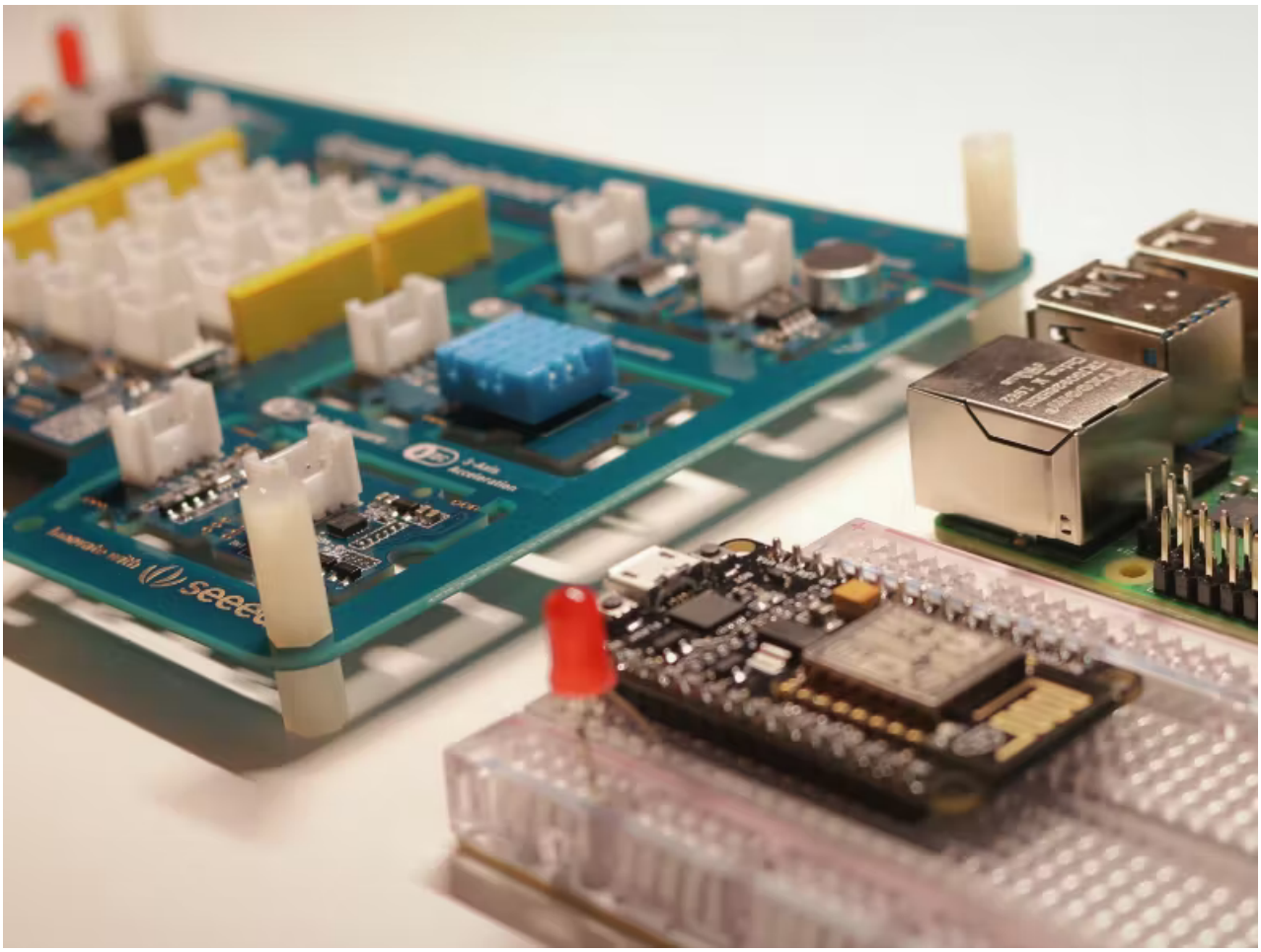
Smart Home IoT System Based on Raspberry Pi 4

hackster.io/SeeedStudio/smart-home-iot-system-based-on-raspberry-pi-4-a64c0d

Seeed



Things used in this project

Hardware components

Raspberry Pi 4B	× 1
NodeMCU v2 - Lua based ESP8266 development kit	× 1
Grove Beginner Kit for Arduino	× 1

Software apps and online services

Arduino IDE

Story

Thanks to the original writers **Huang Hao** and **Wu Shu** for their great support for this article.

1. Installing system for Raspberry with integrated Home Assistant
First of all, we need to flash the system into the SD card of Raspberry Pi. A system image with an integrated Home Assistant is chosen in this tutorial which is convenient for the installation.

Download from here: https://drive.google.com/drive/folders/1XE-JXsdtrfDiireDyo32zDYrqbWPdlFB?usp=sharing

After that, flash it into an 8G or above sd card using etcher.

1 / 2
If you are the first time use the raspberry pi, I suggest you follow the raspberry pi wiki at first.

In the system32 partition of the burned card, add the file wpa_supplicant.conf with the following content (modify the SSID and PSK). After the completion, insert the Raspberry Pi to start.

```
country=CN
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="Connected wifi name"
psk="wifi password"
key_mgmt=WPA-PSK
priority=1
}
```

Then boot the Pi and connect it to WiFi after system initialization. We can see a folder named *homeassistant* under /home/pi and lots of files in the folder in the terminal. That is to say, we don't need to install it manually.

Find the IP address of Pi using *ipconfig*.

Access Home Assistant web page using this address and port number 8123. In this tutorial, we use *192.168.123.116:8123*

You need some preparations such as creating an account the first time you access it. After that, all devices sharing the same local area network are able to access the home assistant control panel installed in the Raspberry Pi using a browser with this IP.

2. Controlling Lamp using Home Assistant
We can run Home Assistant but devices can never be controlled without being connected. So let's try connecting a Mi Lamp on my desk and check out if we can control it.

Open *Yeelight* on our mobile phone, and set desk lamp able to be controlled via LAN. Obtain the current IP address of the desk lamp in the device information.

Open the file */home/pi/homeassistant/configuration.yaml* using nano in the terminal.

Add the text to the end of the file:

```
yeelight:
devices:
192.168.123.182:
name: MI Light
```

Reboot Home Assistant using:

```
sudo systemctl stop home-assistant@pi
sudo systemctl start home-assistant@pi
```

After that, we can see a light controller on the home page and we can use it to turn on/off our lamp.

3. Controlling I/O ports of ESP8266 using Home Assistant
How can Home Assistant work with DIY devices? The answer is **MQTT**.

- Message Queuing Telemetry Transport, or MQTT, is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. An increasing number of devices and nodes are using this protocol to communicate with each other.

We highly recommend looking through the document on HACHINA website to get more information about MQTT: https://www.hachina.io/docs/7125.html

MQTT service has already been integrated into our system and the only thing that you need to do is add some code to the configuration file of Home Assistant.

NOTE that does not forget a space behind the colon or errors that may occur.

```
mqtt:
broker: 127.0.0.1
```

After reboot service, we can see a tab named MQTT in Developer Tools in the web UI.

We enter *ha/switch1* to monitor the topic in the *Listen to a topic* panel and enter the same topic name to *Publish a packet* panel. Click *PUBLISH* after entering *ON* to *Payload* and we can see the message we sent just now in the monitor window below. It means the MQTT service is available.

Next, let's flash the program to our ESP8266 board in order to monitor the same topic (ha/switch1) and turn ON/OFF the LED lamp via I/O port according to the Payload.

But before that, we have to change the *SSID* and *password* into our own WiFi name and password. Set mqtt_server as IP address of Raspberry Pi

After uploading, open serial monitor and you can see the command will be updated once we send an ON/OFF. The LED lamp will be turned ON/OFF simultaneously.

In order to add a button to control the LED in Home Assistant, we need to add some code to the file named *configuration.yaml*.

```
switch:
- platform: mqtt
state_topic: "ha/switch1"
command_topic: "ha/switch1"
payload_on: "ON"
payload_off: "OFF"
```

After rebooting service, we can see a button named MQTT Switch has been added to the homepage.

We can also use this I/O port to control some relays and turn ON/OFF some other devices such as lights or fans.

It's not difficult to control an air conditioner or television using infrared.

We use the infrared receiver to read data from remote control just after we press the power button. Once an ON has been received, the infrared transmitter sends the data and the air conditioner will be turned on.

```
uint16_t OFF[101] = {5832, 7344,  514, 3426,  516, 3426,  516, 3428,  514, 3428,
518, 1374,  514, 3430,  512, 3428,  514, 3428,  518, 1374,  518, 1374,  516, 1376,
516, 1376,  518, 3424,  514, 1378,  518, 1374,  518, 1374,  520, 1372,  514, 3428,
516, 3426,  518, 3424,  518, 1374,  518, 1376,  514, 1378,  516, 3426,  518, 3424,
516, 1376,  514, 1376,  514, 1378,  516, 3426,  516, 3426,  518, 3426,  516, 1374,
514, 3428,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  518, 1374,  516, 1376,
518, 1376,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  518, 1374,  518, 3422,
518, 3424,  514, 3426,  516, 7196,  518};
uint16_t ON[101]= {5832, 7322,  540, 3426,  516, 3426,  510, 3430,  518, 3426,
516, 1376,  516, 3426,  516, 1376,  514, 1380,  516, 1374,  516, 1376,  516, 1376,
514, 1376,  514, 3428,  516, 1378,  512, 3430,  516, 3424,  518, 1376,  514, 3428,
518, 3424,  516, 1378,  516, 1374,  512, 3428,  518, 3424,  516, 1378,  514, 3426,
514, 1378,  518, 1374,  514, 3426,  518, 3424,  514, 1376,  516, 1378,  514, 3426,
518, 3424,  518, 1374,  516, 3428,  514, 1378,  516, 3424,  516, 1378,  516, 1376,
516, 1376,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  516, 1376,  516, 3424,
516, 3426,  518, 3426,  516, 7194,  518};
```

DHT11 can be used to get temperature and humidity of the indoor environment and send data back to Home Assistant. Once the value exceeds the threshold, the air conditioner will be truned automatically.

4. Controlling Home Assistant devices with voice assistant

You can control Home Assistant devices with Tmall Elf. Just say "Hey Tmall Elf, turn on the light!"

Give the device you want to add to Tmall Elf a friendly name and add *tmall_genie: true* to customize.yaml.

For more information, you can also have a look at this essay: https://bbs.hassbian.com/thread-6795-1-1.html and add this smart hardware in theTmall Elf app and control it with voice!

Read more

Code

ESP8266 control air conditioner switch machine program

C/C++

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>

//ir
const uint16_t kIrLed = 4;  // ESP8266 GPIO pin to use. Recommended: 4 (D2).

IRsend irsend(kIrLed);  // Set the GPIO to be used to sending the message.
uint16_t OFF[101] = {5832, 7344,  514, 3426,  516, 3426,  516, 3428,  514, 3428,
518, 1374,  514, 3430,  512, 3428,  514, 3428,  518, 1374,  518, 1374,  516, 1376,
516, 1376,  518, 3424,  514, 1378,  518, 1374,  518, 1374,  520, 1372,  514, 3428,
516, 3426,  518, 3424,  518, 1374,  518, 1376,  514, 1378,  516, 3426,  518, 3424,
516, 1376,  514, 1376,  514, 1378,  516, 3426,  516, 3426,  518, 3426,  516, 1374,
514, 3428,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  518, 1374,  516, 1376,
518, 1376,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  518, 1374,  518, 3422,
518, 3424,  514, 3426,  516, 7196,  518};  // UNKNOWN FB60AF8D
uint16_t ON[101]= {5832, 7322,  540, 3426,  516, 3426,  510, 3430,  518, 3426,
516, 1376,  516, 3426,  516, 1376,  514, 1380,  516, 1374,  516, 1376,  516, 1376,
514, 1376,  514, 3428,  516, 1378,  512, 3430,  516, 3424,  518, 1376,  514, 3428,
518, 3424,  516, 1378,  516, 1374,  512, 3428,  518, 3424,  516, 1378,  514, 3426,
514, 1378,  518, 1374,  514, 3426,  518, 3424,  514, 1376,  516, 1378,  514, 3426,
518, 3424,  518, 1374,  516, 3428,  514, 1378,  516, 3424,  516, 1378,  516, 1376,
516, 1376,  518, 1374,  516, 3426,  516, 1376,  516, 3426,  516, 1376,  516, 3424,
516, 3426,  518, 3426,  516, 7194,  518};  // UNKNOWN 4C3D7CCD



// Update these with values suitable for your network.
const char* ssid = "xxx";
const char* password = "xxx";
const char* mqtt_server = "xxx";//youe raspberry pi ip address

WiFiClient espClient;
PubSubClient client(espClient);
int SwitchedPin = 0;
String switch1;
String strTopic;
String strPayload;

void setup_wifi() {
 Serial.begin(115200);
 irsend.begin();
  delay(100);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  payload[length] = '\0';
  strTopic = String((char*)topic);
  if(strTopic == "ha/switch1")
    {
    switch1 = String((char*)payload);
    if(switch1 == "ON")
      {
        Serial.println("ON");
        digitalWrite(SwitchedPin, HIGH);
        irsend.sendRaw(ON, 101, 38);  // Send a raw data capture at 38kHz.
        Serial.println("ON");

      }
    else
      {
        Serial.println("OFF");
        digitalWrite(SwitchedPin, LOW);
        irsend.sendRaw(OFF, 101, 38);  // Send a raw data capture at 38kHz.
        Serial.println("OFF");

      }
    }
}


void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.subscribe("ha/#");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void setup()
{
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  pinMode(SwitchedPin, OUTPUT);
```

```
  digitalWrite(SwitchedPin, LOW);
}

void loop()
{
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

Home Assistant configuration.yaml

YAML

```
mqtt:
  broker: 127.0.0.1

switch:
  - platform: mqtt
    state_topic: "ha/switch1"
    command_topic: "ha/switch1"
    payload_on: "ON"
    payload_off: "OFF"

tts:
  - platform: google_translate
    language: 'zh-cn'

media_player:
  - platform: vlc

yeelight:
  devices:
    192.168.123.182:
      name: MI Light


tunnel2local:
 # frpc命令位置
  frpc_bin: "/home/pi/bin/frpc"


group: !include groups.yaml
automation: !include automations.yaml
script: !include scripts.yaml
```