Controlling Air Conditioner using MQTT Cloud

# OBJECTIVE:

This project controls the Air Conditioner remotely; the solution is to cool down a place even at a long distance with any phone using MQTT.

# Components Required :

|  |  |
| --- | --- |
| Hardware | Uses: |
| * NodeMcu(ESP8266) | Integrating Cloud Mqtt With user |
| * IR Transmitter | Transmitting IR pulse for control |
| * BC547 | Increasing the range of IR |
| * Power Supply Module | Regulating supply for this circuit |

|  |  |
| --- | --- |
| Software | Uses: |
| * Arduino IDE | Programming for Esp8266 |
| * MQTT Server | Cloud |
| * MQTT Dashboard | APP for User interface |
| * HeatPump IR Library | IR Libraries compatible to ESP8266 |

# Code:

|  |  |
| --- | --- |
| Name | Description |
| IR\_SEND\_PIN | The PIN where the IR LED is connected |
| WIFI\_SSID | Your WiFi network name |
| WIFI\_PASS | WiFi password |
| MQTT\_HOST | Mqtt hostname like mqtt.mydomain.com |
| MQTT\_USER | MQTT Username |
| MQTT\_PASS | MQTT Password |

# Working:

Basic structure:

Connection to the WiFi

Connection to the MQTT Server

Emitting IR commands to the A/C

# Steps:

Arduino Configuration:

1. Start New project in Arduino.
2. Link for that program: [https://www.sysrun.io/wp-content/uploads/2017/02/daikin-remote-clean-1.zip](https://www.sysrun.io/wp-content/uploads/2017/02/daikin-remote-clean-1.zip%20)
3. With Software uploaded(Aduino,MQTT Dashboard), Hardware connected(NodeMCU,IR, Power supply board )and MQTT connection enabled.
4. Add library By going to Arduino- >Sketch- >Include Library - >Add zip library->import respective library

1. PubsubClient,

2. HeatPumpIR

3. ESP8266WiFi

4.Timer

5. ESP8266mDNS

6. DNSServer.h

7.ESP8266WebServer.h

8.WiFiManager.h

Access Point Configuration:

1. when your ESP starts up, it sets it up in Station mode and tries to connect to a previously saved Access Point
2. if this is unsuccessful (or no previous network saved) it moves the ESP into Access Point mode and spins up a DNS and WebServer (default ip 192.168.4.1)
3. using any wifi enabled device with a browser (computer, phone, tablet) connect to the newly created Access Point
4. because of the Captive Portal and the DNS server you will either get a 'Join to network' type of popup or get any domain you try to access redirected to the configuration portal
5. choose one of the access points scanned, enter password, click save
6. ESP will try to connect. If successful, it relinquishes control back to your app. If not, reconnect to AP and reconfigure.

MQTT Configuration:

1. Login with CloudMQTT

Create new instances->choose name,plan,region ->submit

Goto your new instances –it have Server,username,Password,port(add to program)

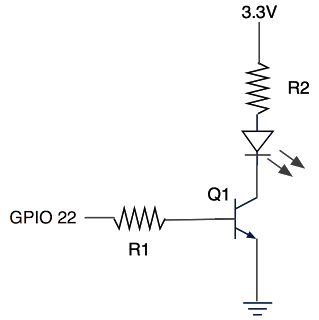
1. The software will listen for commands on a specific topic configured via the constant MQTT\_COMMAND\_CHANNEL.
2. Download MQTT Dashboard (ios,Android)
3. In MQTT app we have to link with cloudMQTT by giving servername,Username and passwords
4. The default configuration is “cmnd/daikin/# .
5. To set the power on the A/C, you just publish “1” to the topic “cmnd/daikin/power“.
6. The MCU will now send the command via IR
7. The current values of every setting (power, mode, fan etc.) is obtained as JSON to the topic “stat/daikin/”

{"power":1,"mode":3, "fan":2,"temp":23,"vswing":0,"hswing":0}

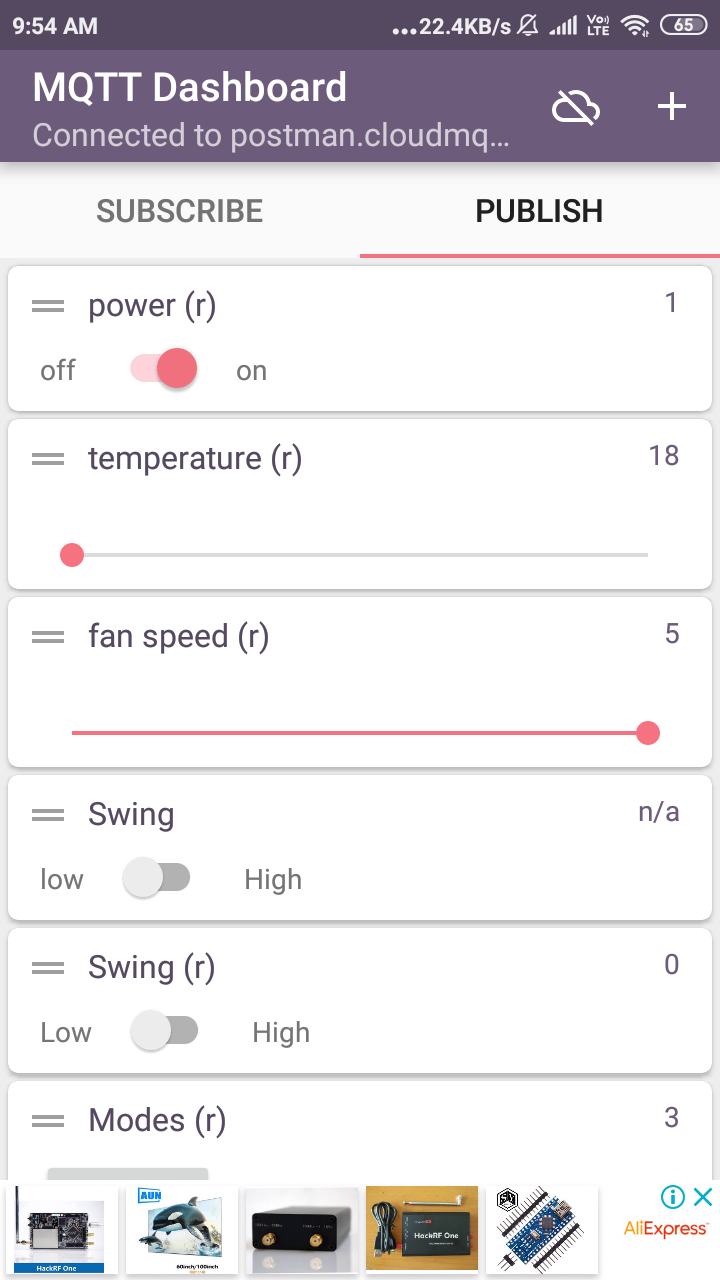
# Command overview and value mapping:

* Command Descr → Value
* cmnd/daikin/power POWER\_OFF → 0 POWER\_ON → 1
* cmnd/daikin/mode MODE\_AUTO → 1 MODE\_HEAT → 2 MODE\_COOL → 3 MODE\_DRY → 4 MODE\_FAN → 5
* cmnd/daikin/fan FAN\_AUTO → 0 FAN\_1 → 1 FAN\_2 → 2 FAN\_3 → 3 FAN\_4 → 4 FAN\_5 → 5
* cmnd/daikin/temp Numeric value VDIR\_AUTO → 0 HDIR\_AUTO → 0

# Circuit Diagram:



# App Interface:



# Hardware :

