جامعة عفت
EFFAT UNIVERSITY

Project Report On

Smart Temperature and Lighting System

ECE 342L - Analog Control System

Submitted by:

Durid Al Masri

Alhassan Al Saggaf

Nawar Al Imam

Nizar Al Rayhani

Under the Guidance of:

Dr. Nema Salem

Department of Electrical and Computer Engineering

Effat University

Table of Contents

1. Abstract

In this project, we used a Raspberry Pi to create a Smart Home System that uses temperature measurements to manage household equipment. The system employs a 2-channel relay module to automatically turn on and off a fan and a light, and a DHT11 sensor to detect the room's temperature and humidity. We also developed a basic website that allows users to monitor the sensor data and, if they so want, manually operate the devices in order to make the system easier to use. We choose the Raspberry Pi because it can run Python scripts smoothly, manage several jobs at once, and connect to the internet with ease. This project demonstrates how control engineering and embedded systems may be applied to improve the comfort, intelligence, and efficiency of commonplace residential settings.

2. Introduction

As more individuals search for methods to improve the comfort, energy efficiency, and automation of their homes, smart home technology is becoming more widespread. Embedded systems—small, specialized devices with the ability to perceive, decide, and act—are one of the main instruments used to enable smart homes. The goal of this project is to use the Raspberry Pi, a powerful and inexpensive single-board computer, to create a straightforward but efficient smart home system.

Our system's primary function is to regulate household appliances, such as lights and fans, according to humidity and temperature. We accomplish this by using a relay module to turn devices on or off and a DHT11 sensor to gather environmental data. Additionally, we included a web interface for customers to view the current conditions and, if necessary, remotely operate the system. This project demonstrates how real-world automation may be achieved by designing a simple embedded system using inexpensive components and straightforward programming. It also demonstrates how the Raspberry Pi provides smart home applications with a solid combination of control, connectivity, and user involvement.

3. System Design

3.1. Overview of the System:

Our Smart Home System is intended to automatically operate connected items like a fan and a light while also keeping an eye on the temperature and humidity levels in the space. The brain of the system is a Raspberry Pi, which continually gathers information from a DHT11 sensor and utilizes it to inform choices. For instance, the fan activates when the temperature rises over a predetermined level. The fan shuts off when it falls back down.

The system comes with a basic webpage that lets the user examine real-time sensor data and, if they'd like, manually control the fan and light in addition to automated control. Because of this, users may depend on automation or assume direct control as necessary, giving the system flexibility.

3.2. System Architecture:

The architecture of the system is built around the Raspberry Pi 4, which handles both the sensor readings and the control logic. Here's how it works:

1. **The DHT11 sensor** measures temperature and humidity and sends the data to the Raspberry Pi through a GPIO pin.
2. **The Raspberry Pi** processes the incoming data using a Python script.
3. Based on the temperature value, it sends signals to a **2-channel relay module**, which turns the **fan** and **light** on or off.

4. The system also runs a **Flask-based web server**, which lets the user monitor sensor data and send control commands from a web browser on the same network.
5. All operations are powered using a stable 5V power supply connected to the Raspberry Pi.

This setup allows both **automatic** and **manual (web-based)** control of the appliances.

3.3. Components Specifications

Components	*Specifications*	
Raspberry Pi 4	**Main controller that processes data, runs scripts, and hosts the web server**	
DHT11 Sensor	Measures temperature and humidity	
2-Channel Relay	Switches the fan and light on/off based on control logic	
Fan (5V or 12V)	Used to cool the room when temperature is high	
Light (Led/Bulb)	Controlled manually or automatically through the relay	
Flask Web Server	Hosts a local webpage to monitor and control the system	
Power Supply (5V, 2.5A)	Powers the Raspberry Pi and connected modules	

Here's the **Components and Connections** section drafted for your report based on your Smart Home System with Raspberry Pi, DHT11, 2-channel relay, fan, and light:

4. Components and Connections

4.1. Raspberry Pi 4 Model B

The Raspberry Pi acts as the main controller, running the software that reads sensor data, processes it, and controls appliances via the relay. It also hosts the web interface for remote monitoring and control.

- **Power:** Powered by a stable 5V, 2.5A supply through the USB-C port.
- **GPIO Pins:** Used for sensor input and relay control signals.

4.2. 2-Channel Relay Module

The relay module acts as a switch to turn appliances on or off based on control signals from the Raspberry Pi.

- **Connections:**
  - Relay IN1 and IN2 connected to Raspberry Pi GPIO pins (e.g., GPIO17, GPIO27)
  - Relay VCC connected to 5V pin on Raspberry Pi
  - Relay GND connected to Ground pin on Raspberry Pi
- **Outputs:** Connected to the Fan and Light devices. Each relay channel controls one appliance.

4.3. DHT11 Temperature and Humidity Sensor

This sensor measures the environmental temperature and humidity and sends the digital data signal to the Raspberry Pi via a GPIO pin.

- **Connections:**
  - VCC to 3.3V pin
  - Data to GPIO pin (e.g., GPIO4)
  - GND to Ground pin

4.4. Fan and Light

The appliances controlled by the relay. They are connected to the relay's normally open (NO) contacts and powered separately through the relay switch.

- **Fan:** Connected to relay channel 1
- **Light:** Connected to relay channel 2

4.5. Power Supply

A stable 5V power supply powers the Raspberry Pi and relay module. Appliances like the fan and light may require their own power source depending on their voltage and current ratings.

5. System Workflow

Data Acquisition:

The ambient temperature and humidity are continually measured by the DHT11 sensor. It uses a specific GPIO pin to digitally transmit this data to the Raspberry Pi.

Data Procession:

A Python application that runs on the Raspberry Pi periodically reads the sensor data. In order to control the fan and light, it analyses the temperature value and compares it to preset criteria.

Decision Making and Control:

The Raspberry Pi turns on the fan via the relay channel if the temperature rises over the upper threshold, cooling the space.

The Raspberry Pi turns off the fan if the temperature drops below the lower threshold.

In a similar vein, lighting control may be automated using human input or preset logic.

User Interaction via Web Interface:

- A user-friendly webpage is hosted using a Raspberry Pi-powered Flask-based web server. With this interface, users can:
- See the temperature and humidity values in real time.
- To switch the fan or light on or off, manually override the automated control.

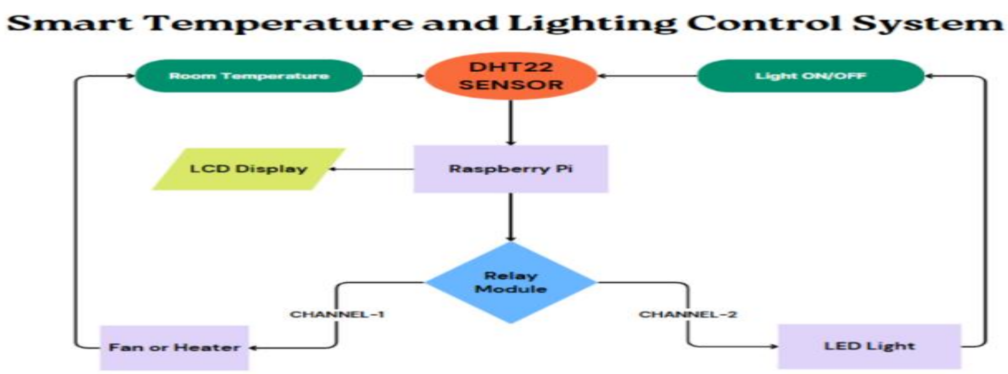
Relay Operation:

The 2-channel relay controls the power supply to the fan and light, turning them on or off in response to signals from the Raspberry Pi.

Continuous Monitoring:

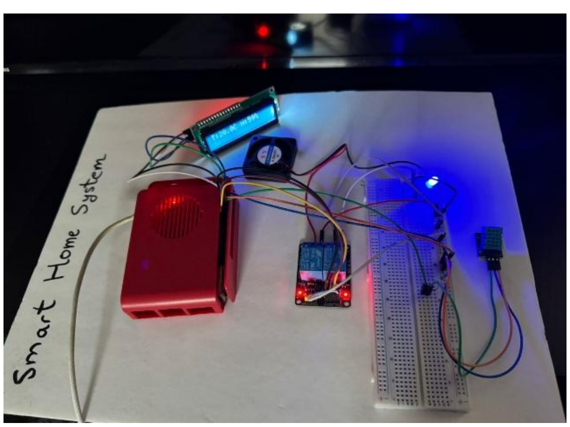
In order to guarantee that the environment is constantly observed and managed, either automatically or by human commands, the system repeatedly goes through these processes.

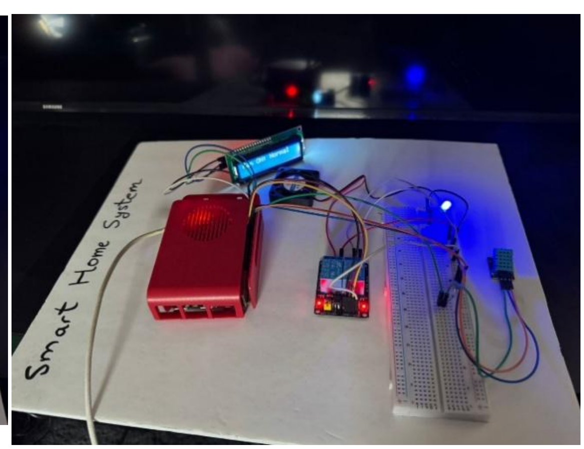
Block Diagram:

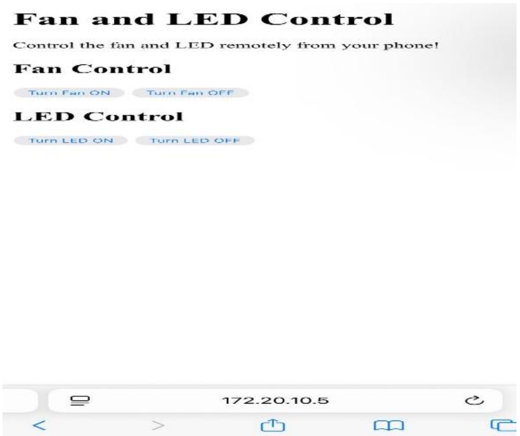


Result:

The Raspberry Pi, DHT11 sensor, 2-channel relay module, fan, and light were used to successfully install and test the smart home system. The following results were noted:







<u>Precise Sensor Readings:</u>

The DHT11 sensor regularly recorded humidity and temperature within the anticipated limits. Both the web interface and the Raspberry Pi terminal showed these measurements in real time.

<u>Dependable Automatic Control:</u>

The fan was automatically turned on when the temperature over the predetermined threshold (for example, 28°C) and off when the temperature fell below the lower limit (for example, 24°C). The fan and light were promptly and flawlessly turned on and off by the relay module.

<u>Web Interface Functionality:</u>

Using the internet, users may manually regulate the fan and light and keep an eye on environmental data. Effective two-way communication was demonstrated by the Raspberry Pi's rapid reception of commands sent from the web interface.

<u>System Stability</u>: The configuration ran flawlessly for long stretches of time, demonstrating the dependability of both the software and hardware.

Energy Efficiency:

The system efficiently optimized energy use by automating appliance management depending on ambient conditions, only turning on the fan and light when necessary.

6. Conclusion

This Smart Home System project effectively illustrated how to automate and improve home comfort and energy efficiency using embedded systems and analogue control principles. We created a system that can automatically control lights and fans and monitor environmental conditions by utilizing a Raspberry Pi, a DHT11 sensor, and a 2-channel relay module.

One important component of contemporary smart homes is the ability to remotely operate equipment and monitor sensor data, which was made possible by the addition of a web-based interface. The system demonstrated the strength of the hardware and software integration by functioning dependably under constant usage.

All things considered, this project demonstrates the benefits of utilizing Raspberry Pi for Internet of Things and smart home applications because of its networking capabilities, multitasking capability, and programming simplicity. It provides a scalable framework for further improvements, including spreading automation to other household appliances, incorporating voice control, or adding more sensors.

7. References:

- ChatGPT
- *AI Paraphrasing Tool*. (n.d.). QuillBot. https://quillbot.com/paraphrasing-tool
- Raspberry Pi OS – Raspberry Pi
- Free Circuit Diagram Maker: Draw a circuit diagram | Canva
- https://www.scribbr.com/citation/generator/