Smart Home Automation Project

## Created by (Name, Details)

Table of Contents

[Created by (Name, Details) 1](#__RefHeading___Toc686_3150598930)

[Overview 3](#__RefHeading___Toc688_3150598930)

[Requirements & Components Used 3](#__RefHeading___Toc690_3150598930)

[The Circuit Design 3](#__RefHeading___Toc692_3150598930)

[The components were connected as given below 3](#__RefHeading___Toc694_3150598930)

[The Program Excution Code 4](#__RefHeading___Toc696_3150598930)

[C Program 4](#__RefHeading___Toc698_3150598930)

[For Python 5](#__RefHeading___Toc700_3150598930)

[Other Services Used 5](#__RefHeading___Toc717_3150598930)

[Simulation 6](#__RefHeading___Toc702_3150598930)

# Overview

We creating smart home IoT Device with Arduino in Wokwi Simulator In this project, we'll create a simple simulation of a smart home system that includes a temperature sensor and a relay-controlled lamp.

# Requirements & Components Used

In this project, we have used the following components

* Arduino Uno
* DHT22 Temperature Sensor (for simulating temperature data)
* Relay Module (for controlling a lamp)
* LED (to represent the lamp)
* Breadboard and wires to connect the components

# The Circuit Design

## *The components were connected as given below*

* We connected the VCC and GND pins of the DHT22 to the 5V and GND on the Arduino.
* We connected the data pin of the DHT22 to any digital pin on the Arduino (e.g., pin 2).
* We connected one end of the relay module to a digital pin on the Arduino (e.g., pin 3) and the other end to GND.
* We connected one leg of the LED to the NO (Normally Open) terminal of the relay module.
* We connected the other leg of the LED to a current-limiting resistor (330-470 ohms) and then to the 5V on the Arduino.

# The Program Excution Code

## *C Program*

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#define DHTPIN 2 // DHT22 data pin

#define DHTTYPE DHT22 // DHT22 sensor type

#define RELAY\_PIN 3 // Pin connected to the relay module

DHT\_Unified dht(DHTPIN, DHTTYPE);

float temperature = 0.0;

void setup() {

dht.begin();

pinMode(RELAY\_PIN, OUTPUT);

digitalWrite(RELAY\_PIN, LOW); // Turn off the relay initially

}

void loop() {

sensors\_event\_t event;

dht.temperature().getEvent(&event);

temperature = event.temperature;

// Check the temperature and control the lamp

if (temperature >= 25.0) {

digitalWrite(RELAY\_PIN, HIGH); // Turn on the lamp

} else {

digitalWrite(RELAY\_PIN, LOW); // Turn off the lamp

}

delay(2000); // Delay for 2 seconds

}

## *For Python*

import Adafruit\_DHT

DHTPIN = 2 # DHT22 data pin

DHTTYPE = Adafruit\_DHT.DHT22 # DHT22 sensor type

RELAY\_PIN = 3 # Pin connected to the relay module

dht = Adafruit\_DHT.DHT22(DHTPIN)

temperature = 0.0

def setup():

dht.begin()

GPIO.setup(RELAY\_PIN, GPIO.OUT)

GPIO.output(RELAY\_PIN, GPIO.LOW) # Turn off the relay initially

def loop():

humidity, temperature = Adafruit\_DHT.read\_retry(DHTTYPE, DHTPIN)

# Check the temperature and control the lamp

if temperature >= 25.0:

GPIO.output(RELAY\_PIN, GPIO.HIGH) # Turn on the lamp

else:

GPIO.output(RELAY\_PIN, GPIO.LOW) # Turn off the lamp

time.sleep(2) # Delay for 2 seconds

\*\*Step 6: Upload the Code\*\*

Copy and paste the code into the Arduino IDE within the Wokwi platform. Then, click the "Upload" button to program the Arduino.

# Other Services Used

We have used Blynk IoT Management Platform to ensure that we can control the IoT device remotely

We have also successfully implemented with Firebase

We have used the MQTT protocol and for which we used HiveMQ protocol broker

We have a created and android app using the MIT appinventor 2

# Simulation

Once the code is uploaded, and we run the simulation. We observe how the temperature sensor data affects the lamp (LED) based on the defined threshold (25.0°C in this example).