KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

MTech in ARTIFICIAL INTELLIGENCE

PANCHKHAL, KAVRE

An Assignment

For

Simulating Humidity Sensors Using CISCO Packet Tracer

SUBMITTED TO

Prof. Sudan Jha

SUBMITTED BY

Prabesh Regmi
MTech AI 230008

August, 2024

Abstract

Technology is increasingly important in all aspects of modern life. One of these needs is to construct a smart home that can be controlled automatically or with various devices like smartphones, laptops, etc. The design and implementation of Internet of Things (IoT) systems require platforms with smart things and components. This implementation can be done effectively with package tracking software that includes IoT functionalities for controlling and simulating a smart house. IoT technology can be used for a variety of real-world applications, including homework, treatment, campus, and business environments. This study focuses on a humidity monitoring system that incorporates gadgets like humidity sensors, microcontrollers, humidifiers, and other different devices. The goal of this research is to create a simulation of a smart humidity control system that can be automatically controlled according to the humidity level in the environment and then to demonstrate the notion of a smart humidity control system. Use of Cisco Packet Tracer feature simulated smart humidity control system and Internet of Things devices being monitored. Simulation results show that smart objects can be connected and effectively monitored, pointing to the possibility of real-world deployment.

Keywords - IOT technology, Cisco Packet Tracer, Home gateway, IoT server, IoT monitor, Microcontroller, Personal computer

Introduction

The expression "Internet of Things" (IoT), coined back in 1999 by Kevin Ashton, the British technology pioneer who co-founded the Auto-ID Center at the Massachusetts Institute of Technology, is becoming more and more mainstream [1]. The Internet of Things (IoT) and Internet of Services (IoS) ideas are key components of the broader Industry 4.0 technology. These interconnected things and services allow modern smart factories and integrated value chains to run smoothly [2]**.** The Internet of Things (IoT) is the networked interconnection of physical or virtual objects embedded with sensors and technologies that communicate and share data with other devices via the Internet. This comprises a wide range of things, such as machinery, gadgets, appliances, and actuators. Each object has a distinct identity and is linked to the internet [3].

Cisco Packet Tracer is a comprehensive, networking technology teaching and learning program that offers a unique combination of realistic simulation and visualization experiences, assessment and activity authoring capabilities, and opportunities for multi-user collaboration and competition. The innovative features of Packet Tracer help students and teachers collaborate, solve problems, and learn concepts in an engaging and dynamic social environment. Some of the major features of the Cisco Packet Tracer are:

- Simulates network devices and protocols
- Supports multiple protocols and technologies
- Includes a range of network devices and topologies
- Provides a comprehensive set of networking tools
- Provides a safe and virtual environment for network experimentation
- Reduces the need for physical hardware
- Allows for quick and efficient network configuration and troubleshooting

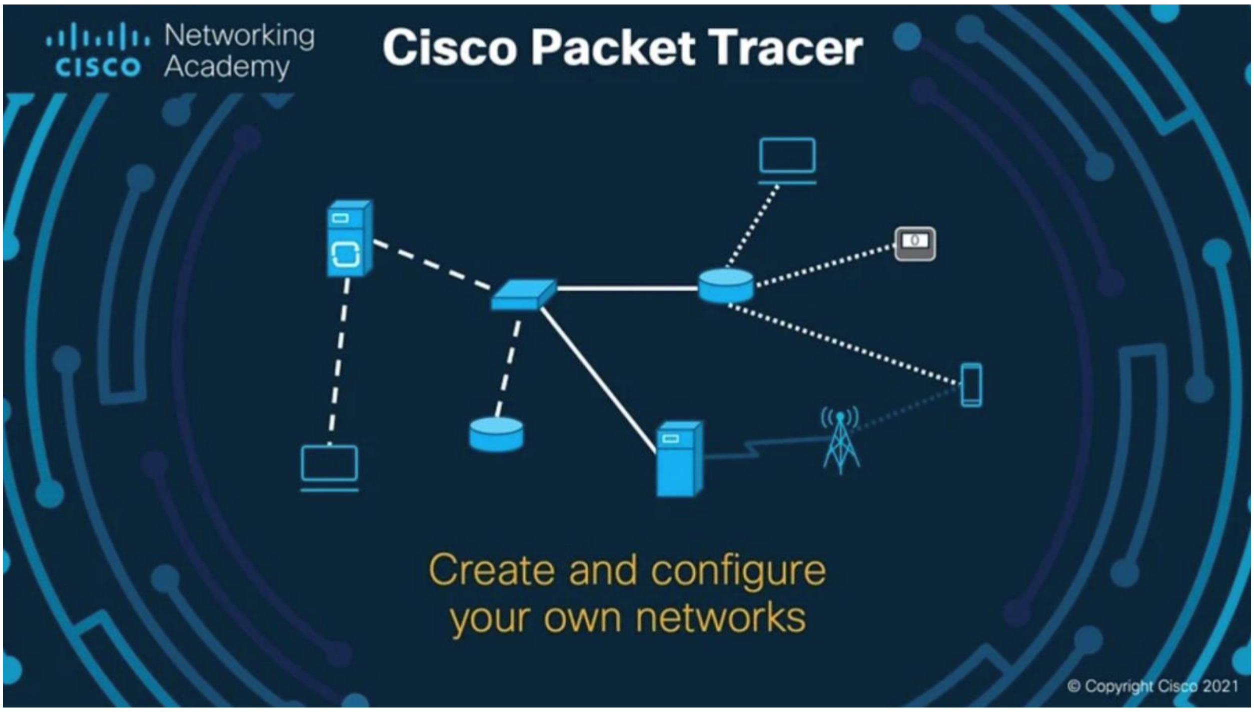

Figure 1 Cisco Packet Tracer Interface

Methodology

This study focuses on the short development life cycle of an IoT project using microcontroller, sensors, home devices and network connections. The freshly released Cisco version was picked since, in addition to networking functionality, it covers microcontrollers, sensors, home devices and provides a programming environment. Visual Basic, JavaScript, and Python are among the programming languages supported. Cisco Packet Tracer was used for developing the IoT platform because it offers a variety of network components that simulate a real network.
Cisco packet tracer enables developers to view the flow of data packets and carry out analysis on the data. A humidity sensor is used to measure the humidity level and the humidifier is set to ON/OFF according to the condition stored on the server.

IoT System Development

Router-PT has four Ethernet ports and can be used to route between different networks. Its first Ethernet port Fa0/0 is connected to the switch to which devices like humidifiers, server, and different PC's are connected. Its second Ethernet port Fa0/1 is connected to the home gateway. The home gateway has four Ethernet ports and a wireless access point configured with the "Home Wi-Fi" service set identifier (SSID) with WEP password 0123456789. Microcontrollers, PC, and smartphones are connected to the home gateway via Ethernet and Wi-Fi. The humidity sensor and LCD display are connected to the microcontroller and small JavaScript code is written in the microcontroller to read the humidity level from the environment via the humidity sensor and display it in LCD display and also provides data of the humidity level via API in the IoT server. Server is used to store the rules and control the humidifier according to the humidity level read from the humidity sensor. Fig 2 shows the entire simulation of the smart humidity control system

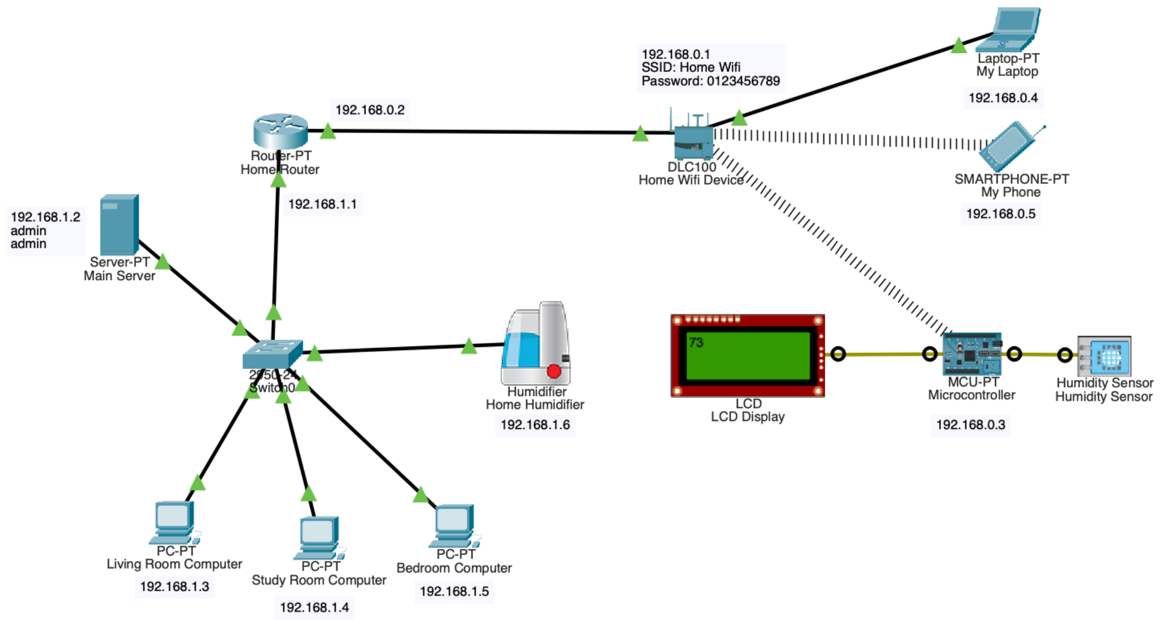

Figure 2 Smart humidity control system

The Cisco packet tracer interface provides inbuilt devices to be added to the network. The first step is to select a router from the network devices and connect the switch to its Ethernet port Fa0/0. The second step is to connect devices to the switch. In this simulation, five wired devices were attached to the switch. The devices configured included a server, humidifier, and three different PC's. The router's Ethernet port Fa0/0 was given static IP 192.168.1.1 and turned on and other devices connected to the switch were given their own different IP addresses with gateway 192.168.1.1, i.e. IP address of Fa0/0 port of a router. Fig 3 shows the IP configuration for the first Fa0/0 port of the router and Fig 4 shows the IP configuration for the server. The user was registered to the main server via a web browser in the desktop tab in the server configuration. Then the IoT server was turned on in the config of the humidifier and the remote server was set to the IP address of the main server with the username and password used while creating the user in the main server.

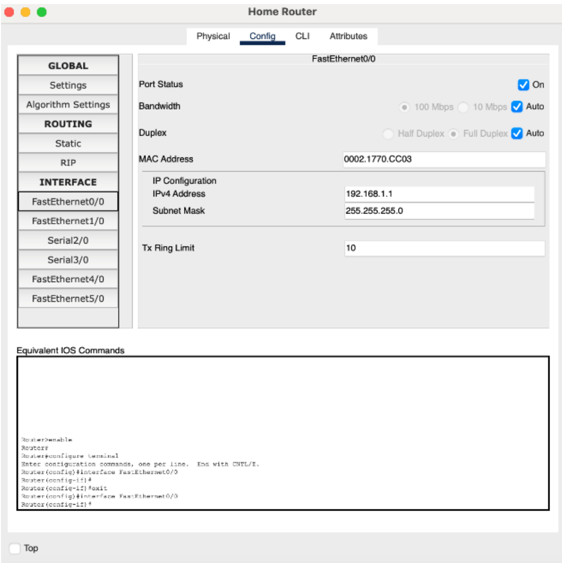

Fig 3 Router Fa0/0 config

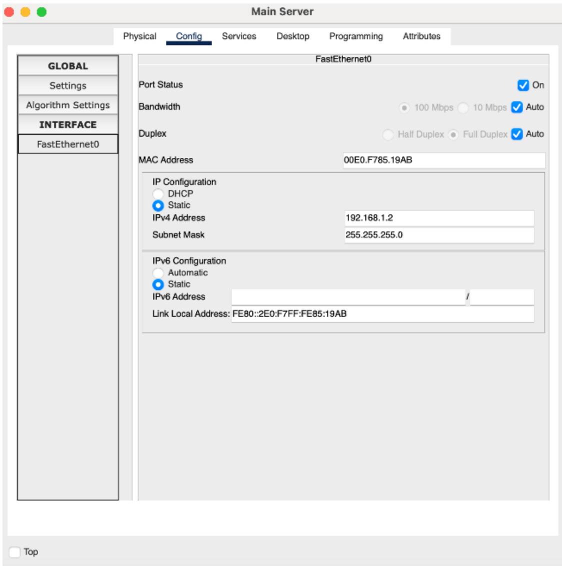

Fig 4 Server config

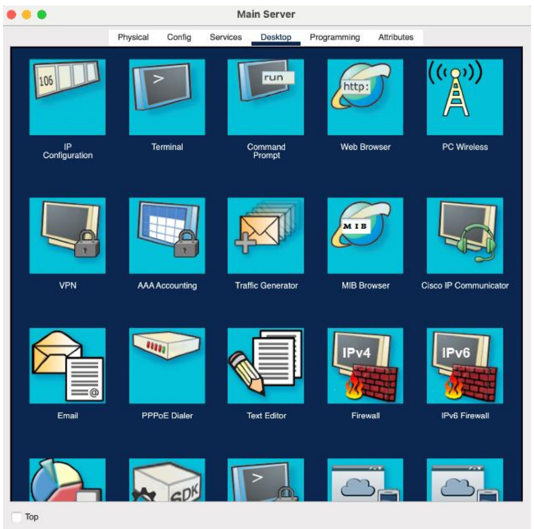

Fig 5 Main server desktop tab

Fig 6 User registration interface on the main server

Now, the home gateway is connected to the router's second Ethernet port Fa0/1 with static IP 192.168.0.2, and the gateway's IP was set to 192.168.0.1. The gateway was then configured for wireless network to have wireless SSID along with WEP password. The microcontroller and smartphone are connected wirelessly and the laptop is connected with an Ethernet cable to the home gateway. The gateway for all the devices connected to the home gateway is set to 192.168.0.2, i.e. IP address of the Fa0/1 Ethernet port of the router. Figure 7 shows, the home gateway's wireless setup, and figure 8 shows the wireless configuration for the microcontroller. Alongside the microcontroller's IoT server was also turned on with a remote server address same as for the humidifier.

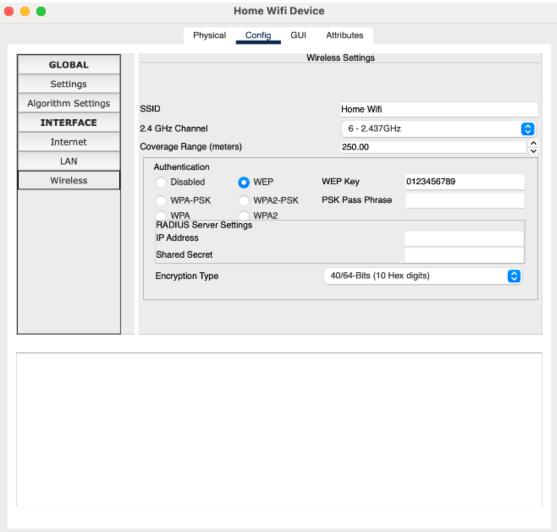

Fig 7: Home gateway Wi-Fi configuration

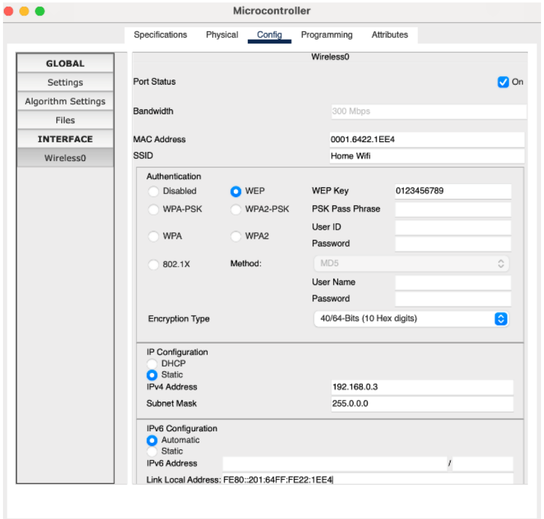

Fig 8: Microcontroller Wi-Fi configuration

In the next step, microcontroller's A0 pin was connected to the humidity sensor's pin A0, then the D0 pin was connected to LCD's D0 pin. A small JavaScript code was added in the main.js file in the programming tab of the microcontroller where the microcontroller takes the humidity reading from its A0 pin and writes in its D0 pin to the LCD display panel. Microcontroller also serves the data from the humidity sensor to the IoT server. Here's the sample code for the microcontroller:

```
// Function to set up the initial configuration
function setup() {
   // Attach an interrupt to pin A0, calling an anonymous function to process sensor data
         attachInterrupt(A0, function() {
                  processData(getSensorData());
         });
   // Setting up the IoEClient with type and state configuration
         IoEClient.setup ({
            type: "MicroController",
            states: [
               {
                  name: "Humidity",  // Name of the state
                  type: "number",    // Type of data (number)
                  unit: "%",         // Unit of measurement (percentage)
                  decimalDigits: 2,  // Number of decimal digits
               }
            ]
         });
}
// Main loop function that runs repeatedly
function loop(){
   // Read analog value from pin A0, map it to a range of 0 to 100, and round it to the nearest
integer
   var humidity = Math.floor(map(analogRead(A0), 0, 1023, 0, 100) + 0.5);
   // Report the humidity state to the IoEClient
   IoEClient.reportStates([humidity]);
   // Print the humidity value to the serial monitor
   Serial.println("Humidity: " + humidity);
}
// Function to get sensor data from pin A0
function getSensorData(){
   // Read analog value from pin A0, map it to a range of 0 to 100, and round it to the nearest
integer
   return Math.floor(map(analogRead(A0), 0, 1023, 0, 100) + 0.5);
}
// Function to process the data received from the sensor
function processData(data) {
   // Custom function to write the data (e.g., to an output device or storage)
   customWrite(0, data);
}
```

Then the server's IP was visited from any of the computers or smartphones connected and logged in with a created user account. Then it is verified that the microcontroller is sending the humidity level read from the humidity sensor correctly and the humidifier can be seen in the IoT server. Next, conditions were added in the IoT server that the humidifier turns on automatically if the humidity level is below 75% and turns off automatically if the humidity level reaches or exceeds 75%. Fig 9 shows the different IoT devices registered in the main server and Fig 10 shows the conditions to set the humidifier on and off according to humidity level in the environment.

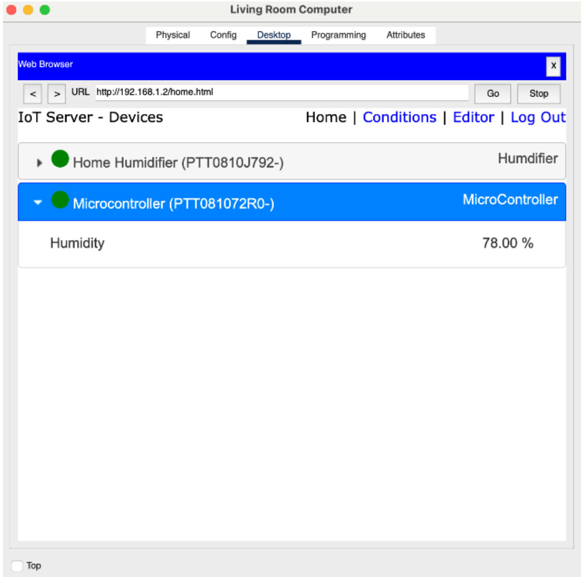

Fig 9: IoT devices registered in main server

Fig 10: Conditions in the main server

Results

After completing all the main procedures of designing a smart humidity control system, the system was tested, Fig 11 shows that the humidifier is set to ON automatically when the reading for humidity level is 74% which is below 75%. Fig 12 shows the humidifier set to OFF when humidity level is 80% which is above 75%.

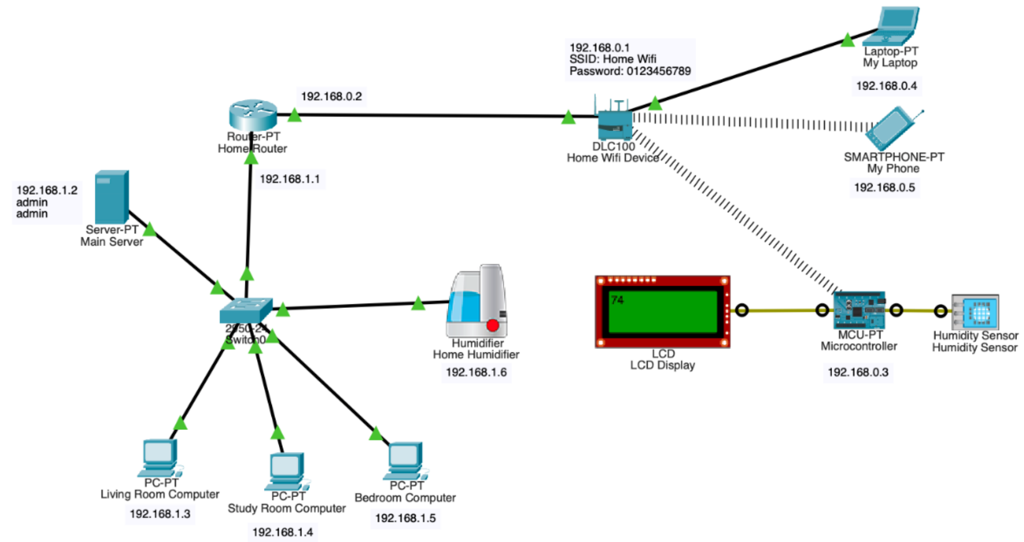

Fig 11 Humidifier status is shown on when humidity level is 74%

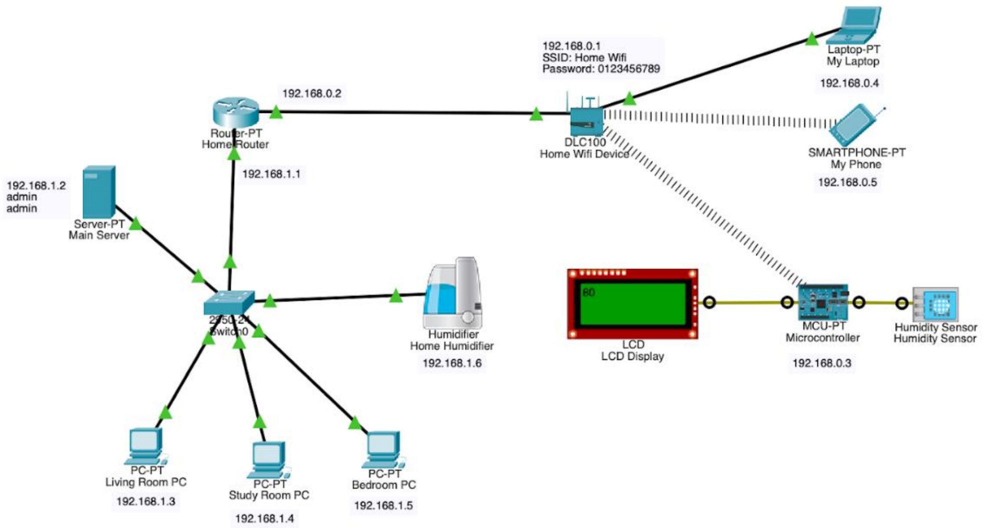

Fig 12 Humidifier status is shown off when the humidity level is 80%

At the end, a message was passed between different devices randomly in the simulation and it was found to be successful.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	Main Server	Microcontroller	ICMP		0.000	N	0	(…	(delete)
	Successful	Living Room PC	Microcontroller	ICMP		0.000	N	1	(…	(delete)
	Successful	Study Room PC	My Phone	ICMP		0.000	N	2	(…	(delete)
	Successful	Main Server	Home Humidifier	ICMP		0.000	N	3	(…	(delete)

Conclusion

This research aimed to simulate a smart humidity control system. The technology innovation and the hike in automated system usage are the motivations behind this work. Such security measures are very critical and IoT is providing a new and excellent concept to make our surroundings smarter. In this study, Cisco packet tracer was utilized to design and simulate the smart humidity control system. Cisco public tutorials and previous research were very beneficial in the implementation process. The result showed that the devices can be controlled and monitored using end-user devices. Cisco packet tracer gives various facilities that make simulation easy. The results proved that there is an opportunity to apply this model in real life. The IoT concept can be applied in various domains.

References

[1] T. Kramp, R. van Kranenburg, and S. Lange, "Introduction to the internet of things," in *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*, Springer Berlin Heidelberg, 2013, pp. 1–10. doi: 10.1007/978-3-642-40403-0_1.

[2] N. Gwangwava and T. B. Mubvirwi, "Design and Simulation of IoT Systems Using the Cisco Packet Tracer," *Advances in Internet of Things*, vol. 11, no. 02, pp. 59–76, 2021, doi: 10.4236/ait.2021.112005.

[3] G. Alfarsi, J. Jabbar, R. M. Tawafak, S. I. Malik, A. Alsidiri, and M. Alsinani, "Using Cisco Packet Tracer to simulate Smart Home." [Online]. Available: www.ijert.org