4. AC Control APP

1. Files:

a. MainActivity(app)
b. Scheduler(app)
c. TCPClient(app)
d. ACControl.ino(ESP)
e. IRRecord.ino(ESP)

2. Architecture:

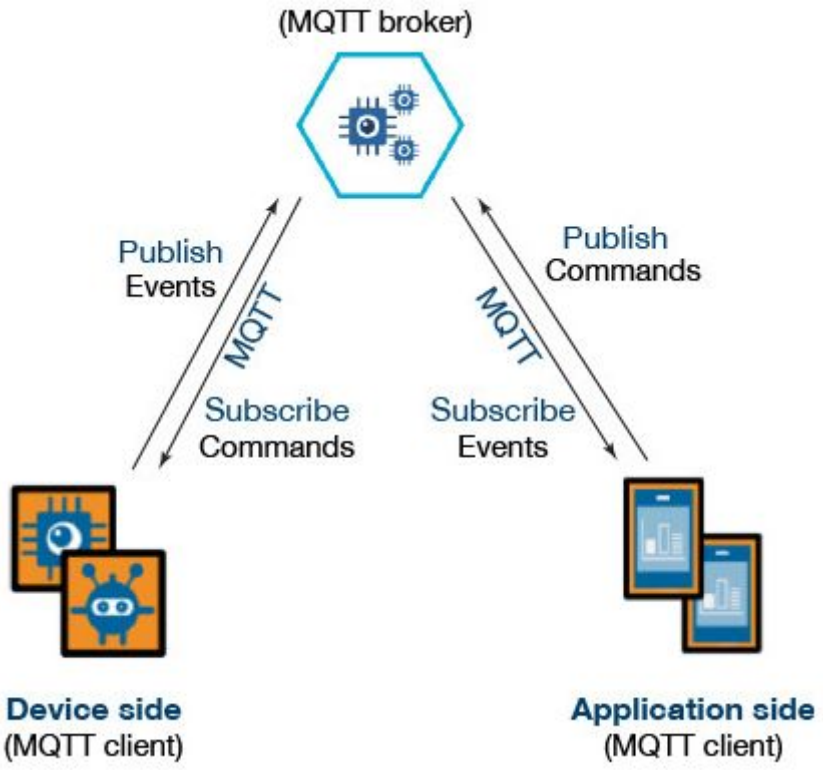


Fig 1: Architecture

This project has three parts:

1. The mobile app which is used to control the AC.
2. ESP8266(low-cost Wi-Fi module) which receives data from app and controls the AC through an IR blaster.
3. ESP8266 with IR receiver to record raw IR signals from remote.

Fig 2. ESP8266 box

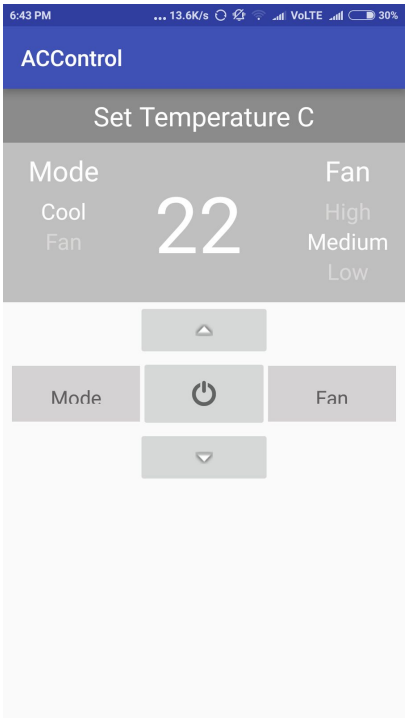


Fig 3. Application to control AC

The mobile app and the ESP8266 are connected to the internet via wifi. They don't have to be in the same wifi network to work. When we turn on the ESP8266 for the first time, it does not have the credentials to connect to any wifi AP, so instead of connecting to wifi, it turns into a wifi hotspot and allows the other devices to connect to it. The wifi network it creates is open and anyone can connect to it. When we connect to this wifi, we can go the the web page hosted in the ESP which displays the list of wifi networks available and options to enter username and password of a wifi. After saving the credentials and restarting the ESP, it connects to your desired wifi network.

The app and ESP communicate via MQTT. The MQTT server is on cloudmqtt.com which allows 10 connections for free(enough for testing). This is done so that we can test if ESP and app are able to communicate properly even if they are not in the same network. This cannot be done on IITB network as the port that is required to connect to cloudmqtt is blocked.

Recording IR Codes:

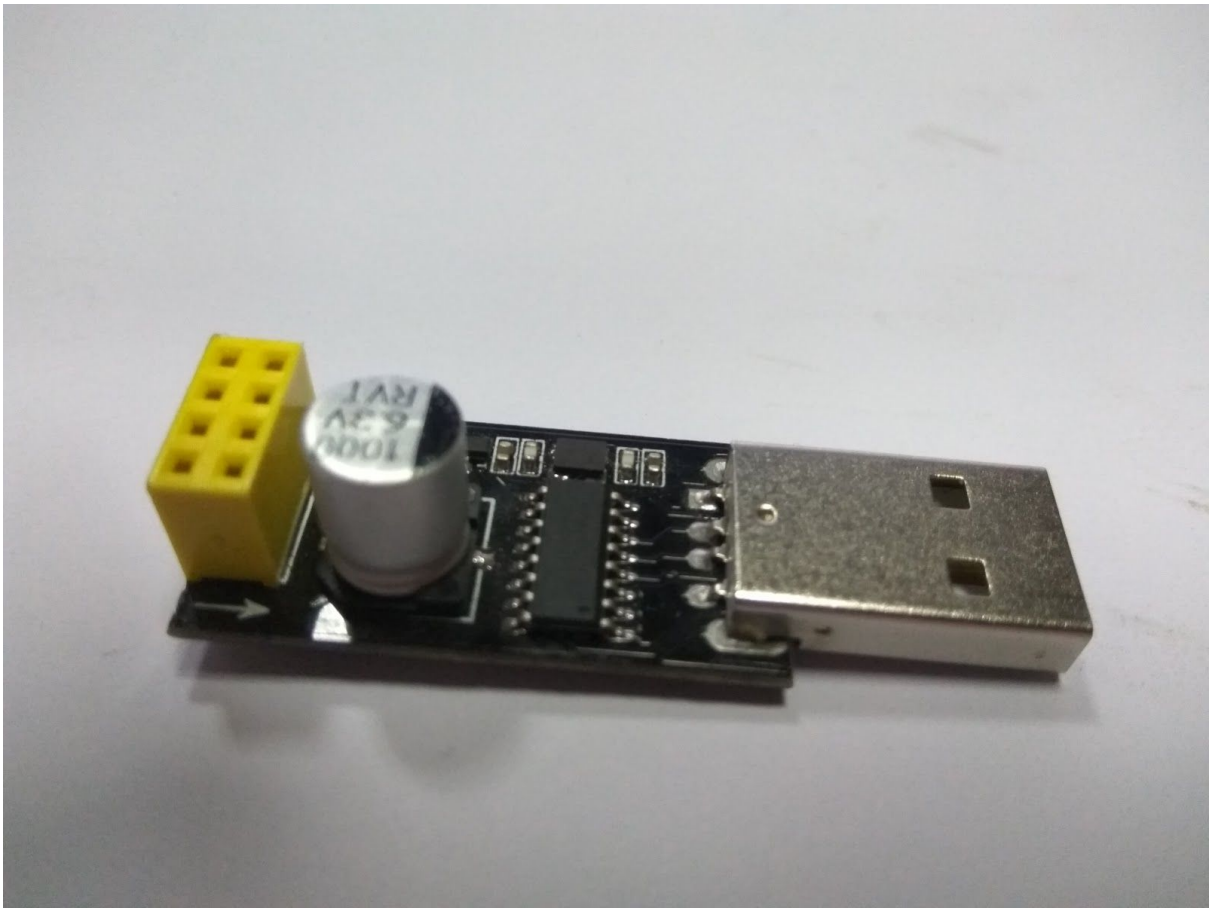
To record IR codes from a remote, use the connect IR receiver(TSOP1738) to ESP and run the code the the file IRRecord.ino.

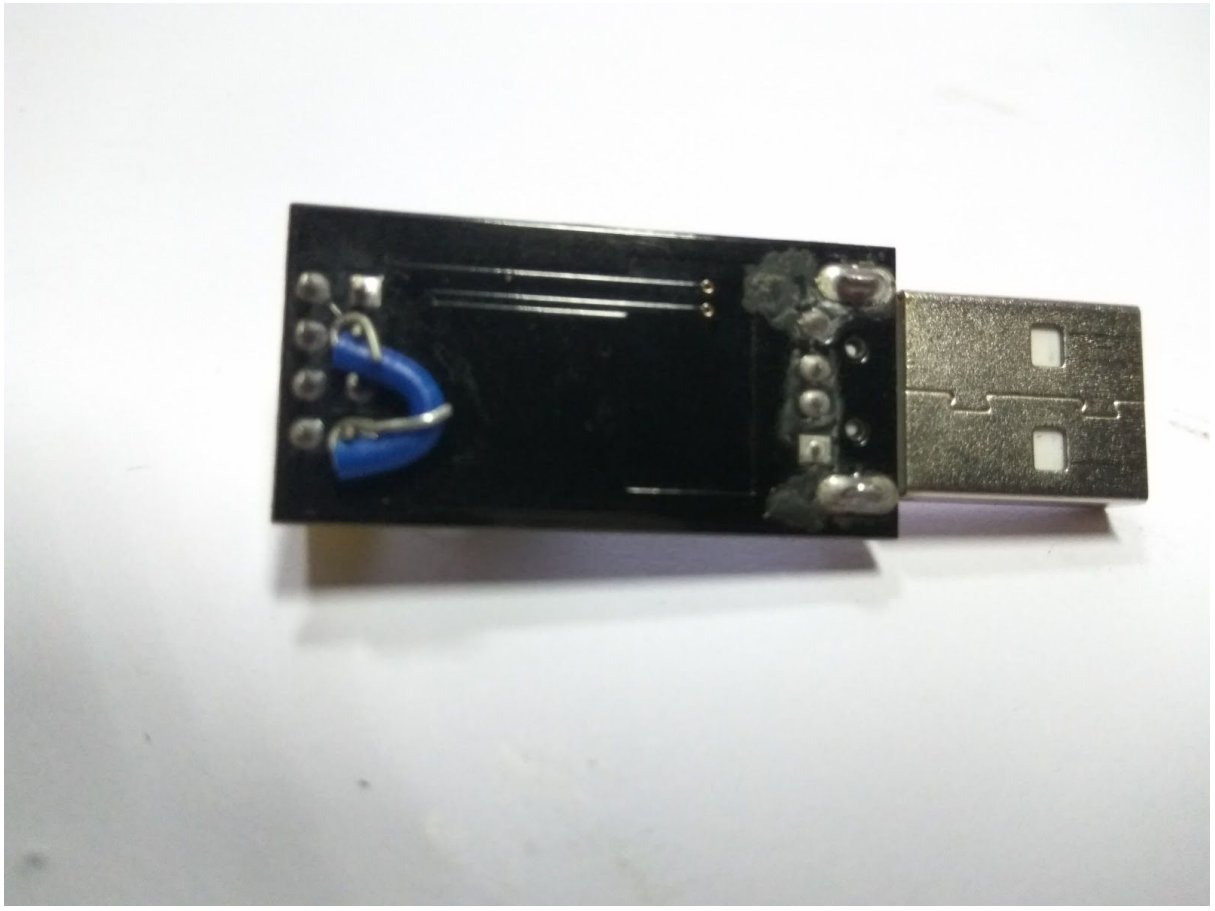
Programing ESP0/1:

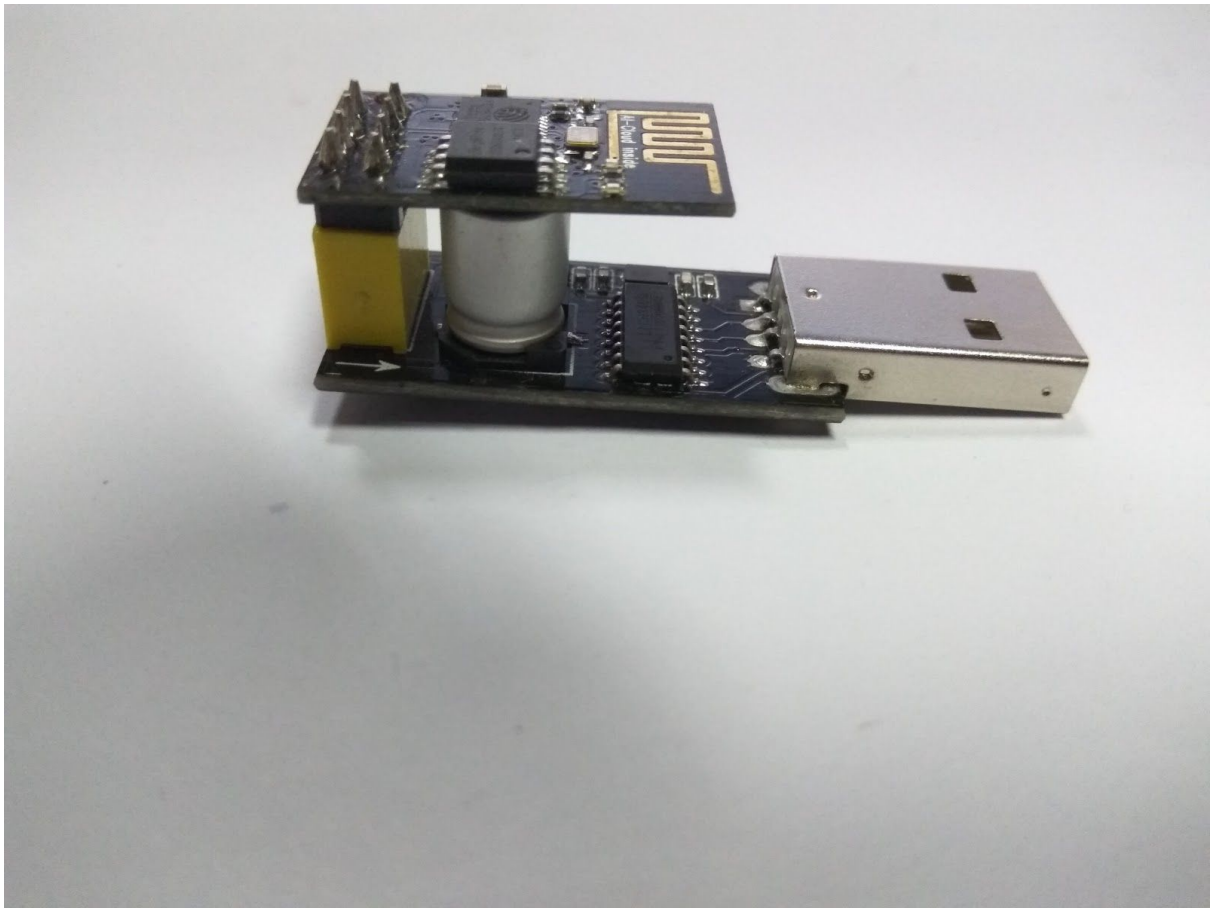
Programming ESP01 can be a difficult job if you are new to it, even if you have a esp programmer. So I am going to show you how I hacked the ESP programmer to make the programming easy.

GPIO-0 needs to be connected to GND in order to program the board. However, during the normal use, it must be disconnected.

If you are using a ESP programmer then you can either use a breadboard to ground GPIO0 or hardwire it directly into the ESP programmer. Using the breadboard method is time consuming and tedious. The easier method is to connect the ground and the GPIO0 through a wire on the programmer as shown in the image(Fig 5). The way in which I connected the wire shown below was only for testing purpose, you should do it properly so that is does not touch any other pin.







Note:- Use GPIO0 carefully. When ESP01 is booting and GPIO0 is connected to the ground, it enters the flash mode. This also happens if you connect an led between GPIO0 and ground.

Dependencies:

1. The android app uses Paho MQTT library to send MQTT messages to the ESP.
2. ESP uses PubSubClient library for receiving/sending MQTT messages.
3. MQTT server is on cloudmqtt.

Hardware Used:

1. ESP8266
2. ESP01
3. Tsop1738
4. IR blaster