Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

A Smart Home Design Based On Arduino

Novita Delima Siahaan[1], Beni Satria[2]
[1,2]Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia

ArticleInfo

Keywords:
smart home,
arduino,
sensors,
security

ABSTRACT

An Arduino-based smart home is the implementation of a smart system that controls various devices in the house such as lights, fans, doors, and others to increase comfort, security, and energy efficiency. Arduino acts as the brain of this system, receiving input from sensors and providing output to the controlled devices. The system can be accessed and controlled remotely via the internet with the Blynk app, allowing users to control their home even when they are not at home. In this design, sensors such as gas sensors, temperature sensors and humidity sensors are used to detect environmental conditions and occupant activities. Arduino uses information from these sensors to make automatic decisions, such as turning on the lights when it detects movement at night or sounding an alarm when there is an LPG gas leak, adjusting the room temperature as needed. Security is also a main focus in this design. Door and window sensors are installed to detect intrusions, and notifications can be sent to users via the Blynk app if there is suspicious activity. By using Arduino as the main platform, this smart home design can be easily customized and developed further according to user needs. From the research results it can be seen that the system has functioned as it should, in accordance with the initial design

Corresponding Author:
Novita Delima Siahaan
Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia
siahaannovitadelima@gmail.com

INTRODUCTION

The Technological advances in recent decades have led to the development of the smart home concept which is increasingly popular. Smart home allows users to control and monitor various electronic devices at home automatically or remotely via the internet. This cannot be separated from the development of more and more Arduino development boards on the market. Arduino, as an affordable and easy-to-use hardware development platform, offers great potential in designing efficient and affordable smart home systems. By using Arduino as a base, users can create various smart home systems that suit their needs, from controlling lights, temperature, security, to energy monitoring.

This research aims to design and implement an Arduino-based smart home system that can control and monitor various electronic devices in the home, as well as detect LPG gas leaks. By utilizing a gas sensor to detect LPG gas leaks and a DHT11 sensor to measure temperature and humidity, this system is expected to provide convenience, comfort and security for home residents. LPG gas leaks are a serious threat to home safety. This gas has no smell or color, making it difficult to detect without the help of a device. With a gas sensor integrated in the smart home system, home residents can immediately detect a gas leak and

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

take the necessary precautions, such as turning off the gas supply and notifying the authorities.

Apart from that, using the DHT11 sensor to measure temperature and humidity also provides great benefits in managing the environment in the home. Information about temperature and humidity can help in regulating the use of air conditioning, the use of heating devices, and air ventilation, thereby increasing occupant comfort and saving energy. By combining gas sensors and temperature/humidity sensors in an Arduino-based smart home system, it is hoped that it can create a safer, more comfortable and efficient environment for home residents.

Literature Review

Smart Home

Smart home, or smart home, is a concept in which a home is equipped with various electronic systems and devices that are connected and automated to improve comfort, security, energy efficiency and overall home management. Smart homes use information and communications technologies (ICT), such as sensors, automatic controls, and computer networks, to provide smarter, more integrated functionality in everyday life.

Smart homes allow users to control and monitor electronic devices, such as lights, air conditioners, security devices, door locks, cameras, and other household devices, via mobile apps or other internet-connected devices. By using a smart home, users can perform various actions, such as regulating room temperature, controlling lighting, checking home security, and managing energy more efficiently.

Apart from that, smart homes can also be equipped with a smart energy management system that allows users to optimize energy use at home, such as setting a schedule for using heavy equipment or using energy from renewable sources. Smart homes can also improve accessibility for people with special needs, such as smart systems that can be operated by voice or gestures. With the continued development of technology, the smart home concept continues to evolve and offers new possibilities to improve the quality of life, energy efficiency and security at home.

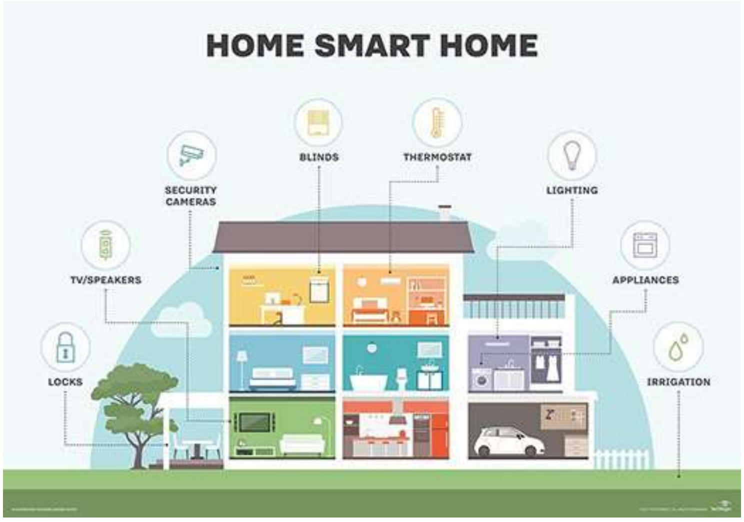

Figure 1. Smart Home illustration

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

Arduino

Arduino is a popular hardware and software platform for developing electronic prototypes. This platform was first developed in Italy in 2005 by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. Arduino was created to provide an affordable and easy-to-use tool for developers, hobbyists, and electronics beginners.

The definition of Arduino Nano is a small development circuit board that includes a microcontroller and supports the use of a breadboard. The Arduino Nano was specifically designed and produced by the Gravitech company using the Atmega328 (for Arduino Nano V3) or Atmega168 (for Arduino Nano V2) microcontroller base.

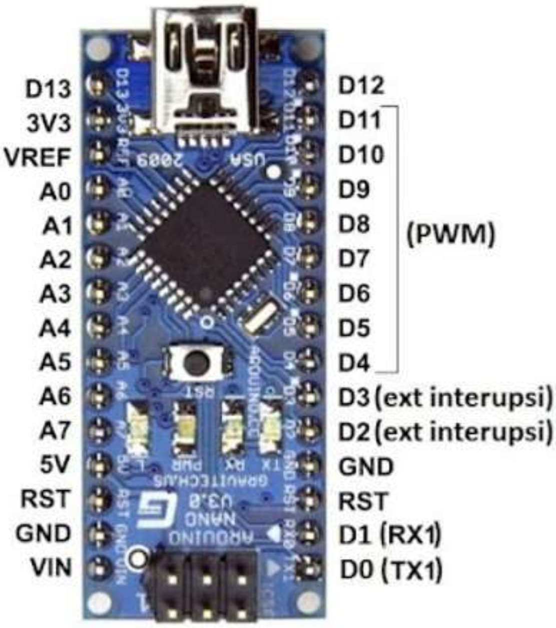

Figure 2. Arduino Nano pinout

You can see a summary of the Arduino Nano specifications in the table below:

Table.1 Arduino Nano Specifications

Microcontroller Type	Atmega328
Operating Voltage	5 Volts
Recommended Voltage	7 - 12 Volts
Voltage Limit	6 - 20 volts
Digital Input/Output Pins	14
PWM pins	6
Analog Input Pins	8
Current Per Pin	40 mA
Flash Memory	32 KB (2 KB for bootloader)
SRAM	2KB
EEPROM	1KB
Clock Speed	16 MHz
Long	4.3 cm
Wide	1.8 cm
Heavy	5 grams

NodeMCU ESP01

ESP-01 is a WiFi module that allows Microcontrollers to easily access WiFi networks. It is one of the mainly incorporated WiFi chips in the industry, which assimilates antenna switch, radio frequency balun, power amplifier, low noise receiver amplifier and power executive elements.

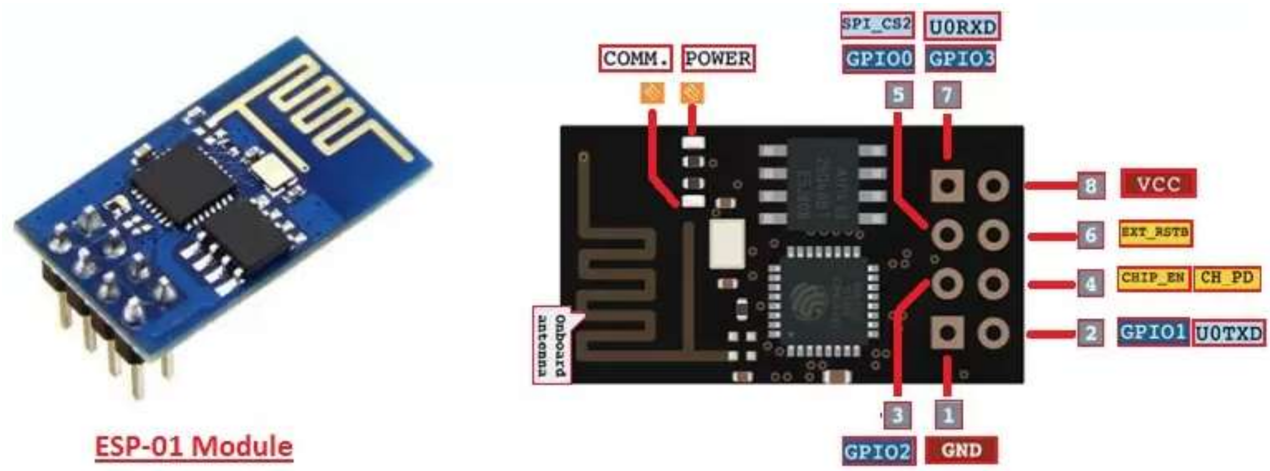

Figure 3. ESP01 Module

a. This module requires minimum internal circuitry, the entire solution, including the front end module is designed to occupy minimum PCB area.
b. The ESP-01 module is called a system on chip (SOC) because it acts as a stand-alone Microcontroller, so we don't need to connect it with other microcontrollers (i.e. Arduino, Atmel, PIC Microcontroller, etc.) sequentially to use the I/O Pins- his.
c. The ESP-01 also integrates an advanced version of SRAM L-106 Tensilica diamond series with WiFi function. It also integrates with certain devices via GPIO and the code for the application is provided in the SDK.

The ESP-01 pinout is explained below with detailed explanation.

Table. 2 ESP01 Pin Functions

Pin#	Type	Parameter
Pin #1	VCC	This pin is used for 3.0 to 3.6V power supply input.
Pin #2	GND	This pin is used for Ground.
Pin#3	RESET	This pin is used for external reset signal (Low voltage level: On).
Pin#4	ADC(OUT)	This pin is an analog to digital converter.
Pin#5	CH_PD	This is Chip Activation. High: On, chip is working properly; Low: Off, small current.
Pin#6	GPIO0 (FLASH)	This is general purpose I/O, If low on reset/power on brings the chip into serial programming mode.
Pin#7	GPIO1(TX)	These are general purpose I/O and Serial TXd.
Pin#8	GPIO3(RX)	These are general purpose I/O and Serial RXd.
Pin#9	GPIO4	This is general purpose I/O.

MQ-2 sensors

The MQ-2 sensor is a sensor that is sensitive to cigarette smoke. The main material of this sensor is SnO2 with low conductivity in clean air. If there is a gas leak, the sensor conductivity becomes higher, with every increase in gas concentration, the sensor

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

conductivity also increases. The MQ-2 sensor is sensitive to LPG, Propane, Hydrogen, Carbon Monoxide, Methane and Alcohol gases as well as other flammable gases in the air.

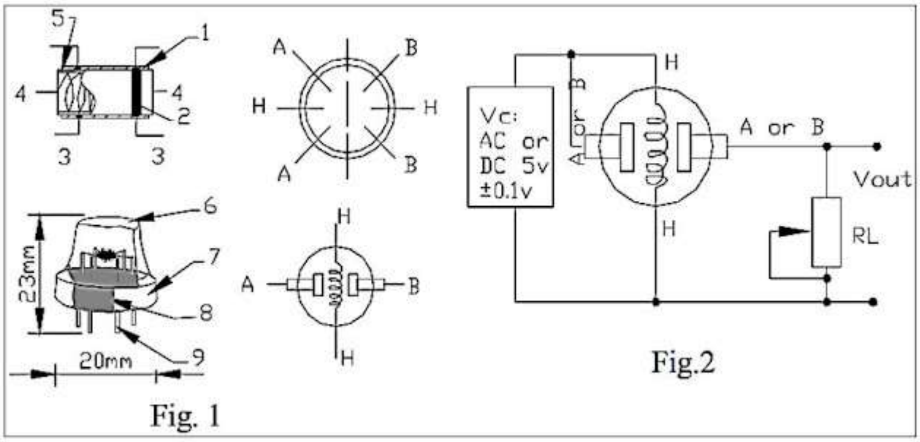

Figure 4. MQ-2 Sensor Structure

Figure 5. MQ-2 Sensor Module

MQ2 consists of 4 pinout legs, including;

1. VCC as a power supply to the module, get connected to the 5V pin of your microcontroller.
2. GND is the ground leg.
3. DO indicates the presence of flammable gas. D0 will be HIGH when the gas concentration does not exceed the threshold, and vice versa.
4. A0 shows the relationship between gas concentration and output voltage. When a higher concentration produces a high voltage, vice versa, a low concentration will produce a low voltage too.

MQ-2 Sensor Characteristics The MQ-2 sensor has 2 voltage inputs, namely VH and VC. VH is used for the voltage on the internal heater and Vc is the source voltage. The power supply required for the MQ-2 sensor is Vc < 24VDC and VH = 5V ± 0.2V AC or DC voltage. This gas and smoke sensor detects the concentration of flammable gases in the air as well as smoke and the output reads as an analog voltage. The sensor can measure flammable gas concentrations from 300 to 10,000 ppm. It can operate at temperatures from -20 to 50° C and consumes less than 150 mA at 5V. The sensitivity of the MQ-2 Sensor characteristics can be seen in Table 2.

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

How to Calibrate Sensors

The MQ2 principle works in hot conditions, the sensor calibration can deviate if left or stored for a long time. Therefore, it is necessary to pay attention to how to calibrate so that the sensor output can be maximized.

1. When first used after a long period of storage (a month or more), the sensor needs to be warmed up for 1 day to 2 days for maximum accuracy.
2. The sensor was just used, then left. It only needs to be warmed for 5-10 minutes. During the warm-up period, the sensor will initially read high (high ppm) and gradually decrease until the sensor stabilizes.

Table 3. MQ-2 Sensor Sensitivity Characteristics

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	3K Ω – 30KΩ (1000ppm iso – butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H_2 100ppm-2000ppm Alcohol
σ (3000/1000) isobutane	Consentration Slope rate	≤0.6	
Standart Detecting Condition	Temp: 20°C ± 2°C Vc: 5V ± 0.1 Humidity: 65% ± 5% Vh: 5V ± 0.1		
Preheat time	Over 24 hour		

DHT-11 Temperature-Humidity Sensor

DHT 11 is a sensor that is capable of detecting temperature and humidity in the area around the sensor. This sensor consists of a thermistor to check temperature and a capacitive sensor to check humidity. Usually this sensor is packaged directly with the module so that the module already contains a sensor and chip to convert analog voltage into digital signals. As in the picture above, the module consists of 3 pins, namely:

a. VCC (+): This is the pin for input voltage into the module.
b. Gnd (-): This is the pin for inputting ground or zero into the module.
c. Out: Is a pin for flowing digital signals into the circuit/microcontroller.

DHT 11 Sensor Specifications

Table. 4 Specifications DHT 11 Sensor

Input Voltage	3 Volts
Current 0.3mA	Idle 60uA
Sampling period	2 seconds
Data output	Serial
Resolution	16 bits
Temperature between 0°C to 50°C	(2°C accuracy)
Humidity between 20% to 90%	(5% accuracy)

The shape of the DHT 11 sensor is as in the picture below.

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

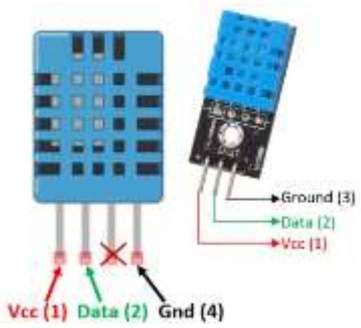

Figure 6. SHT 11 Temperature/Humidity Sensor

The DHT11 sensor is factory calibrated and outputs serial data so it is very easy to set it up. The connection diagram for this sensor is shown below.

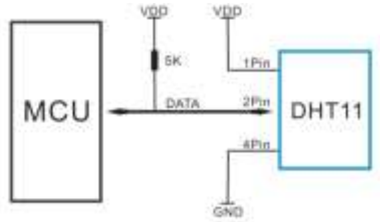

Figure 7. DHT 11 Sensor Connection and MCU

From the picture above it can be seen, the data pin is connected to the MCU I/O pin and a 5K pull-up resistor is used. This data pin outputs temperature and humidity values as serial data. If you want to connect a DHT11 to an Arduino then there is a ready to use library that will give you a quick start.

METHOD

The research process lasted approximately 4 months, starting from January to April 2024, starting from title consultation to the thesis preparation process. Each step in the research must be explained in order to gain knowledge to solve a problem that will be faced in this research. With scientific, systematic and logical steps at every stage of the research. To make it easier in the research stages, a flow chart is needed as a flow in making hypotheses in a study. The flow chart of the research stages is as in Figure 8 below.

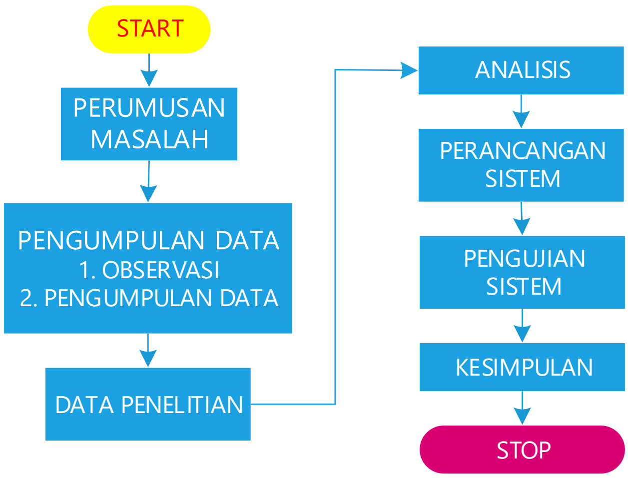

Figure 8. Flow chart of research stages

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

The explanation of the image above is as follows:

a. Early stage

   The first step in carrying out this research is by conducting a literature study to obtain information regarding the classification of a problem that occurs regarding Home Security Systems, then carrying out data collection regarding the complexity of the problem, after obtaining the data, the next stage is to carry out an analysis.

b. Analysis Stage

   The next step is to enter the analysis stage process, stage This analysis process will be carried out to determine the method research in solving future problems completed, from the results of data that has been accumulated so that it can be done designing a system or tool that will start from hardware and software design up to the design stage products that will be created and developed, once completed The next process designed is to carry out testing on a system or tool that has been created, so that you can find out level of success in creating the system or tool.

c. Conclusion

   After carrying out the analysis process to the testing stage, The next step is to draw a conclusion from the results tests that have been carried out so that they can be completed problems resulting from observations that have been made.

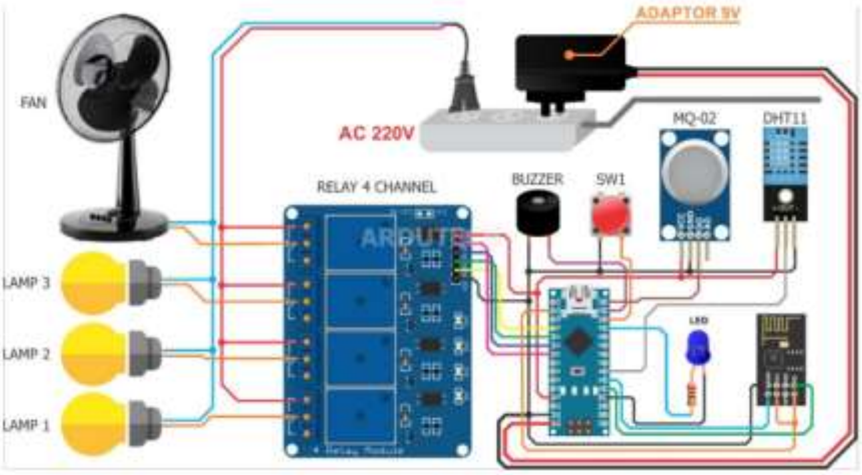

Figure 9. Complete circuit image

RESULTS AND DISCUSSION

Hardware Design Results

The smart home hardware design consists of two modules that are designed separately, making it more flexible in later installation. This smart home works in accordance with the initial design objectives, as in Figure 11 below.

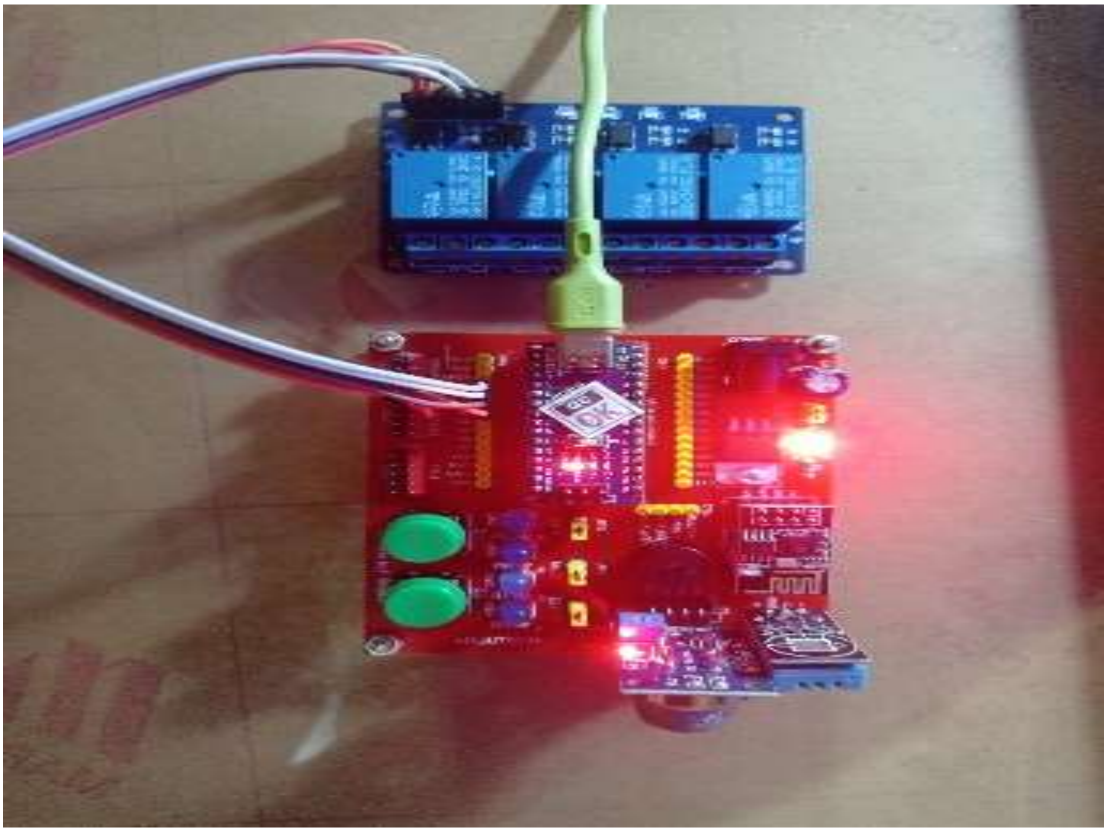

Figure 10. Smart home module testing stage

In terms of modeling, there are several devices made with Arduino Nano and ESP01 with the following system:

Table 5. The Component of System

Feature	Component
Gas Leak Notification	MQ2 sensors
Thief Notification	Sensor With Switch
Headlight Control	Relay 1
Center Light Control	Relay 2
Taillight Control	Relay 3
Fan Control	Relay 4
Night Light Brightness	LED 4
Doorbell	Buzzers
Temperature & Humidity Monitoring	DHT11

Table 6. Arduino Nano Connection With ESP-01 Module

Arduino Nano V3	ESP-01 Module
D2	RX
D3	TX
3.3 V	VCC
GND	GND

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

Table 7. Arduino Nano Connection with DHT11 Sensor

Arduino Nano V3	DHT11
D4	DATA
5V	+
GND	-

Table 8. Arduino Nano Connection With MQ2 Sensor

Arduino Nano V3	MQ2
D12	D0
5V	VCC
GND	GND

The next step is to install the Blynk application on your smart phone. Fill the token obtained from the Blynk server into the source code. Do the settings according to the instructions on the Blynk website. After success, the serial monitor can be seen as in Figure 12.

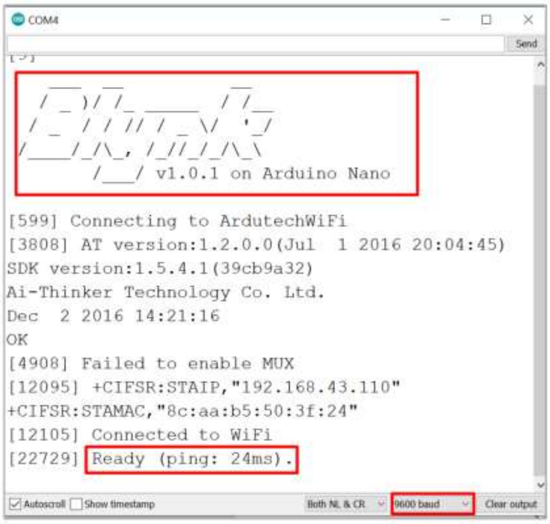

Figure 11.. Blynk display on the serial monitor

It can also be seen on the Dashboard (Blynk web) that was created earlier. So the results are as in Figure 13.

Figure 12. Display of the Blynk Dashboard menu

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

You can see that the Smart Home device is online. If it's not online then check the ID token, or WiFi.

System Testing

The next stage is testing how the tool works. At this stage, each function of each sensor and switch used will be tested.

Steps for testing sensors with software

1. DHT11 Sensor Testing:

- Create a simple sketch to read data from the DHT11 sensor.
- Upload the sketch to the NodeMCU and open the Serial Monitor to ensure that the sensors are reading temperature and humidity correctly.

```
#include "DHT.h"
#define DHTPIN 2 // The pin to which the DHT11 sensor is connected
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
 Serial.begin(115200);
 dht.begin();
}

void loop() {
 float h = dht.readHumidity();
 float t = dht.readTemperature();
 Serial.print("Humidity: ");
 Serial.print(h);
 Serial.print("% Temperature: ");
 Serial.print(t);
 Serial.println("C");
 delay(2000);
}
```

.MQ-02 Sensor Testing:

- Create a simple sketch to read data from the MQ-02 sensor.
- Upload the sketch and check the values read by the sensor to ensure the sensor is working properly.

```
#define MQ2_PIN A0

void setup() {
 Serial.begin(115200);
}

void loop() {
 int sensorValue = analogRead(MQ2_PIN);
```

```
 Serial.print("MQ-02 Sensor Value: ");
 Serial.println(sensorValue);
 delay(2000);
}
```

Testing Relays and Actuators (Lights and Fans):

Make a sketch to control the relay and ensure that the relay can turn the lights and fan on and off on command.

```
#define RELAY1 5
#define RELAY2 4
#define RELAY3 0

void setup() {
 pinMode(RELAY1, OUTPUT);
 pinMode(RELAY2, OUTPUT);
 pinMode(RELAY3, OUTPUT);
}

void loop() {
 digitalWrite(RELAY1, HIGH);
 delay(1000);
 digitalWrite(RELAY1, LOW);
 delay(1000);

 digitalWrite(RELAY2, HIGH);
 delay(1000);
 digitalWrite(RELAY2, LOW);
 delay(1000);

 digitalWrite(RELAY3, HIGH);
 delay(1000);
 digitalWrite(RELAY3, LOW);
 delay(1000);
}
```

Buzzer and LED Testing:

Sketch the controls for the buzzer and LED, ensuring they can function as commanded.

```
#define BUZZER 12
#define LED 13

void setup() {
```

```
 pinMode(BUZZER, OUTPUT);
 pinMode(LED, OUTPUT);
}

void loop() {
 digitalWrite(BUZZER, HIGH);
 digitalWrite(LED, HIGH);
 delay(500);
 digitalWrite(BUZZER, LOW);
 digitalWrite(LED, LOW);
 delay(500);
}
```

Integrate all components in one sketch, ensuring they function together according to the desired logic. For example, when the MQ-02 sensor detects gas, the relay will turn on the fan and the buzzer will sound.

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

#define MQ2_PIN A0
#define RELAY1 5
#define RELAY2 4
#define RELAY3 0
#define BUZZER 12
#define LED 13

void setup() {
 Serial.begin(115200);
 dht.begin();
 pinMode(RELAY1, OUTPUT);
 pinMode(RELAY2, OUTPUT);
 pinMode(RELAY3, OUTPUT);
 pinMode(BUZZER, OUTPUT);
 pinMode(LED, OUTPUT);
}

void loop() {
 float h = dht.readHumidity();
 float t = dht.readTemperature();
 int mq2Value = analogRead(MQ2_PIN);
```

```
 Serial.print("Humidity: ");
 Serial.print(h);
 Serial.print("% Temperature: ");
 Serial.print(t);
 Serial.println("C");
 Serial.print("MQ-02 Sensor Value: ");
 Serial.println(mq2Value);

 if (mq2Value > threshold) { // Adjust the threshold value as needed
 digitalWrite(RELAY1, HIGH);
 digitalWrite(RELAY2, HIGH);
 digitalWrite(BUZZER, HIGH);
 digitalWrite(LED, HIGH);
 } else {
 digitalWrite(RELAY1, LOW);
 digitalWrite(RELAY2, LOW);
 digitalWrite(BUZZER, LOW);
 digitalWrite(LED, LOW);
 }
 delay(2000);
}
```

IoT Connection Testing:

- Implement code to connect NodeMCU to a Wi-Fi network and send sensor data to a server or IoT platform such as Blynk.

```
#include <ESP8266WiFi.h>
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

void setup() {
 Serial.begin(115200);
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED) {
 delay(1000);
 Serial.println("Connecting to WiFi...");
 }
 Serial.println("Connected to WiFi");
}

void loop() {
 // Code to send sensor data to IoT server
}
```

Jurnal Scientia
Volume 13 , Number 03, 2024, DOI 10.58471/ scientia.v13i03
ESSN 2723-7486 (Online)
https://infor.seaninstitute.org/index.php/pendidikan

The above tests cover every component in an IoT-based home security system and ensure that all parts function well individually and in an integrated system. Make sure to make adjustments and troubleshoot if any components do not function as expected.

After testing each sensor is carried out, functional testing is then carried out. The steps are as follows:

1. Preparation:
    - Make sure all components are installed according to the diagram
    - Verify power sources (9V adapter and 220V AC) are connected properly
2. Connectivity Testing:
    - Check the WiFi connection of the module (likely ESP8266) to the network
    - Verify the Arduino can communicate with the WiFi module
3. Sensor Test:
    - DHT11 (temperature and humidity sensor): Read temperature and humidity values, make sure they are accurate
    - MQ-02 (gas sensor): Test with gas leak simulation, detection verification
4. Actuator Testing:
    - 4 channel relay: Activate each channel, make sure it works
    - Fan: Turn on and off via relay
    - Lamps (LAMP 1, 2, 3): Control the on/off of each lamp
5. Test Notification:
    - Buzzer: Activate in various scenarios (gas detection, high temperature)
    - LED: Verify status indicator is working
6. User Interaction Testing:
    - Switch (SW1): Test the function to enable/disable the system
7. Security Scenario:
    - Gas leak simulation: Activate MQ-02 sensor, check response (buzzer, notification)
    - Fire simulation: Increase DHT11 temperature, verify alarms and actions (e.g. fan active)
8. Integration Test:
    - Verify all components work together in different scenarios
    - Make sure the data from the sensor triggers the appropriate action (eg: high temperature -> fan on)
9. Remote Testing:
    - Access the system via application or web interface
    - Control devices (lights, fans) remotely
    - Receive real-time notifications when an incident occurs
10. Reliability Test:
    - Run the system for long periods of time
    - Monitor the consistency of sensor readings and system response

The test results for each part can be seen on the smart phone screen as in the image below.

CONCLUSION

From the discussion and testing above, the following conclusions can be drawn: The IoT-based home security system that has been built and tested has several main components, including temperature and humidity sensors (DHT11), gas sensors (MQ-02), relay modules to control lights and fans, buzzers, LEDs, and Wi-Fi connectivity. via NodeMCU. This system is designed to provide better security and easy control over the internet network. Here are the overall conclusions about this system: Component Integration and Functionality: All components integrated in this system, such as the DHT11 and MQ-02 sensors, work well in detecting environmental changes such as temperature, humidity and the presence of harmful gases. The relay module successfully controls electrical devices (lights and fans) according to the conditions detected by the sensor. Buzzers and LEDs provide effective warning signals when dangerous conditions are detected, such as the presence of hazardous gases. Connectivity and Monitoring: NodeMCU successfully connects the system to a Wi-Fi network, allowing sensor data to be sent to a server or IoT platform such as Blynk or Firebase. This connectivity enables remote monitoring, so homeowners can receive notifications or monitor the condition of their home in real-time via an application or online dashboard. Reliability and Responsiveness: The system shows good reliability in detecting environmental changes and providing fast response through connected actuators (relays, buzzers, LEDs). Fast and accurate system responsiveness is key in emergency situations to prevent potential harm or further damage.

REFERENCES

[1] Abdul Hafidz Murtono Soehady, Siswanto, “Android Application for Monitoring and Controlling the VC0706 Camera in the Room with the Arduino Uno R3 Microcontroller, Pir SR510 Sensor and WIFI Shield CC3000 at PT. Indolakto”, SKANIKA VOLUME 1 NO. MARCH 1, 2018, pp. 270 – 276.
[2] Aryza, S., Lubis, Z., Indrawan, MI, Efendi, S., & Sihombing, P. (2021). Menganalisis Pemodelan Berbasis Data Desain Baru Sistem Pembangkit Listrik Piezoelektrik. Institut Penelitian dan Kritikus Internasional Budapest-Jurnal (BIRCI-Journal), 4(3), 5537-5547.
[3] Aryza, S., Wibowo, P., & Saputra, D. (2022, Juli). Rancang Bangun Alat Pengontrolan Proses Pemanasan Produksi Biodisel Dari Minyak Jelantah Berbasis Arduino Mega. Di dalam*Prosiding Seminar Nasional Sosial, Humaniora, dan Teknologi*(hlm. 121-127).
[4] Clifford MJ dan Eastwood D., “Desain pelacak surya pasif baru,” Sol.Energi, vol. 77, tidak. 3 Tahun 2004, hlm. 269–280.
[5] Dedi Setiawan, Joni Eka Candra, Cosmas Eko Suharyanto, "Designing a Home Security Control System with Smart CCTV Using Arduino Based on Telegram", INFOTEKJAR: National Journal of Informatics and Network Technology- Vol. 4 no. 1 (2019) September Edition, pp. 185 – 190.
[6] Central Bureau of Statistics, "Crime Statistics 2018," Central Bureau of Statistics, Jakarta, 2018.
[7] J. Waworundeng, LD Irawan, and CA Pangalila, "Implementation of PIR Sensors as Motion Detectors for Home Security Systems using IoT Platforms," Cogito Smart Journal, vol. 3, no. 2, pp. 152–163, 2017.

[8] Mubarok, I. Sofyan, AA Rismayadi, and I. Najiyah, "Home Security System Using RFID, PIR Sensor and Microcontroller Based GSM Module," Journal of Informatics, vol. 5, no. 1, pp. 137–144, 2018.
[9]. FP Juniawan, DY Sylfania, and EA Dika, "Archive Room Security System Prototype Using SMS Gateway Based Microcontroller," in National Conference on Information Systems, 2018, pp. 1042–1047.
[10] .M. Bimo, B. Setiyono, and A. Sofwan, "Designing a Door Lock System in a Smart Home Using an Atmega16 Microcontroller on the Android Platform," Transient, vol. 6, no. 4, pp. 575–581, 2017.
[11] .MR Asad, OD Nurhayati, and ED Widianto, "Automatic Home Door Security System Via SMS Based on ATMega328P Microcontroller," Journal of Computer Technology and Systems, vol. 3, no. 1, pp. 1–7, 2015.
[12] .D. Widcaksono and M. Masyhadi, "Design of a Secured Door Automatic System for Home Security Using Arduino-Based SMS," Journal of Electrical Engineering Studies, vol. 3, no. 1, pp. 52–66, 2018.
[13] .S. Ramadhan and LB Handoko, "Design of a Home Security System Based on Arduino Mega 2560," Techo.COM, vol. 15, no. 2, pp. 117–124, 2016.