

Project Module2: Microcontroller & Interfacing

Wireless Home Automation System

BSCE22 | Spring 2024 | Department of Computer & Software Engineering

# Group Members:

1. Ariba Mumtaz | BSCE22020
2. Rameeza Rahim |BSCE22052
3. Waleed Hussain |BSCE21030

# Table of Contents

|  |  |
| --- | --- |
| Titles | Page Number |
| About the project | 2 |
| How MQTT is being used | 2 |
| Device by Device Breakdown & Interfaces being used | 3 |
| Resource links for reference | 6 |

# About the Project

This project aims to make a wireless smart home automation system that has various devices and they all connect to a central server. This project uses open source software such as Home Assistant as the automation platform and EMQX as the MQTT server/broker through which MQTT communication method is established. The plan is to have three devices, the first is a sensing device, second is a control device that can control other high current devices and third is bell that can also detect of the gate is left open.

# How MQTT is being used

Message Queuing Telemetry Transport (MQTT) is a simple lightweight communication protocol that is based on a subscribe and publish model and allows two-way communication between devices. It uses standard tcp/ip stack but can be implemented through other means of communication such as ZigBee or zwave.

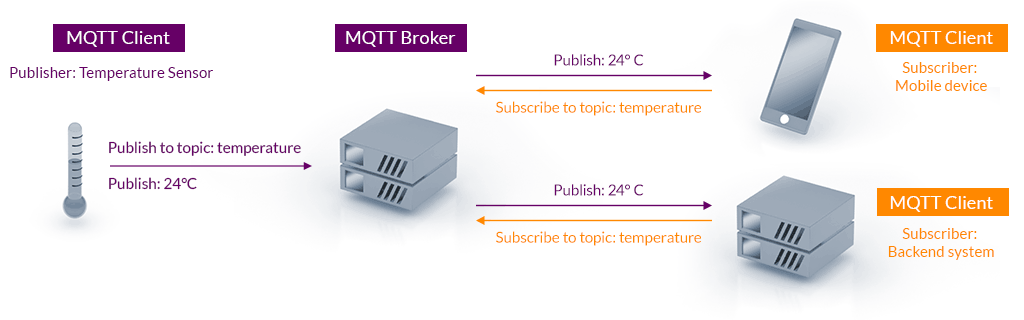


Figure 1: Source "mqtt.org"

As figure 1 illustrates, a mqtt has topics that can be predefined or made on subscription. A client such as a sensor publishes its data on a specified topic to a central server known as the MQTT broker which handles this data and forwards it to the clients that has subscribed to that topic. If any two or more clients are subscribed to the same topic then information sent by one device is broadcasted to all the other devices on the same topic allowing for two way communication between devices.

## EMQX and Home Assistant

Home Assistant is the automation and monitoring platform that is also another MQTT client. EMQX is the sever platform that acts as the broker and facilitates communication.

The idea here is that any sensor value read sent is also read by Home Assistant and compare to the preset triggers. For example, if it detects bell has pressed then it would sound the bell in the house or if it detects that I am not at home and a window or door is opened as if some burglary is being attempted, it will raise the alarm in the house, notify the homeowner and can also call emergency services.

## Why Home Assistant and EMQX

Traditionally smart home devices have been built using proprietary technology and locked down code. This means that not only you can’t repair them, you also have to rely on the built in security of the devices. This meant that any data these devices send over the internet to the manufacturers servers can be used and stored illegally and illicitly to track and target homeowners. This is a breach of privacy and a breach of private information. Such data includes location tracking, camera surveillance, house awareness and even daily routines of the user which if used illegally can allow any hacker or even the manufacturer themselves to know when the home owner is away or his daily activities to do things such as robb houses when the home owner is away, remotely control and monitor other devices on the network and steal passwords and credit card information remotely without letting anyone know simply by monitoring the network traffic.

Home Assistant and EMQX on the other hand are open-source technologies, meaning their code is open to all to be edited or modified and both can be hosted on any computer, even on a raspberry pi and other low power single board computers. This allows for complete control of the information and can be air gapped. This allows for anyone to be built integrations to allow any number of vendors and manufacturers to integrate their devices with home assistant and engineers like us to make custom devices like the ones we are making in this project and integrate them using technologies like MQTT with minimal effort.

# Device Breakdown

In addition to the peripherals being used here, timers and interrupts may be used where appropriate.

## Device 1 (sensing board)

**Sensors used:**

1. SHT30 temperature and humidity sensor
2. LDR module (Light sensor)
3. Window open and close magnetic switch sensor

**Actuators**

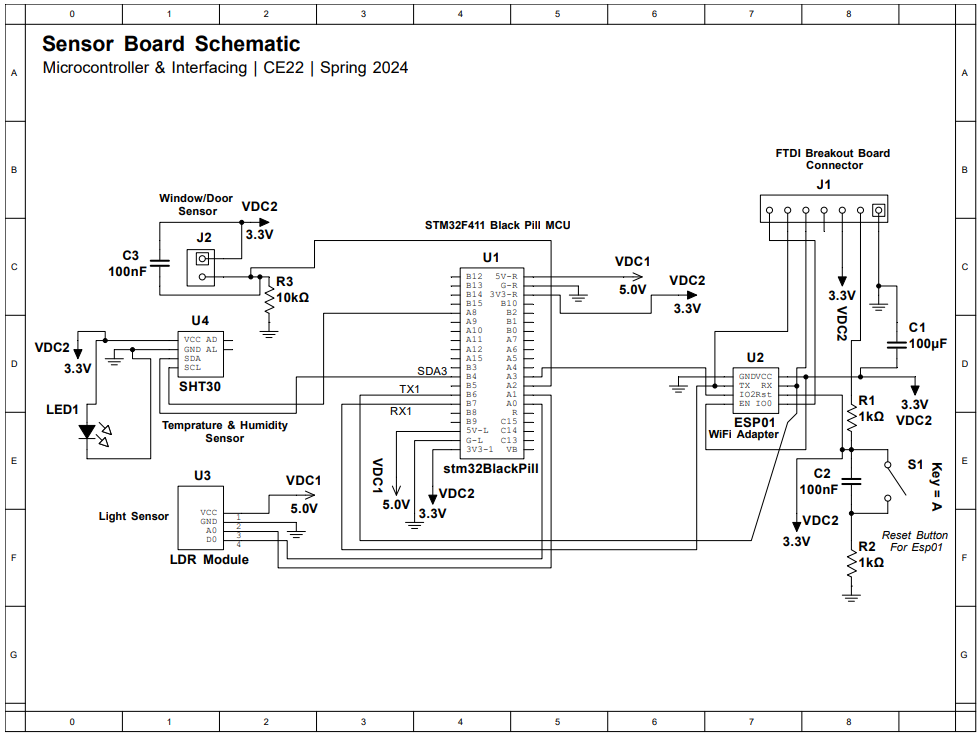
This board will have no actuators, it will be sensing only device. The only actuation is to read sensor values and send them over to the server using ESP01 Wi-Fi module.

**Peripherals Used**

1. UART
2. I2C
3. ADC
4. Digital Inputs and outputs

**Schematic**

*Next Page ->*

****

## Device 2 (Actuation Board)

**Sensors used:**

1. LDR module (Light sensor)

**Actuators**

1. 8 Relay Channels (can be further connected to AC or DC devices)
2. Buzzer
3. ESP01

**Peripherals Used**

1. UART
2. ADC
3. Digital Inputs and outputs

**Schematic**

# 

## Device 3 (The Bell)

**Sensors used:**

1. Bell Button Input
2. Door Sensor Switch Input

**Actuators**

1. ESP01

**Peripherals Used**

1. UART
2. Digital Inputs and outputs

**Schematic**

# Resource links for reference

1. <https://www.emqx.io/> (EMQX)
2. <https://www.home-assistant.io/> (Home Assistant)
3. <https://www.instructables.com/Getting-Started-With-the-ESP8266-ESP-01/> (getting started with esp01)
4. <https://www.microchip.ua/wireless/esp01.pdf> (ESP01 Datasheet)
5. <https://www.mouser.com/pdfDocs/ProductOverview_DFRobot-DFRobot-STM32F411-BlackPill-Development-Board.pdf> (STM32 Black Pill Datasheet)