

Home (/) / Microcontroller Based Projects (/microcontroller-projects)

/ Never Loose your keys again with this IoT Based Smart Key Finder

## Never Loose your keys again with this IoT Based Smart Key Finder

Published June 10, 2021

3

A Ashish Choudhary (/users/ashish-choudhary)  
Author



IoT Based Smart Key Finder

Agree it!!! It has happened to all of us. Many times we misplace our keys and go searching for them everywhere in the house, and after a long search, we end up finding them with much distress. Now, the obvious solution here is to place your keys in their right place, but as engineers, what's the fun in doing that. So, in this tutorial, we are going to build a simple IoT-based **Smart Key Chain** just using ESP8266-01, Buzzer, and Battery. Now in case if you can't find your keys and you remember that you have attached an **IoT keychain** to your keys, so you take out your phone and open Chrome and open your Keychain Webpage. Then you click on the toggle button, and in moments, you hear a beep sound coming from your keychain and with this, you can easily track your keys.

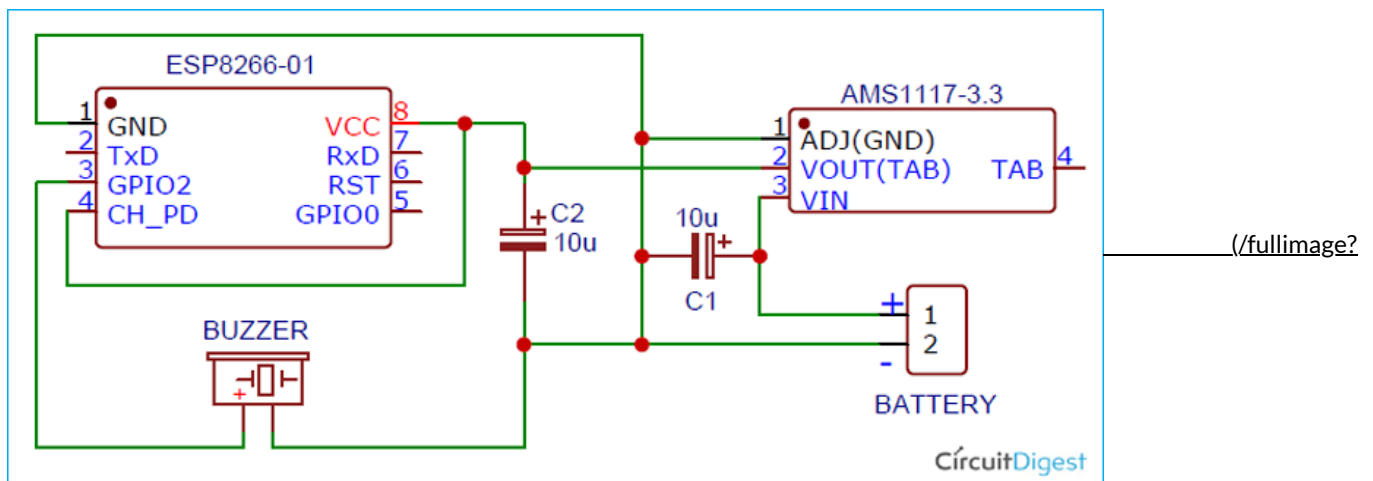
As you can see from the image, we have built this circuit on PCB and we have used [PCBONLINE](https://www.pcbonline.com/) (<https://www.pcbonline.com/>) to fabricate the PCB boards for this project. PCBONLINE is a PCB manufacturing and assembly service provider who has been in the industry since 1999. They can handle advanced PCBs with upto 24 layers adhering to ISO standards. The best part is that we can get high-quality industrial standard PCBs even for low volume orders and can also use their free PCB assembly and functional test services when ordering in bulk. We will explain more on how to use pcbonline later in this article , but for now, lets discuss about the design and working of our IoT based Key chain.

## Components Required

- ESP8266-01
- AMS1117 3.3V Voltage Regulator
- Buzzer
- Lithium Polymer battery
- 2× 10µf Capacitor

## Smart key Finder Circuit Diagram

The complete circuit diagram for the **Esp8266 smart key chain** is shown below. The schematic was drawn using EasyEDA.



[i=circuitdiagram\\_mic/Smart-key-Finder-Circuit-Diagram.png](#)

This complete setup will be powered by the Lithium polymer Battery and AMS1117-3.3V is used to regulate 3.3V for the ESP8266-01 board. VCC and CH\_PD pins of ESP8266-01 are connected to the output pin of AMS1117 while we connect the GND pin to the -ve rail of the battery. The positive terminal of Buzzer is connected to the GPIO2 pin of ESP8266 while the negative terminal of the buzzer is connected to the GND of ESP8266-01. Ok, so we are done with the connections. Now let's have a look at the PCB Design.

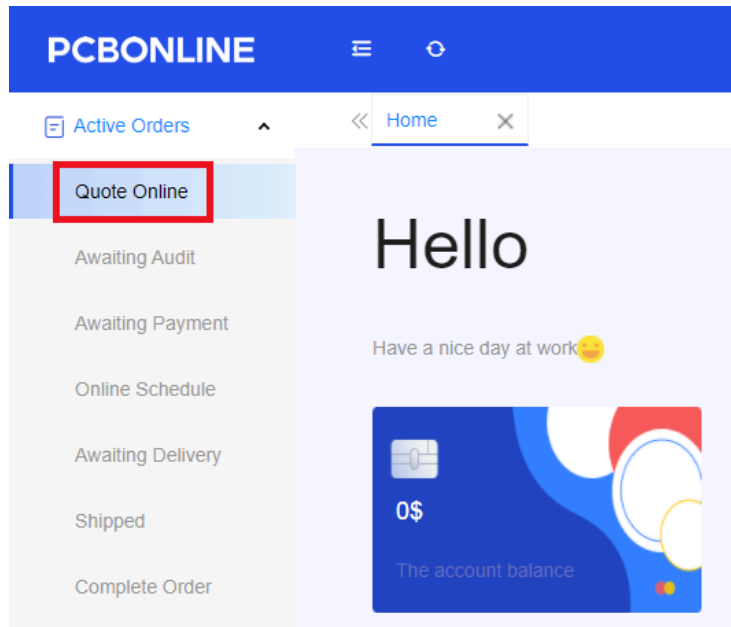
## Fabricating our Smart KeyChain using PCB Online

Now that we understand how the schematic works, we can proceed with building the PCB for our project. You can design the PCB using any PCB software of your choice. If you are a beginner who is just getting started in the world of PCB, we would recommend to try [getting started with eagle for PCB Designing](https://pcbdesignworld.com/article/eagle-pcb-design-software-getting-started) (<https://pcbdesignworld.com/article/eagle-pcb-design-software-getting-started>). If you want to skip the design processes, you can also **download the Gerber file** of this smart key finder project using the below link.

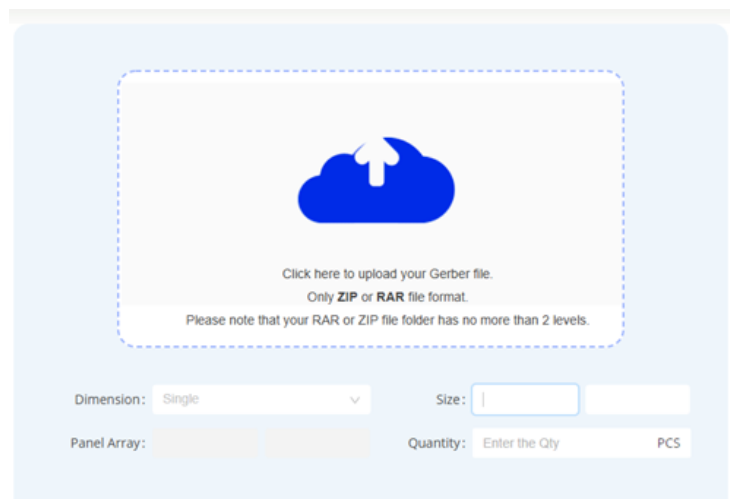
**Gerber File Link** ([https://circuitdigest.com/sites/default/files/ESP8266%20Key-Chain Gerber File.zip](https://circuitdigest.com/sites/default/files/ESP8266%20Key-Chain%20Gerber%20File.zip)).

You can also check out our other [PCB projects \(https://circuitdigest.com/diy-pcb-projects\)](https://circuitdigest.com/diy-pcb-projects) that we have previously built on circuit digest if you are interested. Now that our Design is ready, it is time to get them fabricated using the Gerber file. **Getting your PCB fabricated from PCBONLINE (https://www.pcbonline.com/)** is easy; simply follow the steps given below.

Go to <https://www.pcbonline.com/> sign up if this is your first time. Then, go to the Active Orders tab and then click on 'Quote Online'.



Now in the next window, enter the Gerber file, dimensions of your PCB, the number of layers, and the number of PCB you require.



After entering the Gerber file, you can set a few additional parameters if required, like the material used track spacing, etc. but usually, the default values work fine. In the next step, you have to select the Build time, Shipping country.

Your files have been successfully uploaded. [Back to Upload File](#)

### PCB Specification

**standard** **special**

|                    |                |                    |             |
|--------------------|----------------|--------------------|-------------|
| Material:          | FR4            | Thickness:         | 1.6mm       |
| TG(°C):            | 135            | CTI:               | 175sCTI<250 |
| Layer:             | 2layer         | Hole Copper:       | 20um        |
| Solder Mask:       | green          | Silkscreen:        | white       |
| Inner Copper:      | none           | Outer copper:      | 1oz         |
| Min Track/Spacing: | 5/5mil         | Min Hole Size:     | 0.3         |
| Surface Finish:    | HASL lead free | Surface Thickness: | 0.25-0.5um  |

Board Price \$38.76

Engineering Price \$0

Test Price \$0

### Shipping Cost

**DHL** India \$0

Currency select **38.76**

Order description

The final step is Checkout. To make sure the process is smooth; PCBONLINE first verifies all the details and Gerber file before proceeding with the payment. This way, you can be sure that your PCB is fabrication-friendly and will reach you as committed.

Currency select **38.76**

Order description

You can check your order status from the 'Active Orders' tab.

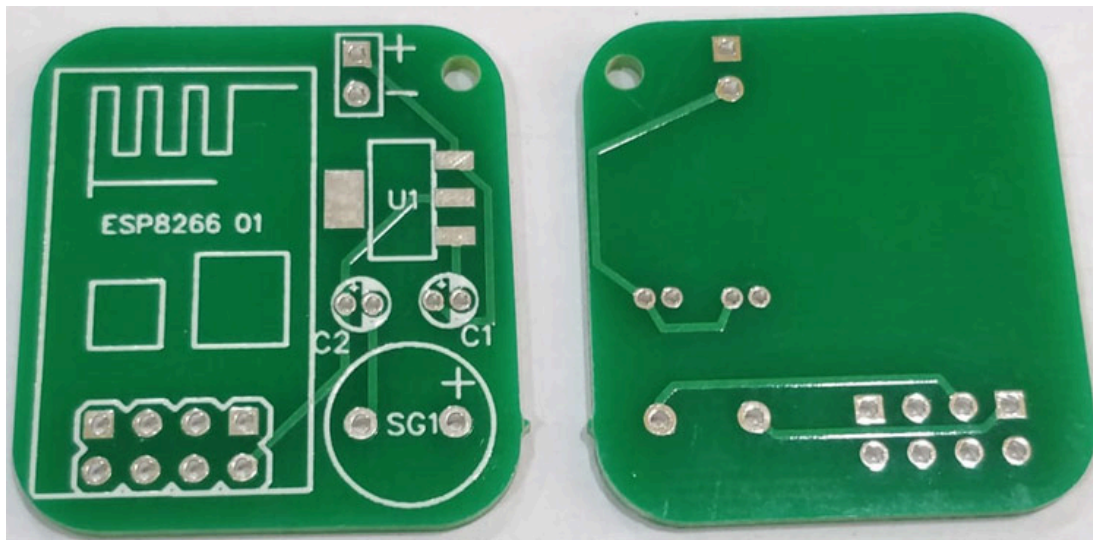
**Active Orders** > Home > Awaiting ... > Awaiting ...

| Quote Online          | Date  | PCB File     | Board Infor... | Type | QTY(PCS) | Price(\$) | Total(\$) | N.W(KG) | Build Time | Action |
|-----------------------|---|--------------|----------------|------|----------|-----------|-----------|---------|------------|--------|
| <b>Awaiting Audit</b> | 2021-05-27  | Gerber_PC... | 2L, 1.6mm...   | FR4  | 5        |           | 38.76     | 0       | 72hours    |        |
| Awaiting Payment      | Our engineering team is reading your request. Please wait for some time.  |              |                |      |          |           |           |         |            |        |
| Online Schedule       | If there's anything wrong with your request, we will contact you immediately. If everything's okay, our audition will complete in half an hour. |              |                |      |          |           |           |         |            |        |
| Awaiting Delivery     | When your order is audited, you can pay and track the production stage online.  |              |                |      |          |           |           |         |            |        |
| Shipped               |   |              |                |      |          |           |           |         |            |        |
| Complete Order        |   |              |                |      |          |           |           |         |            |        |
| Canceled              |   |              |                |      |          |           |           |         |            |        |

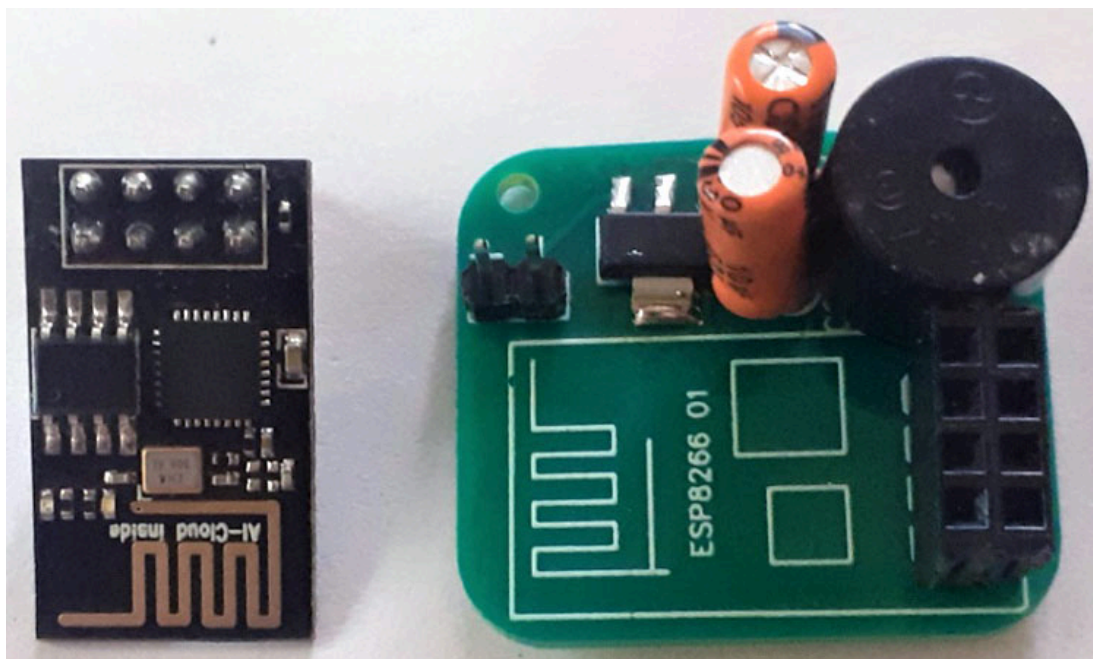
[Remind Audition](#)

## Assembling the Smart KeyChain Board

After a few days, we received our PCB from PCBONLINE in a neat package and as you can see below, the PCB quality was good as always. The top layer and bottom layer were seamlessly done with proper visa and track spacing. The top layer and the bottom layer of the board are shown below.



After making sure that the tracks and footprints were correct, I proceeded with assembling the PCB. The completely soldered board looked like as shown in the image below:



## Smart Key Chain Code Explanation

Complete code for IoT-based Smart Keychain can be found at the end of the page. Here we are explaining the complete code line by line.

So as usual, let's start the code by importing all the required libraries. All the libraries that we are going to use in this code come pre-installed with ESP8266 board files. ESP8266WiFi library is used to connect NodeMCU to a Wi-Fi network.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
```

Then create a webserver object that listens for HTTP request on port 80

```
ESP8266WebServer server(80);
```

Define the ESP8266 pin where Buzzer is connected

```
const int buz_pin = 2;
```

In the next stage, enter Wi-Fi credentials like the user name and password for the Wi-Fi router to which your NodeMCU should connect with.

```
char ssid[] = "Wi-Fi Name";
char pass[] = "Wi-Fi Password";
```

The string variable 'html\_code' contains a simple HTML code for creating a NodeMCU webpage.

```
String html_code =
"<!DOCTYPE html><html><head><style>.button {border: none;padding: 12px 40px;text-align: center;text-decoration: none;display: in
```

The **handleRoot()** function is executed when we open the Webpage in the browser using the NodeMCU IP address. It sends the current buzzer state and a web page with a toggle button to webserver.

```
void handleRoot() {
server.send(200, "text/html", html_code + "Current state: <b>" + buzzing_state);
}
```

**handleBUZ()** is used to change the buzzer state if the button is pressed on the webpage. It adds a header to respond with a new location for the browser to go to the home page again.

```
void handleBUZ() {
buzzing_state = !buzzing_state;
server.sendHeader("Location","/");
server.send(303);
}
```

Inside the **setup()** function, we initialized the baud rate, defined the buzzer pin as an output, and then connect the module with the Wi-Fi using the Wi-Fi name and password.

```
Serial.begin(115200);
delay(10);
pinMode(buz_pin, OUTPUT);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
}
Serial.println("OK!");
```

The first function is used to call the 'handleRoot' function when a client requests URI (Uniform Resource Identifier) "/" while the second function is used to call the 'handleBUZ' function when a POST request is made to URI "/handleBUZ"

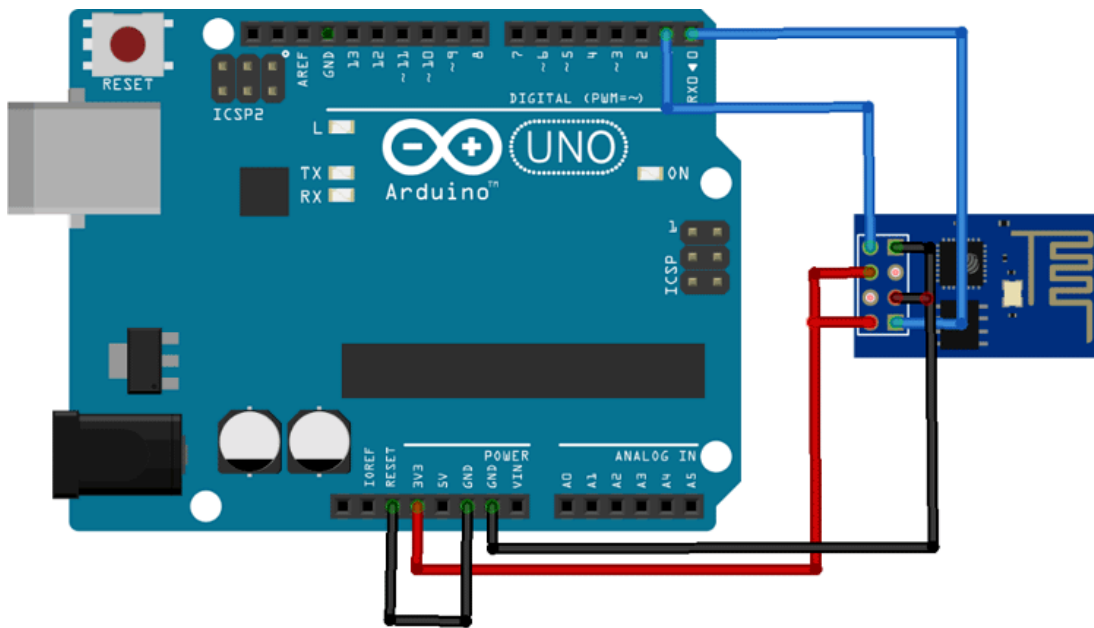
```
server.on("/", HTTP_GET, handleRoot);
server.on("/BUZ", HTTP_POST, handleBUZ);
```

Now in the next stage, we will read the buzzer state from the webpage and then change the GPIO pin state to turn on/off the buzzer.

```
void loop(void){
server.handleClient();
if (buzzing_state == true) {
digitalWrite(buz_pin, HIGH);
delay(400);
yield();
digitalWrite(buz_pin, LOW);
}
}
```

## Programming ESP8266-01 for Smart Key Chain

Instead of using the FTDI board here, I am using Arduino Uno to program the ESP8266-01. The circuit diagram for programming ESP8266-01 using Arduino is given below:



Connections for programming ESP8266 are as follows:

| ESP8266-01 | Arduino Uno   |
|------------|---|
| VCC        | 3.3V  |
| GND        | GND   |
| CH-PD      | 3.3V  |
| RX         | RX  |
| TX         | TX  |
| GPIO-0     | GND   |
| GPIO-2     | Not Connected   |
| RST        | Initially Not Connected. Before hitting upload connect RST to ground and remove after half a second |

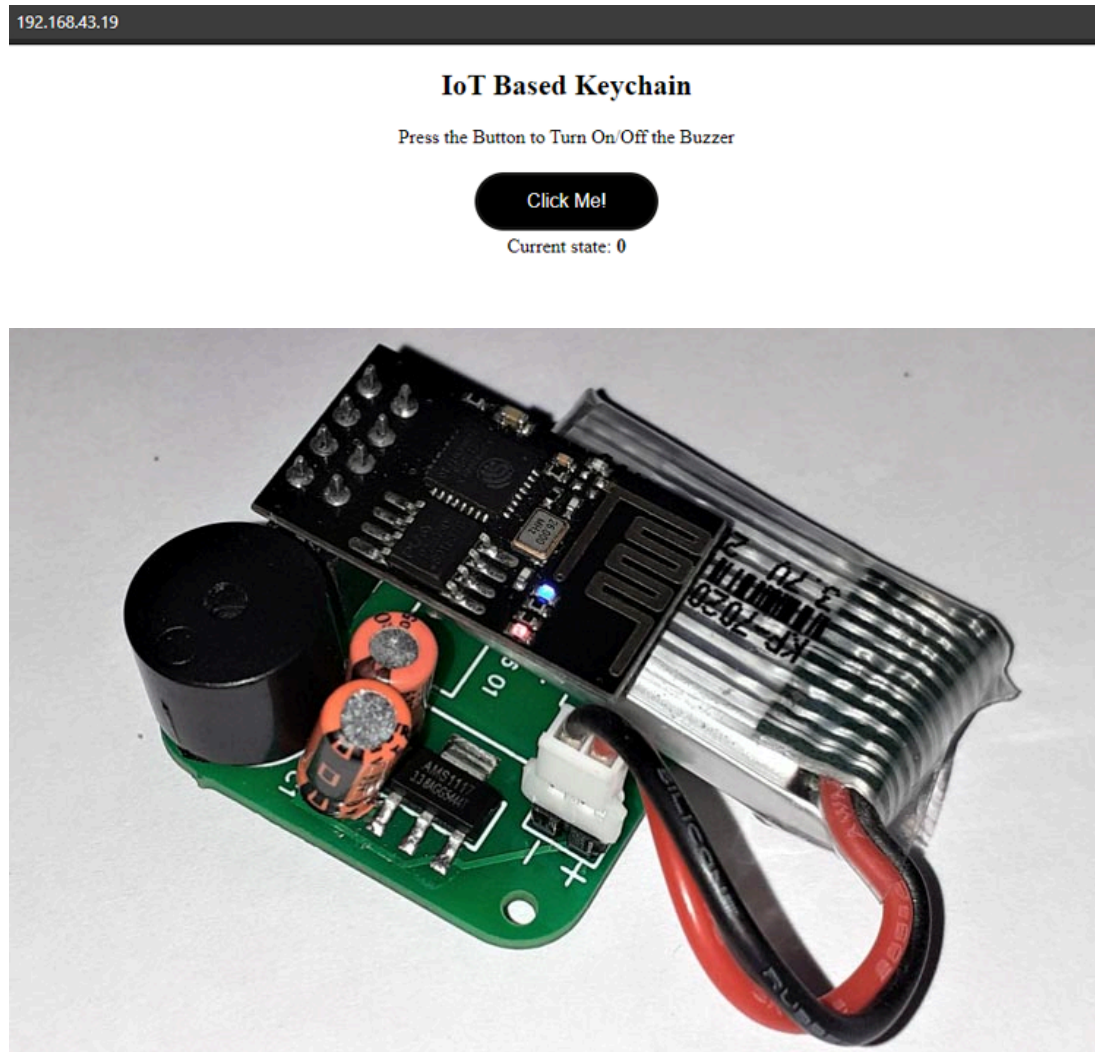
Apart from these connections, connect the Reset pin of Arduino to GND to bypass the Arduino. It will disable Arduino and upload code directly to the ESP8266 board. Now power up the Arduino Uno and open the Arduino IDE. Select the “Generic ESP8266 Module” in Board. Now before clicking on Upload, we have to boot ESP-01 into programming mode. Ground the RST



pin for a second. Now click on Upload in your Arduino IDE.

## Testing our Smart Key Chain Finder

Now that we have assembled the PCB and uploaded the code on ESP8266-01, we are ready to test the keychain. For that power, the setup and wait until ESP connects to Wi-Fi, then get the ESP IP address. Now open your favorite browser and type in the IP address and search. It should open a page as shown below:



Click on 'Click Me' and the Buzzer we will make a sound until you click on the button again. The current state of the buzzer is shown below the toggle button. The complete working video and code for this project are given below. I hope you enjoyed building this project. If you have any questions, please leave them in the comment section.

### Code

---



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
ESP8266WebServer server(80);
const int buz_pin = D2;
char ssid[] = "Galaxy-M20";
char pass[] = "ac312124";
String html_code = "<!DOCTYPE html><html><head><style>.button {border: none;padding: 12px 40px;text-align: center;text-decora
boolean buzzing_state = false;
void handleRoot() {
server.send(200, "text/html", html_code + "Current state: <b>" + buzzing_state);
}
void handleBUZ() {
buzzing_state = !buzzing_state;
server.sendHeader("Location","/");
server.send(303);
}
void handleNotFound(){
server.send(404, "text/plain", "404: Not found");
}
void setup(void){
  Serial.begin(115200);
  delay(10);
  pinMode(buz_pin, OUTPUT);
  Serial.print("\n\nConnecting Wifi... ");
  WiFi.begin(ssid, pass);
```

## Video

### DIY IoT Based Smart Keychain Finder using ESP8266



## Tags

[IoT \(/Tags/Iot\)](#) [ESP8266 \(/Tags/Esp8266\)](#) [PCB \(/Tags/Pcb\)](#)

## Have any question realated to this Article?



WHATSAPP

Ask Our Community Members

<https://chat.whatsapp.com/JR4e0Fc0V20H6jTMDd1daT>

TELEGRAM



DISCORD



FORUM

<https://t.me/Circuitdigest> <https://discord.com/invite/UXJrFJSWpz> <https://circuitdigest.com/forums>

## Comments

Submitted by Charles Malloch (/users/charles-malloch) on Mon, 06/14/2021 - 22:13

Permalink (/comment/34083#comment-34083)

### I would have thought that the (/comment/34083#comment-34083)

I would have thought that the ESP8266, especially when constantly connected to WiFi, would take a lot of power, and would run down a coin cell quickly. I'd love to hear that I was wrong!

How much battery life do you get with this setup?

Submitted by Visvesh kumar (/users/visveshyallamelli89787) on Sat, 01/22/2022 - 10:33

Permalink (/comment/34844#comment-34844)

### can you upload BOM and CPL (/comment/34844#comment-34844)

can you upload BOM and CPL files along with the Gerber file, please?

Submitted by ping (/users/ping) on Mon, 08/14/2023 - 18:12

Permalink (/comment/36170#comment-36170)

### Hello, if it is taken... (/comment/36170#comment-36170)

Hello, if it is taken outdoors, it will leave the network connection. If this is the case, or in another house, will it also lose the connection to the network? If there is someone else's network there, eap-01 will also be Unable to connect, because the code has set itself char ssid[] = "own network name";

char pass[] = "My network password"; This is the name and password of the network known to my home. I hope to see if there is any way, if there is an AP mode, the esp-01 spontaneous method can be connected to any other external network area, as long as the smart phone correspondingly enters the name and password, it can be used (of course, this is another code, enabled for AP mode)

Log in (/user/login?destination=/microcontroller-projects/diy-iot-based-key-chain-finder-using-esp8266%23comment-form) or register (/user/register?destination=/microcontroller-projects/diy-iot-based-key-chain-finder-using-esp8266%23comment-form) to post comments

**DigiKey****FEATURED PRODUCTS  
FROM DIGIKEY**

**ADP2230 Mixed Signal Oscilloscope (MSO) (<https://bit.ly/4cowLzQ>)**

Digilent's ADP2230 MSO allows users to measure, visualize, generate, record, and control circuits.

(<https://bit.ly/4cowLzQ>)

**Würth Elektronik EMC Products (<https://bit.ly/4eHQcoV>)**

The EMC product range includes PCB and Cable ferrites, inductors, capacitors, and more...

(<https://bit.ly/4eHQcoV>)

**UAM Wi-Fi Triple Band Antennas (<https://bit.ly/45IfvL>)**

TE Connectivity's UAM Wi-Fi Triple Band Antennas offer a compact alternative for terminal antennas

(<https://bit.ly/45IfvL>)

**G9KB-E Series High Power PCB DC Relay (<https://bit.ly/3R5kpUL>)**

Omron's G9KB series supports high current applications with high capacity load ratings

(<https://bit.ly/3R5kpUL>)

**Embedded and Microcontrollers (<https://bit.ly/3VUjEkP>)**

Master evaluation boards, microprocessors and other expertise at DigiKey!

(<https://bit.ly/3VUjEkP>)

**3M™ TwinAx High Speed Cable Solutions (<https://bit.ly/3xHSuUr>)**

3M™ TwinAx High Speed Cable Solutions: Thin, low profile cable with extremely tight bend radii

(<https://bit.ly/3xHSuUr>)



(<https://bit.ly/3KlhxPP>)

#### RAC10E-K/277 Series AC/DC Converters (<https://bit.ly/3KlhxPP>)

RECOM Power's 10 W AC/DC converters in a 1.8" x 1.0" case are ideal for EV charging applications



(<https://bit.ly/453b0D5>)

#### RM Series Aluminum Rack Cases (<https://bit.ly/453b0D5>)

Hammond's RM series aluminum cases are suitable for 19" rackmount or desktop/benchtop applications.



## Join 100K+ Subscribers

Your email is safe with us, we don't spam.

Type your email address

Subscribe

Be a part of our ever growing community.



Semicon Media is a unique collection of online media, focused purely on the Electronics Community across the globe. With a perfectly blended team of Engineers and Journalists, we demystify electronics and its related technologies by providing high value content to our readers.



(<https://www.facebook.com/circuitdigest/>)



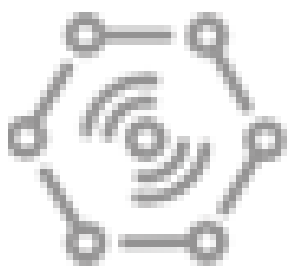
(<https://twitter.com/CircuitDigest>)



(<https://www.youtube.com/channel/UCy3CUAIYgZdAOG9k3IPdLmw>)



(<https://www.linkedin.com/company/circuit-digest/>)

**COMPANY**[Privacy Policy \(/privacy-policy\)](/privacy-policy/)[Cookie Policy \(/cookie-policy\)](/cookie-policy/)[Terms of Use \(/terms-of-use\)](/terms-of-use/)[Contact Us \(/contact\)](/contact/)[Advertise \(/advertise\)](/advertise/)**PROJECT**[555 Timer Circuits \(/555-timer-circuits\)](/555-timer-circuits/)[Op-amp Circuits \(/op-amp-circuits\)](/op-amp-circuits/)[Audio Circuits \(/audio-](/audio-circuits/)[circuits\)](#)[Power Supply Circuits \(/smps-power-supply-circuits\)](/smps-power-supply-circuits/)[Arduino Projects \(/arduino-projects\)](/arduino-projects/)[Raspberry Pi Projects \(/simple-raspberry-pi-projects-for-beginners\)](/simple-raspberry-pi-projects-for-beginners/)[MSP430 Projects \(/msp430-](/msp430-projects/)[projects\)](#)[STM32 Projects \(/stm32-projects-and-tutorials\)](/stm32-projects-and-tutorials/)[ESP8266 Projects \(/esp8266-projects\)](/esp8266-projects/)[PIC Projects \(/pic-microcontroller-projects\)](/pic-microcontroller-projects/)[AVR Projects \(/avr-microcontroller-projects\)](/avr-microcontroller-projects/)[8051 Projects](#)[\(/8051-microcontroller-projects\)](/8051-microcontroller-projects/)[ESP32 Projects \(/esp32-projects\)](/esp32-projects/)[\[iot-projects\\)\]\(#\)](/internet-of-things-</a></div><div data-bbox=)[PCB Projects \(/diy-pcb-projects\)](/diy-pcb-projects/)[\[projects\\)\]\(#\)](/arduino-esp8266-</a></div><div data-bbox=)[All Microcontroller Projects \(/microcontroller-projects\)](/microcontroller-projects/)**OUR NETWORK****CIRCUIT  
DIGEST** [\(https://circuitdigest.com\)](https://circuitdigest.com)**COMPONENTS<sup>101</sup>** [\(https://components101.com\)](https://components101.com)**IOT DESIGN  
PRO** [\(https://iotdesignpro.com\)](https://iotdesignpro.com)

Copyright © 2023 Circuit Digest (/). All rights reserved.