IoT Project Proposal

Project Title: Personalised Climate Control System

Team ID: 46

Introduction

The Project aims to optimize indoor climate to enhance user comfort and energy efficiency in offices and homes.

A. Problems we're trying to solve:

1. Inefficient usage of resources by traditional indoor HVAC systems, which usually waste energy on unoccupied areas.
2. Lack of real-time adaptation to external climate.
3. Lack of proper ventilation system in closed rooms that can get crowded.

B. Objectives of the project

1. Use temperature and humidity sensors to monitor indoor conditions.
2. Implement actuators to mimic control of HVAC (Heating, Ventilation, and Air Conditioning) systems automatically using remotes(controlled by motor), fans and stepper motor.
3. Develop a system to adjust indoor climate based on occupancy and outdoor temperature.
4. Offer a web user interface for manual control and monitoring.
5. Test the system for effectiveness in different room sizes and conditions.

C. Hardware Requirements:

Actuators:

1. **Fan (L9110)**

   Purpose: It is used as a miniature Fan in real life as use of actual Fan is not feasible in this project.

   Quantity: 1

   Link:

   https://robu.in/product/l9110-fan-module-fire-fighting-robot-arduino/

2. **Stepper motor**

   Purpose: It is used to open and close windows for ventilation as use of actual AC is not feasible in this project.

   Quantity: 2

   Link:

   https://robokits.co.in/motors/stepper-motor/stepper-motor-without-gearbox/stepper-motor-5v-unipolar-arduino-compatible?srsltid=AfmBOorBGC5r9uU80bnMqEtbD3vTPJxbvwueDSyEBN1FxoyMOsmxnyI2ETs

3. **KY-005 IR transmitter module**

   Purpose: The KY-005 is a simple IR transmitter that can send signals to an AC, similar to a remote control.

   Quantity: 1

   Link:

   [KY-005 38KHz Infrared IR Transmitter Sensor Module buy online at Low Price in India - ElectronicsComp.com]

   Description of usage:

   Hardware Connection:

1. The KY-005 module will be connected to a GPIO pin on the ESP32 microcontroller
2. Power will be supplied through the ESP32's 3.3V or 5V output
3. The module requires minimal current and can be directly driven by the ESP32

   Signal Transmission:

1. The ESP32 will generate precisely timed digital signals
2. These signals will be transmitted as infrared light pulses by the KY-005
3. The AC unit will receive and interpret these signals as commands

   Control Capabilities:

1. Power on/off functionality
2. Temperature adjustment (increase/decrease)
3. Mode selection (cool, heat, fan)
4. Fan speed control

The system will determine when and how to control the AC based on:

1. Real-time temperature and humidity readings from sensors
2. User-defined comfort preferences
3. Occupancy patterns in the room

4. **LED (Mimicking an AC in HVAC system):**

Purpose: It is used to mimic an AC in real life as use of actual AC is not feasible in this project. (Can be used as an alternative in case physical limitations and surroundings affect the working of the actual system)

It is thus needed to symbolize the fact of AC being turned on by the system using the relay module.

Quantity: About 10 sensitive LEDs are required (preferably of different colours)

Link:

https://robocore.in/product/white-led-5mm-2/?srsltid=AfmBOoqE9VozgheaFi5KGgdcPfpTShCx5EKpZtGt5QF3zkakxZzOQtkabNY&com_cvv=d30042528f072ba8a22b19c81250437cd47a2f30330f0ed03551c4efdaf3409e

5. **Relay Modules:**

Purpose: The primary function of a relay module is to switch electrical devices or systems on and off. It also serves to isolate control circuits, ensuring that low-power devices, such as microcontrollers, can safely control higher voltages and currents.

Series: RE51 (5V relay module)

Quantity: 2 relay modules are required

It is needed to control high power requiring actuators from the microcontrollers.

Link:

1 CHANNEL 5V RELAY MODULE RE51 – QBM INDIA

Functioning of the actuator:

A relay module actuator works like a switch controlled by a low-voltage signal. When the signal is applied, it energizes the relay's coil, creating a magnetic field that either opens or closes the relay's contacts. This allows it to control high-voltage circuits using a low-voltage control signal. It's commonly used in various applications like personalised climate control systems and industrial control systems.

Hardware Implementation Steps (a rough overview) :

1. Choosing Relay and Motor: We have picked a relay module suitable for our motor's requirements. (the motor here is a DC servo motor).
2. Using a Microcontroller: We will use ESP-32 to send signals to the relay.
3. Connecting Motor to Relay: Attach the motor to the relay so the relay can control its power.
4. Power Supply: We will make sure that we use power batteries for this purpose.
5. Wire Everything: Connect the relay, motor, microcontroller, and power supply properly.
6. Program the Microcontroller: We will write appropriate code for controlling the relay and DC servo motor.
7. Test and Adjust: We will make all necessary tweaks to get our implementation working smoothly.

5. Power Development Boards (PWD)

1. ESP 32 (quantity-required-3 ESP32 boards are required)
2. Arduino (1 board is required)

6. Others

1. Breadboard
2. Connecting Wires
3. Batteries (preferably 5 V) (quantity- 5 batteries are required)
4. Dummy fan (to be mounted on servo motor)

Sensors:

1. **Temperature and humidity sensor (DHT22)**

Measures ambient temperature and humidity level.

Series: DHT22

Quantity: At least 2 sensors are required

It is needed to measure the temperature and humidity of the room as well as the temperature outside and it will contribute to the project by providing the temperature and humidity readings which are essential to make the climate control system.

Link of the sensor:

[Buy DHT22 Digital Temperature and Humidity Sensor Module In India]

Details of hardware implementation:

1. **Connecting the Sensor:** Attach the sensor to the microcontroller (Arduino Uno or ESP32) using the correct pins (analog or digital) and ensure proper wiring.
2. **Reading Data:** Write a program to collect data from the sensor through the microcontroller's pins.
3. **Calibration:** Compare the sensor readings with those from a reference device (such as another DHT11 sensor) to identify any differences and adjust the sensor accordingly.
4. **Using the Data:** Display the sensor readings on a screen (like a computer monitor) and use them to control LEDs or other connected hardware.

2. Air Quality sensor (MQ135)

MQ135 is an air quality sensor, which belongs to the series of MQ gas sensors, is widely used to detect harmful gases, and smoke in the fresh air.

Series: Q135

Quantity: 2 sensors of this type are required

It is needed to measure the air quality of the room and it will contribute to the project by providing the readings which are essential to make decisions on the level of carbon dioxide in the room. Also, it can detect various other harmful gases in the room.

Link of the sensor:

[Buy MQ 135 Air Quality/Gas Detector Sensor Module For Arduino Online at Robu.in]

Details of hardware feature implemented:

1. **Sensor Interface**: Connect this sensor to the microcontroller (Arduino Uno or ESP32) using appropriate pins (analog or digital) and ensure proper wiring.
2. **Data Acquisition**: Write code to read sensor data (analog or digital signals) from the microcontroller pins.
3. **Calibration**: Implement calibration routines if necessary to ensure accurate readings from the sensors.
4. **Feedback Mechanism:** After implementing these hardware features and architecture, we will utilize the sensor data to drive the LED display monitor (preferably our computer monitor screens) to display the required data to the user and activate LEDs or other hardware components accordingly.

3. Ultrasound sensor (HCSR04) and Infrared sensor

It will be used to measure occupancy in an indoor location and to generate signals accordingly. It will be implemented using 1 IR motion sensor and 1 ultrasonic sensor where it will keep count of persons entering and exiting a room ultimately giving the occupancy.

Series:HC-SR04 sensor (ultrasonic sensor) and a simple IR sensor

Quantity: 1 HC-SR04 sensor and 1 IR sensor

Link of the sensor:

Ultrasound sensor (HCSR04):

https://robu.in/product/hc-sr04-ultrasonic-range-finder/

Infrared sensor:

https://robu.in/product/ir-infrared-obstacle-avoidance-sensor-module/

4. Light Sensor (LDR)

It is used to measure the light in the room.

Series: LM393

Quantity: 2 sensors are required

It is needed to measure whether the room is currently in use, also it can be used to measure the amount of sunlight received by the room.

Details of hardware implementation:

1. **Sensor Interface:** Connect each sensor to the microcontroller (Arduino Uno or ESP32) using appropriate pins (analog or digital) and ensure proper wiring.
2. **Data Acquisition:** Write code to read sensor data (analog or digital signals) from the microcontroller pins. (here analog)
3. **Calibration:** Implement calibration routines if necessary to ensure accurate readings from the sensor.
4. **Feedback Mechanism:** After implementing these hardware features and architecture, we will utilize the sensor data to drive the LED display monitor (preferably our computer monitor screens) to display the required data to the user and activate LEDs or other hardware components accordingly.

Data collection plan

A. Plan

The basic idea is to collect and store the data using OM2M which is an open -source IoT platform that provides functionalities for managing IoT devices and nodes along with data. For storing the data, we need to place the sensors in a systematic and strategic manner to obtain accurate variations in various physical parameters such as temperature,humidity etc. Following steps can be implemented:

Sensor Placement: Placing temperature and humidity sensors strategically throughout the indoor environment to capture variations in temperature and humidity levels across different areas.

Sampling Frequency: Setting up the sensors to collect data at regular intervals, such as every few seconds or minutes, depending on the desired granularity of monitoring and control. For this project we will retrieve data from our sensors continuously for ten days one hour per day and store the data in a suitable database.

Occupancy Detection: We will integrate occupancy sensors into the system to detect room occupancy status. These sensors should be placed in areas where occupancy status is critical for climate control adjustments.

Let's ponder on how data will be transmitted and stored on OM2M server. Following steps would be implemented:

1. **Set Up OM2M:** We will install and configure the OM2M platform on our server or local machine.
2. **Define Resources:** We will define resources(like node buckets) in OM2M to represent our IoT devices and sensors. Each sensor shall be represented by a unique resource within the OM2M platform.

3. **Collect Sensor Data:** We will write Arduino codes to read sensor values and send them to OM2M using standardized IoT protocols such as HTTP.
4. **Create Containers:** We will create containers within OM2M to store sensor data. Containers are used to organize and store data collected from IoT devices.
5. **Send Data to OM2M:** We shall configure our IoT devices to send sensor data to OM2M using the appropriate APIs or protocols.
6. **Handle Data Storage:** OM2M provides functionalities for storing and managing data received from IoT devices. As sensor data is sent to OM2M, it is stored in the designated containers within the platform's database.
7. **Access Stored Data:** We will use OM2M APIs and implement a dynamic user friendly dashboard (HTML webpage) so that users can retrieve all the required sensor data. (We will include all necessary hyperlinks in the web page to redirect users to our OM2M databases) .We shall query the OM2M platform to retrieve historical data, analyze trends, or visualize sensor readings over time.

We will also use MQTT communication protocol for publishing data on Thingspeak channel and use MATLAB and MathWorks which give necessary plots and graphs which will add to the overall visual experience of the user.

B. Outcomes Inferred from data:

In addition to monitoring and analysis of indoor climatic conditions, following outcomes can be inferred from the data retrieved from the sensors:

1. **Indoor Environmental Quality:** Data from the DHT11 sensor can provide insights into indoor temperature and humidity levels. By analyzing this data, one can assess indoor comfort levels, identify potential issues such as poor ventilation or moisture problems, and optimize indoor environmental conditions for comfort and health.
2. **Air Quality Monitoring:** The MQ135 gas sensor can detect a variety of air pollutants such as carbon dioxide (CO_2), volatile organic compounds (VOCs), and various gases present in indoor or outdoor environments. By monitoring these pollutants, one can assess air quality levels, identify sources of pollution, and take measures to improve air quality and reduce health risks.
3. **Weather Monitoring:** The DHT11 sensor measures temperature and humidity, which are important parameters for weather monitoring and forecasting. By analyzing this data, one can track changes in temperature, monitor weather patterns, and provide localized weather forecasts.
4. **Health and Safety:** Sensor data from the DHT11 and MQ135 sensors can be used for health and safety monitoring. For example, monitoring indoor temperature and humidity levels can help prevent heat-related illnesses or mold growth, while detecting harmful gases with the MQ135 sensor can alert users to potential health hazards such as carbon monoxide (CO) or ammonia (NH_3) leaks.

Conclusion:

To conclude we can say that this project aims at improving the user comfort and optimizing the energy utilization in the field of HVAC (Heat, Ventilation, Air Conditioning) using IoT. The user's preferences are accounted for and automation is provided based on analysing the data collected from the sensors. It optimizes energy usage based on real time data, giving it the potential to significantly improve comfort levels, reduce energy costs, and minimize environmental impact.

The proposed Personalized Climate Control System employs IoT technology to provide adaptive indoor climate control, improving comfort and energy efficiency in homes and offices. The system promises personalized climate management using temperature, humidity, air quality, and light sensors, as well as actuators and human interaction data. The integration of IR blasters allows for seamless control of existing HVAC equipment without requiring replacement, making the system both cost-effective and compatible with legacy infrastructure. Additionally, motorized windows provide automated natural ventilation capabilities, intelligently opening and closing based on indoor and outdoor conditions to optimize air quality and reduce the need for mechanical cooling when possible.

A complete data collection strategy of at least ten days captures numerous environmental characteristics such as temperature variations, humidity levels, air quality, occupancy patterns, and user preferences, allowing for real-time decision making and analysis. This data supports the development of control algorithms that allow for dynamic responses to changing conditions and user needs, with the goal of increasing user pleasure while reducing energy use. Continuous monitoring and performance analysis, combined with proactive maintenance and optimization activities, ensures the system's long-term reliability and efficiency.