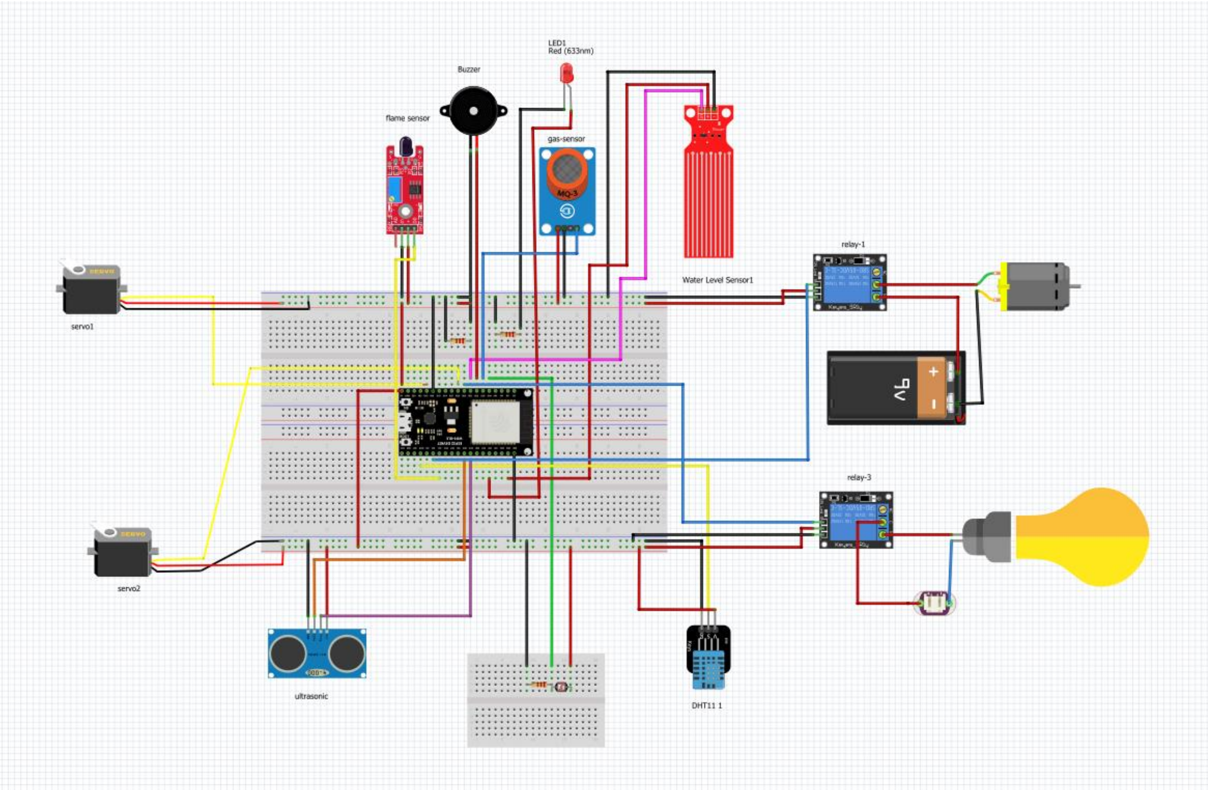


Figure 28:esp32 circuit.

Arduino uno circuit:

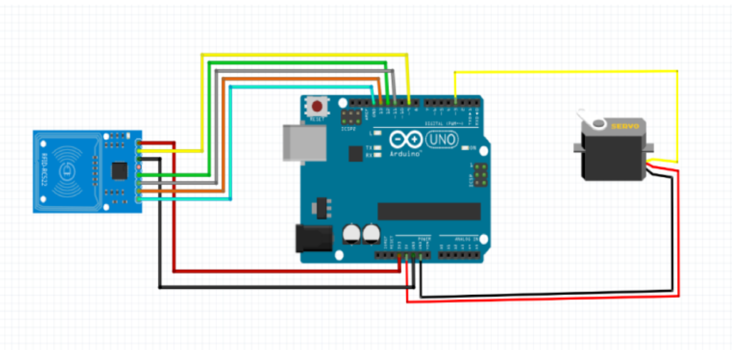


Figure 29:Arduino uno circuit.

1.Smart Garage Automation:

1.1. Ultrasonic Sensor and servo implementation:

In this part we will discuss about smart garage using servomotor and ultrasonic which will measure the distance between the sensors and the object then display the measured distance on the serial monitor then make a test on that distance, if it is less than or equal to 10 cm it means object in range and garage goes ON and our garage will rotate 0 to 90 degree .And if distance is greater than 10 cm alternately garage will off and motor will begin to rotate in the other direction until it returns to the home position.

```
//.....garage-door.....
digitalWrite(trigpin,LOW);
delayMicroseconds(20);
digitalWrite(trigpin,HIGH);
delayMicroseconds(100);
digitalWrite(trigpin,LOW);
 duration = pulseIn(echopin,HIGH);
 distance = duration * 0.034 / 2;
 Serial.print("distance is: ");
 Serial.print(distance);
 Serial.println(" cm");
 if (distance < 10){


  myservo1.write(90);    }
 else{

  myservo1.write(0);
  }
```

Figure 31:smart garage code.

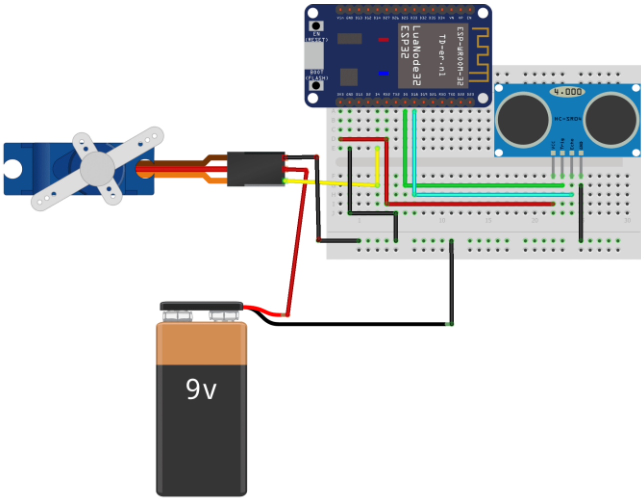


Figure 30:smart garage circuit.

2.Smart locks and Keyless Entry:

2.1. RFID and Servo motor implementation:

At a rudimentary level, RFID systems consist of three main components: an RFID tag or smart label, an RFID reader, and an antenna. RFID tags contain an IC and an antenna, which are used to transmit data to the RFID reader (also called an interrogator). Each RFID tag has its key. So, when the RFID tag has been scanned by the reader, it tries to find whether the key of the tag matches the authorized key. If the key matches, then the door will open with hand of servo motor.

```
void loop() {

  if ( ! rfid.PICC_IsNewCardPresent())
    return;
  if ( ! rfid.PICC_ReadCardSerial())
    return;
  Serial.print("NUID tag is :");
  String ID = "";
  for (byte i = 0; i < rfid.uid.size; i++) {
    ID.concat(String(rfid.uid.uidByte[i] < 0x10 ? " 0" : " "));
    ID.concat(String(rfid.uid.uidByte[i], HEX));
    delay(300);
  }
  ID.toUpperCase();
  if (ID.substring(1) == UID && lock == 0 ) {
    servo.write(0);
    Serial.print("Door is locked");
    delay(1500);
    lock = 1;
  } else if (ID.substring(1) == UID && lock == 1 ) {
    servo.write(180);
    Serial.print("Door is open");
    delay(1500);

    lock = 0;
  } else {

   Serial.print("Wrong card!");
    delay(1500);
```

Figure 33:smart locks code.

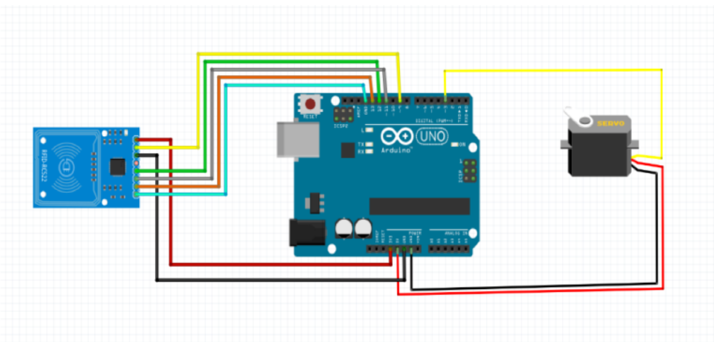


Figure 32:smart locks circuit.

3.Air conditioning system:

An air conditioning system with DC motor control using a DHT11 sensor and ESP32 is a setup that combines various technologies to create an efficient and automated cooling solution. Here's the working principle of it.

3.1 dc motor controlling using DHT11 sensor data:

The ESP32 reads temperature and humidity data from the DHT11 sensor using its digital input/output pins. It compares the obtained values with a predefined temperature and humidity setpoint. Based on this comparison, the ESP32 determines whether to activate the Relay that activate the DC motor.

If the temperature exceeds the setpoint, indicating a need for cooling, the ESP32 activates the DC motor, which powers the air conditioning system. The motor runs at a fixed speed suitable for the system's cooling capacity. The air conditioning system's compressor and other components start operating, absorbing heat from the indoor air and releasing it outside, thus providing cooling.

Once the temperature falls below the setpoint or reaches the desired range, the ESP32 switches off the Relay, thereby turning off the air conditioning system. This prevents overcooling and helps conserve energy.

The DHT11 sensor continues to provide real-time temperature and humidity readings to the ESP32, allowing the system to monitor the environment. If the temperature rises above the setpoint again, the ESP32 will activate the Relay, restarting the cooling process.

```
void sensor_DHT()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); //


  if(t > 30){

    digitalWrite(D2,HIGH);

  }


}
```

Figure 34:DHT11 code.

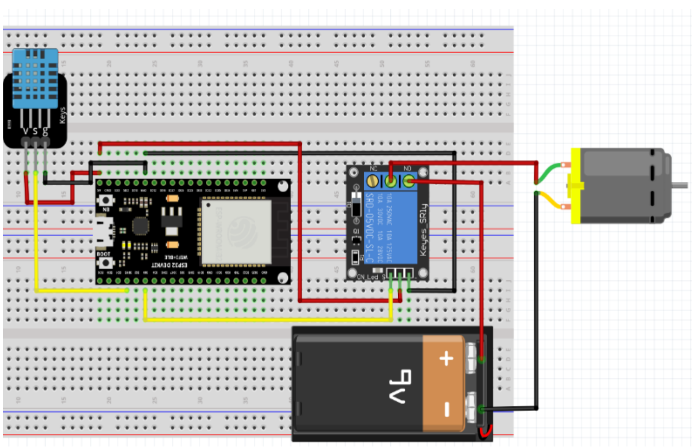


Figure 35:DHT11 circuit.

4.Outdoor lamp automation:

4.1Controling relay using LDR sensor values:

Outdoor lamp automation using an LDR sensor and an ESP32 microcontroller provides a convenient and energy-efficient solution. The LDR (Light Dependent Resistor) sensor detects ambient light levels and sends corresponding values to the ESP32. The ESP32, a powerful and versatile microcontroller, receives these values and processes them to control a relay connected to the outdoor lamp.

```
void outdoorlights(){
   LDR_Val = analogRead(sensor);   /*Analog read LDR value*/
    Serial.print("LDR Output Value: ");
    Serial.println(LDR_Val);    /*Display LDR Output Val on serial monitor*/
    if(LDR_Val > 2500) {       /*If light intensity is HIGH*/
        Serial.println(" High intensity ");
        digitalWrite(led,LOW); /*LED Remains OFF*/
    }
    else {
     /*Else if Light intensity is LOW LED will Remain ON*/
        Serial.println("LOW Intensity ");
        digitalWrite(led,HIGH);  /* LED Turn ON LDR value is less than 100*/
    }
    delay(2000);
}
```

Figure 37:outdoor lamp code.

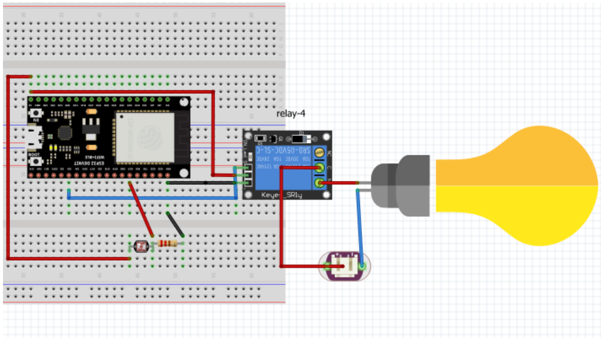


Figure 36:outdoor circuit.

5. Safety system:

5.1 Gas Sensor:

In this part we will realize a gas leak detection system controlled by the ESP32 board connected to the Internet. It mainly uses MQ-4 gas sensor, when the sensor detects gas leak, it will show us the values of Gaz sensor in serial monitor, if it is high than 1000 the buzzer tone and the window open by 90 degrees.

```
  //...... Gas sensor....
    int analogSensor = analogRead(smokeA0);

Serial.print("Pin A0: ");
Serial.println(analogSensor);
// Checks if it has reached the threshold value
if (analogSensor > sensorThres)
{
  tone(buzzer, 1000, 200);
  myservo2.write(90);
}
delay(100);
```

Figure 39:gaz sensor code.

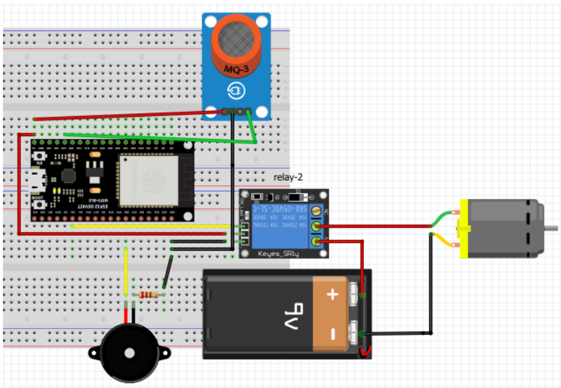


Figure 38:gaz sensor circuit.

5.2 Flam Sensor:

the combination of a flame sensor and ESP32 microcontroller provides a robust and intelligent fire detection solution. By promptly detecting flames and turn on buzzer, this setup enhances safety and enables proactive fire prevention measures.

```
void  flamedetected(){
flame_detected = digitalRead(flame_sensor);

  if (flame_detected == 1)

  {

    Serial.println("Flame detected...! take action immediately.");

    digitalWrite(buzzer, HIGH);

    delay(200);

  }

  else

  {

    Serial.println("No flame detected. stay cool");

    digitalWrite(buzzer, LOW);

  }

  delay(1000);
}
```

Figure 41:flame sensor code.

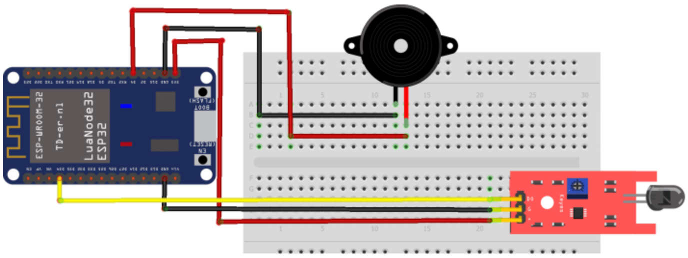


Figure 40:flame sensor circuit.

6. Efficient water management:

The water level sensor will measure the level of water then send the data to ESP32 which will make certain procedure. When the water level reaches a specified threshold, the Led will be on. This way, the Led will indicate when the water level reaches the desired level.

```
// ........water level .......
  digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
delay(10);                        // wait 10 milliseconds
value = analogRead(SIGNAL_PIN); // read the analog value from sensor
digitalWrite(POWER_PIN, LOW);// turn the sensor OFF
      Serial.print("water value is :");
  Serial.print(value);
  if (value > THRESHOLD) {
  Serial.print("The water is detected");
  digitalWrite(LED_PIN, LOW);  // turn LED ON
} else {
  digitalWrite(LED_PIN, HIGH);   // turn LED OFF
 }
}
```

Figure 43:efficient water management code.

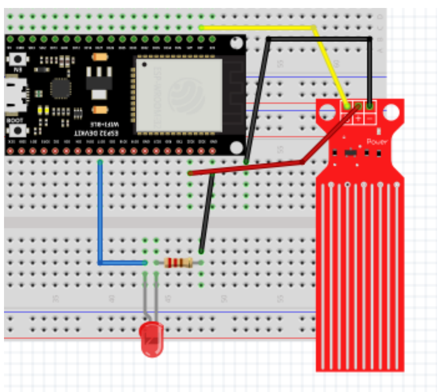


Figure 42:efficient water management circuit.

7. IOT cloud Interfacing:

Know we show the process of connecting and controlling various household appliances using the ESP32 microcontroller board and the power of the Internet of Things (IoT). By integrating ESP32 boards with the Arduino IoT Cloud, it becomes possible to remotely monitor and control appliances such as lights, fans, doors, and sensors through a cloud-based interface. The ESP32's built-in Wi-Fi capabilities allow for seamless connectivity to the internet, enabling users to conveniently interact with their appliances from anywhere using a computer, smartphone, or other internet-connected devices. By leveraging the capabilities of the ESP32 and the Arduino IoT Cloud, users can automate their homes, improve energy efficiency, and enhance overall convenience and control over their appliances.

Figure 44:arduino iot cloud dashboard.

7.1 Controlling room lamp:

Controlling a room lamp using the Arduino IoT Cloud involves utilizing an ESP32 microcontroller board with built-in Wi-Fi capabilities. The ESP32 is connected to the internet and linked to the Arduino IoT Cloud platform. Through the cloud platform, a user interface is created to control the lamp, providing options to turn it on or off. When a command is issued from the cloud platform, it is received by the ESP32, which processes the command and controls an LED connected to one of its digital pins. To simulate the lamp, the LED is turned on or off based on the command received. The ESP32 also provides feedback to the cloud platform regarding the lamp's status, allowing the user interface on the dashboard to reflect the real-time state of the lamp. By leveraging the ESP32 and Arduino IoT Cloud, remote control and monitoring of the room lamp using an LED are achieved seamlessly and conveniently.

```
/*
  Since Relay2 is READ_WRITE variable, onRelay2Change() is
  executed every time a new value is received from IoT Cloud.
*/
void onRelay2Change()  {
    if (relay2)
  {
    digitalWrite(D2,HIGH);
  }
  else
  {
    digitalWrite(D2,LOW);
  }

  // Add your code here to act upon Relay2 change
}
```

Figure 45:room lamp code.

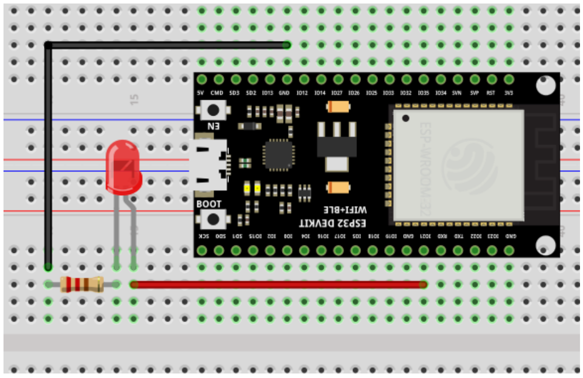


Figure 46:room lamp circuit.

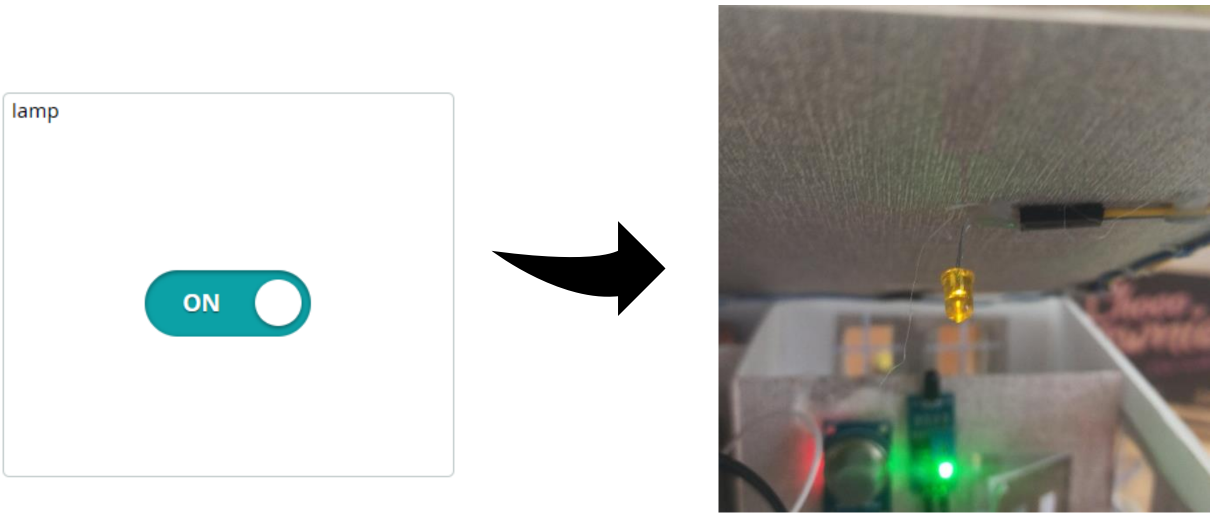


Figure 47:room lamp working test.

7.2 Controlling the fan:

Controlling a fan using a relay, ESP32, and Arduino IoT Cloud involves connecting the ESP32 microcontroller board to a relay module, which serves as an electronic switch for controlling the fan's power supply. The ESP32 is then connected to the Arduino IoT Cloud platform, creating a seamless link between the board and the cloud. Through the cloud platform, a user interface is designed on the dashboard to control the fan, providing options to turn it on or off. When a command is issued from the cloud platform, it is transmitted to the ESP32 via the internet. The ESP32 processes the command and controls the relay accordingly, either closing the circuit to activate the DC motor or opening the circuit to deactivate it. The ESP32 also provides feedback to the cloud platform, indicating the fan's current state. This feedback updates the user interface on the dashboard, allowing users to monitor and control the fan remotely using the Arduino IoT Cloud.

```
void onRelay1Change()  {
  if (relay1)
  {
    digitalWrite(D0,HIGH);

  }
  else
  {
    digitalWrite(D0,LOW);
  }

  // Add your code here to act upon Relay1 change
}
```

Figure 49:fan code.

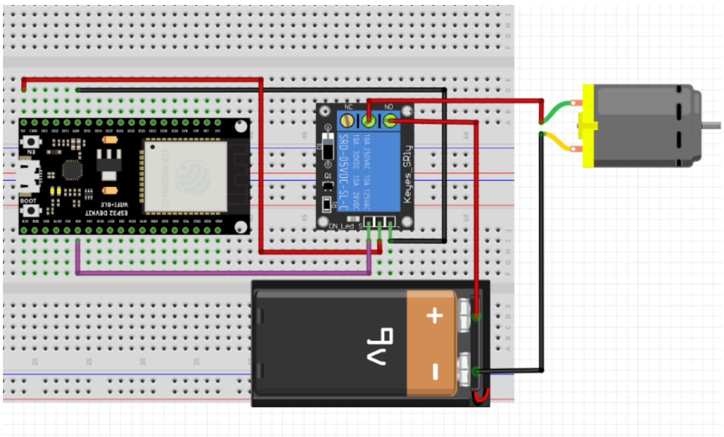


Figure 48:fan circuit.

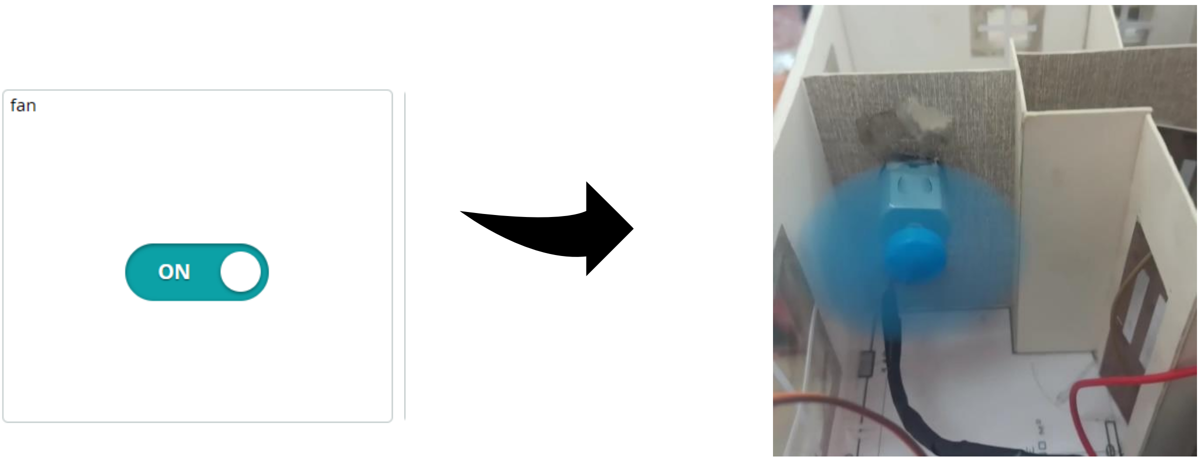


Figure 50:fan working test.

7.3 Controlling the window:

Controlling a window using a servo motor, ESP32 microcontroller, and Arduino IoT Cloud involves connecting the ESP32 board to a servo motor that controls the window's opening and closing mechanism. The ESP32 is then linked to the Arduino IoT Cloud platform, establishing a seamless connection between the board and the cloud. A user interface is created on the Arduino IoT Cloud dashboard, providing controls to set the desired angle for the servo motor. When a command is issued from the cloud platform, it is transmitted to the ESP32 via the internet. The ESP32 receives and processes the command, adjusting the servo motor to the specified angle, which in turn opens or closes the window to the desired position. Feedback regarding the window's current angle is sent back to the cloud platform, allowing the user interface on the dashboard to reflect the real-time status of the window. By leveraging the servo motor, ESP32, and Arduino IoT Cloud, remote control and monitoring of the window with precise angle control are achieved, providing convenience and flexibility for window management.

Figure 52:window code.

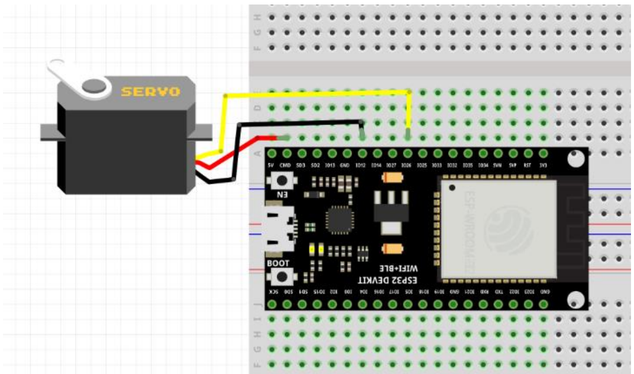


Figure 51:window circuit.

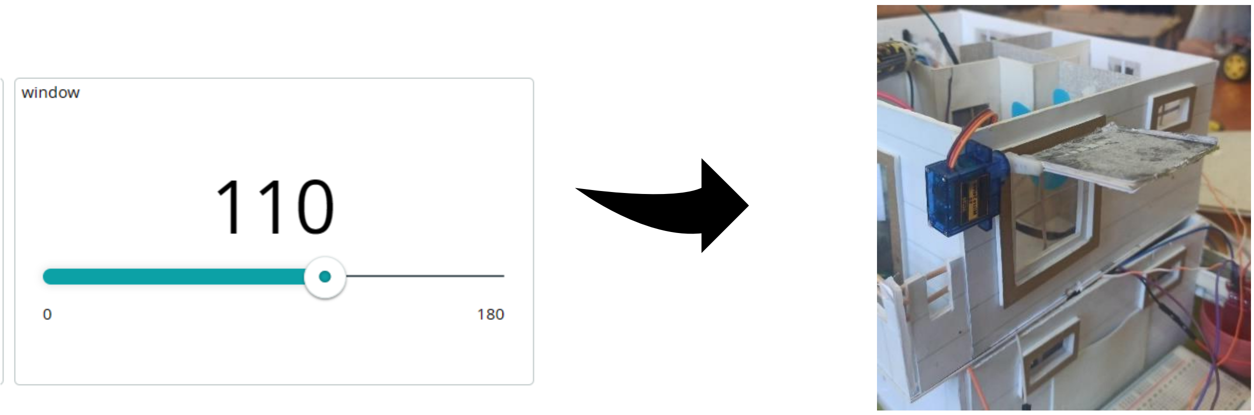
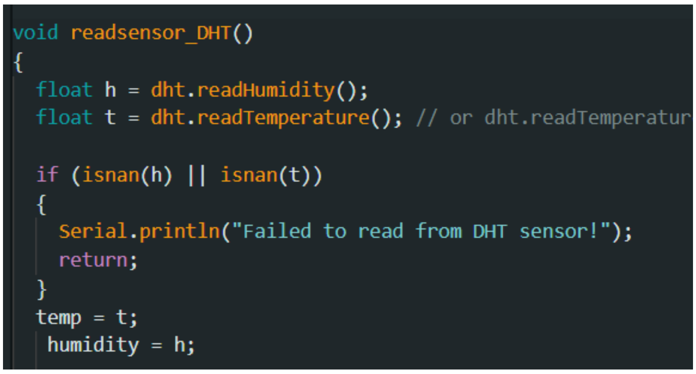


Figure 53:window working test.

7.4 Sending DHT sensor data from ESP32 to Arduino Cloud:

To send DHT sensor data from an ESP32 to the Arduino IoT Cloud, the ESP32 microcontroller board is connected to a DHT sensor, which measures temperature and humidity. After setting up an account on the Arduino IoT Cloud platform and creating a new project, the ESP32 is linked to the cloud platform using the Arduino IoT Cloud library and appropriate credentials. The ESP32 is programmed to periodically read data from the DHT sensor and establish a connection with the cloud platform. The temperature and humidity values are formatted, typically as JSON, and transmitted to the Arduino IoT Cloud. On the cloud platform, a visualization is created on the dashboard to display the received sensor data, such as temperature and humidity graphs or numerical values. The ESP32 continues to update the cloud platform with the latest sensor data, ensuring real-time monitoring of temperature and humidity. This setup allows for remote access and monitoring of DHT sensor data from the ESP32 using the Arduino IoT Cloud, enabling users to keep track of environmental conditions conveniently.



```
void readsensor_DHT()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperatur

  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  temp = t;
   humidity = h;
```

Figure 56:DHT11 code

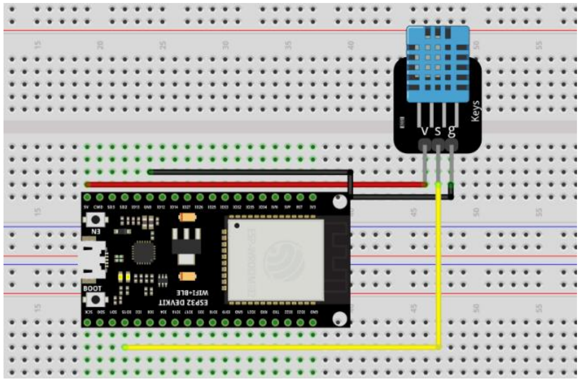


Figure 55:DHT11 circuit.

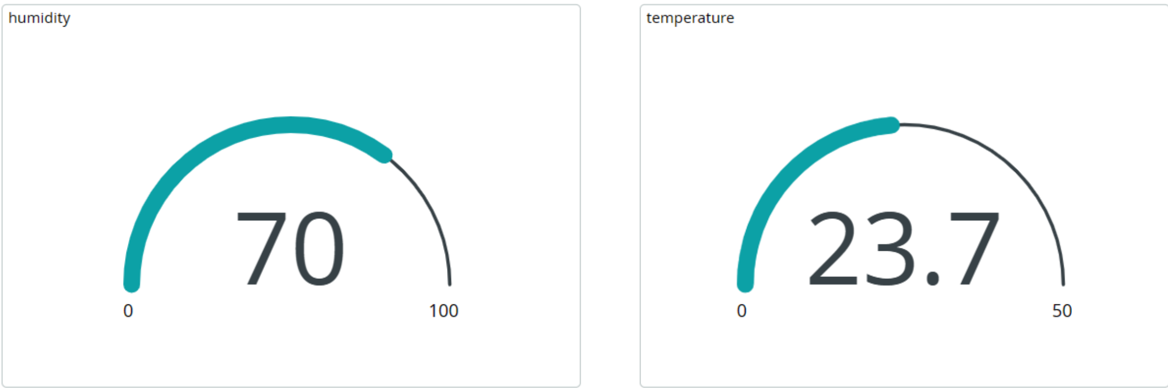
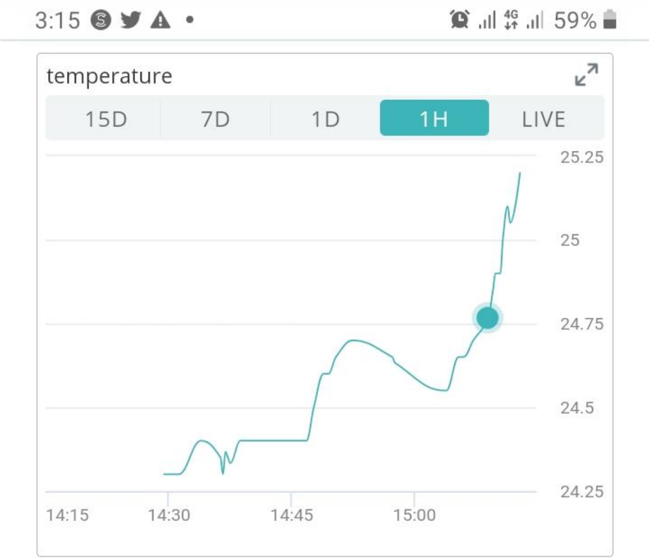
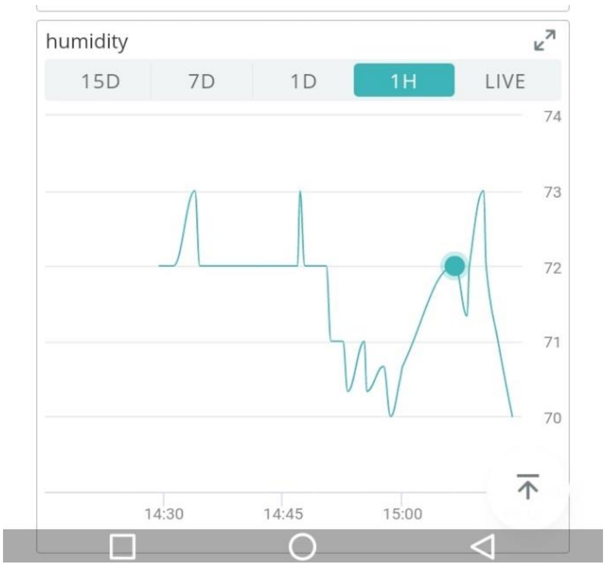
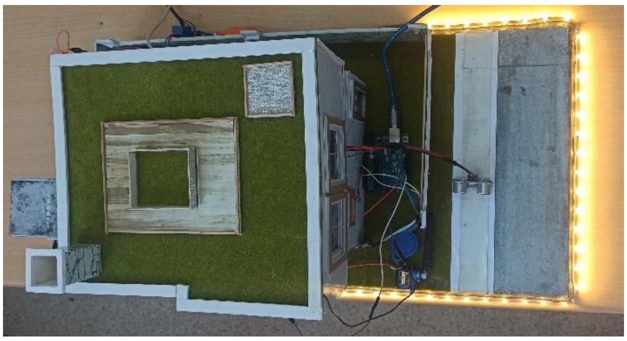


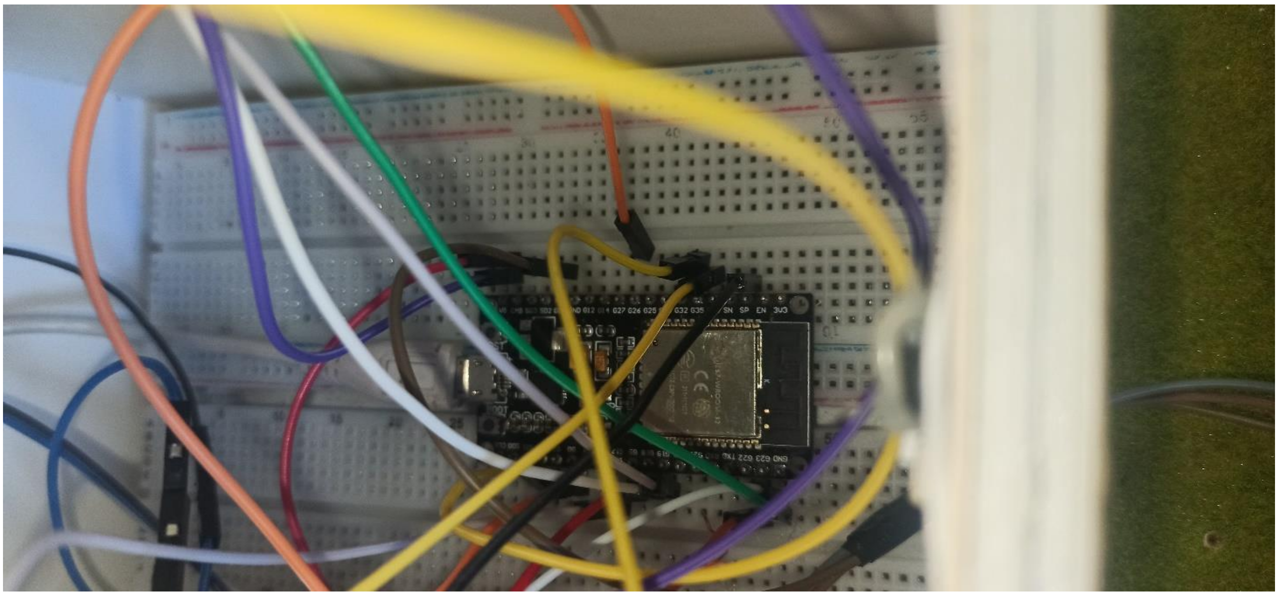
Figure 54:temperature and humidity level.

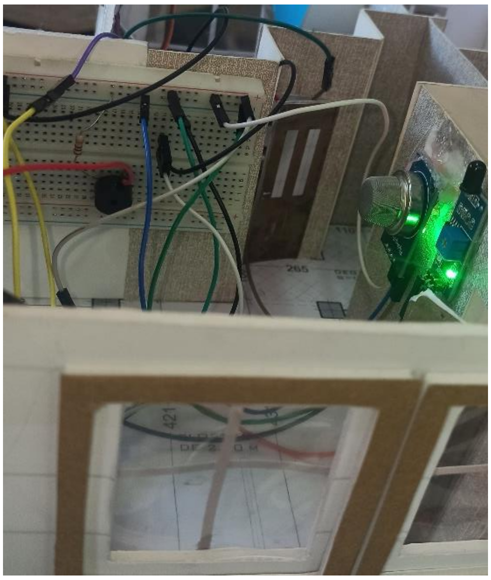


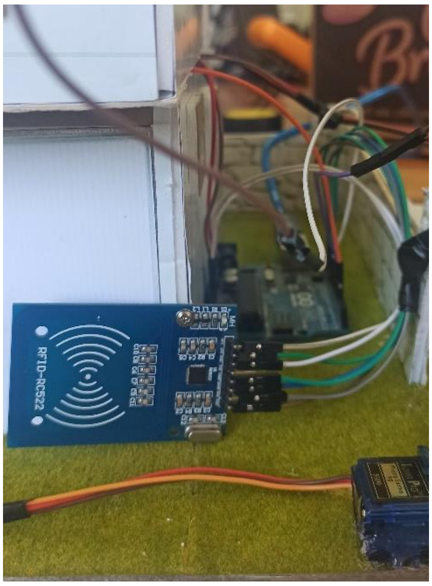


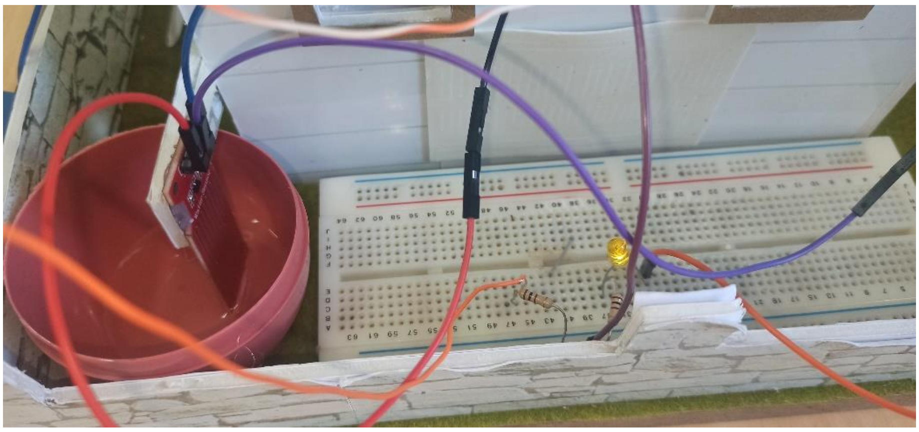
8.Project Final implementation:











Conclusion:

In conclusion, the project focused on implementing smart automation using IoT technologies, specifically utilizing the ESP32 microcontroller, Arduino IoT Cloud, and various sensors. The project covered several areas of automation, including garage automation, smart locks with keyless entry, air conditioning system control, outdoor lamp automation, safety systems, efficient water management, and IoT cloud interfacing.

The implementation of ultrasonic sensors and servo motor for smart garage automation allowed for secure access control and convenient opening and closing of the garage door. The implementation of RFID and servomotor in the smart locks enabled keyless entry and enhanced security measures.

The air conditioning system was enhanced by utilizing the DHT11 sensor data to control the DC motor, providing efficient cooling and temperature regulation based on real-time environmental conditions.

Outdoor lamp automation was achieved through the integration of LDR sensors and relays, allowing for automatic control of the lamps based on the ambient light levels, resulting in energy savings and increased convenience.

The safety system was strengthened by incorporating gas and flame sensors, enabling the detection and alerting of potential hazards in the environment, ensuring the safety of the occupants.

Efficient water management was addressed, although further details were not provided in the titles.

The project showcased the power of IoT cloud interfacing using the Arduino IoT Cloud. It demonstrated the control of various devices, such as room lamps, fans, and windows, through a cloud-based interface, enabling remote access and monitoring from anywhere. Additionally, the project highlighted the transmission of DHT sensor data from the ESP32 to the Arduino Cloud, facilitating real-time monitoring and analysis of temperature and humidity levels.

In summary, the project successfully implemented smart automation using IoT technologies, providing enhanced convenience, energy efficiency, safety, and remote-control capabilities. The combination of the ESP32, Arduino IoT Cloud, and sensor integration showcased the potential of IoT in creating a connected and automated environment

References

[1] M. McCullough, in *Digital Ground: Architecture, Pervasive Computing, and Environmental Knowing*, MIT Press, 2004.

[2] R. DeMara, in *Home Automation Basics: Practical Applications Using Visual Basic 6.0. Prentice Hall PTR.*, (2003).

[3] M. &. H. R. Hauben, in *Netizens: On the History and Impact of Usenet and the Internet.* , IEEE Computer Society Press, (1997).

[4] [Online]. Available: https://www.helpwire.app/blog/iot-smart-home/.

[5] [Online]. Available: https://www.smarthomenews.info/benefits-and-challenges-of-smart-homes/.

[6] [Online]. Available: https://www.arduino.cc/pro/platform-overview.

[7] [Online]. Available: https://docs.arduino.cc/arduino-cloud/getting-started/iot-cloud-getting-started.

[8] [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[9] [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf.

[10] [Online]. Available: https://www.arduino.cc/en/Guide/ArduinoSensors.