

[HOME](#) [BLOG](#) [PROJECTS](#) [ABOUT](#) [RSS](#)

JANUARY 29, 2018

Manually Setting Particle Wifi Creds

I recently created a number of [wifi touch lights](#) to help my family keep in “touch” with each other. It was a fun project, but there was a bit of a challenge getting everyone’s light set up. I was giving the lights as gifts, and I did not want my family members (some of them not as tech savvy) to have to go through the Photon wifi setup using a [web application](#) or [mobile app](#).

Luckily, Particle’s API provides a way to set wifi credentials for your Photon from within the source code of the flashed program. They provide great documentation describing this [API](#), however they don’t provide any tutorials for the steps necessary to set wifi credentials from within your code. Below I describe how to set up wifi credentials manually for your Particle Photon.

The first step is to place your photon in [SEMI_AUTOMATIC](#) mode. This must be done outside of the `setup` and `loop` functions.

```
SYSTEM_MODE(SEMI_AUTOMATIC);

setup() {
  // Setup code goes here
}

void loop() {
  // Do more stuff here
}
```

`SEMI_AUTOMATIC` mode does not immediately connect to the particle cloud, but waits until you manually call `Particle.connect()` to connect to the cloud. Once connected to the cloud, your particle automatically manages the connection.

Once your Photon is put into `SEMI_AUTOMATIC` mode, you can manually set up your wifi connection at the beginning of your `setup` function. You can specify up to five sets of credentials for Photons and seven sets of credentials for Cores. If more credentials are specified, the oldest credentials will be overwritten.

```
SYSTEM_MODE(SEMI_AUTOMATIC);

void setup() {
  WiFi.on();
  WiFi.disconnect();
  WiFi.clearCredentials();
  WiFi.setCredentials("SSID", "password", WPA2, WLAN_CIPHER_AES);
  WiFi.setCredentials("ALTERNATE_SSID", "other_password", WPA2, WLAN_CIPHER_AES);
  WiFi.connect();
  waitUntil(WiFi.ready);
  Particle.connect();

  // Rest of setup code goes here
}

void loop() {
  // Do more stuff here
}
```

You can find more details on `setCredentials` in the [Particle Reference Docs](#). The first argument is the name of the wifi access point/router. The second argument is the password to connect to the wifi access point. Both the third and fourth arguments are required if the access point is not accessible when the credentials are set.

The third argument specifies the authentication procedure. It must be one of `WPA2`, `WPA`, or `WEP`. `WPA2` is the most modern and secure authentication procedure, and `WEP` is the oldest. The fourth argument specifies the security cipher. Possible security cipher options are `WLAN_CIPHER_AES`, `WLAN_CIPHER_TKIP`, and `WLAN_CIPHER_AES_TKIP`. Of these `WLAN_CIPHER_AES` is the most modern and secure, and `WLAN_CIPHER_TKIP` is the oldest. `WLAN_CIPHER_AES_TKIP` is a mixed mode cipher configuration.

This gives you a working solution, but you can clean this code up a bit by placing the wifi setup code in its own function:

```
SYSTEM_MODE(SEMI_AUTOMATIC);

setup() {
```

```

    setupWifi();

    // Rest of setup code goes here
}

void loop() {
    // Do more stuff here
}

void setupWifi() {
    WiFi.on();
    WiFi.disconnect();
    WiFi.clearCredentials();
    WiFi.setCredentials("SSID", "password", WPA2, WLAN_CIPHER_AES);
    WiFi.setCredentials("ALTERNATE_SSID", "other_password", WPA2, WLAN_CIPHER_AES);
    WiFi.connect();
    waitUntil(WiFi.ready);
    Particle.connect();
}

```

This is a pretty good solution, but if you commit your code to a source code management system like Git (which you should), then this makes it hard to commit your code without committing your wifi credentials. This can be remedied by placing our credentials in a separate file:

```

/*
 * src/wifi_creds.h
 */

// See https://docs.particle.io/reference/firmware/photon/#setcredentials- for details
struct credentials { char *ssid; char *password; int authType; int cipher; };

const credentials wifiCreds[] = {
    // Set wifi creds here (last entry will be tried first when connecting)
    // {.ssid="SSID", .password="password", .authType=WPA2, .cipher=WLAN_CIPHER_AES}
};

```

```

/*
 * src/main.ino
 */

#include "wifi_creds.h"

SYSTEM_MODE(SEMI_AUTOMATIC);

setup() {
    setupWifi();

    // Rest of setup code goes here
}

void loop() {

```

```

    // Do more stuff here
}

void setupWifi() {
  WiFi.on();
  WiFi.disconnect();
  WiFi.clearCredentials();
  int numWifiCreds = sizeof(wifiCreds) / sizeof(*wifiCreds);
  for (int i = 0; i < numWifiCreds; i++) {
    credentials creds = wifiCreds[i];
    WiFi.setCredentials(creds.ssid, creds.password, creds.authType, creds.cipher);
  }
  WiFi.connect();
  waitUntil(WiFi.ready);
  Particle.connect();
}

```

You can commit `src/wifi_creds.h` to be used as a template, and then not commit this file when you have added credentials to the `wifiCreds` array.

What if you want to flash firmware onto your Photon, but don't want to overwrite the wifi credentials currently on your Photon? You can make one more tweak to your code to allow for updating the Photon's firmware without manually specifying wifi credentials:

```

/*
 * src/wifi_creds.h
 */

// Uncomment the line below if specifying credentials in this file
// #define WIFI_CREDENTIALS_SPECIFIED

// See https://docs.particle.io/reference/firmware/photon/#setcredentials- for details
struct credentials { char *ssid; char *password; int authType; int cipher; };

const credentials wifiCreds[] = {
  // Set wifi creds here (last entry will be tried first when connecting)
  // {.ssid="SSID", .password="password", .authType=WPA2, .cipher=WLAN_CIPHER_AES}
};

```

```

/*
 * src/main.ino
 */

#include "wifi_creds.h"

#ifdef WIFI_CREDENTIALS_SPECIFIED
SYSTEM_MODE(SEMI_AUTOMATIC);
#endif

setup() {

```

```
#ifndef WIFI_CREDENTIALS_SPECIFIED
    setupWifi();
#endif

// Rest of setup code goes here
}

void loop() {
    // Do more stuff here
}

void setupWifi() {
    WiFi.on();
    WiFi.disconnect();
    WiFi.clearCredentials();
    int numWifiCreds = sizeof(wifiCreds) / sizeof(*wifiCreds);
    for (int i = 0; i < numWifiCreds; i++) {
        credentials creds = wifiCreds[i];
        WiFi.setCredentials(creds.ssid, creds.password, creds.authType, creds.cipher);
    }
    WiFi.connect();
    waitUntil(WiFi.ready);
    Particle.connect();
}
```

By commenting or uncommenting `#define WIFI_CREDENTIALS_SPECIFIED` you can control if code-specified credentials are placed on the Photon.

Setting up particle projects in this way will give you the flexibility to set your Photon's wifi credentials manually, without having to use additional tools such as Particle's mobile or web applications.

Cheers!

Related: [Building a Wifi Touch Light](#) [Manual Switch Touch Lights](#)

What do you think?

7 Responses



Upvote



Funny



Love



Surprised



Angry



Sad

2 Comments

dev/bleSi



Disqus' Privacy Policy



Login ▾

Favorite

Tweet

Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

**Xerxes Clayton** • 2 years ago

For those wondering, this file exists in the src directory inside of touch_light. Enter that directory and edit the file. You must uncomment both:

```
// #define WIFI_CREDENTIALS_SPECIFIED
```

and

```
// {.ssid="SSID", .password="password", .authType=WPA2, .cipher=WLAN_CIPHER_AES}
```

Cheers!

^ | ▾ • Reply • Share ›

**Tony Abbey** • 2 years ago

Thank you for this blog on how to add extra wifi credentials. I have one question which I don't believe you answered here - do I just run this code once as an addition at the start of my existing code, flash it, then delete it from the Web IDE and reflash? Does this then make the Photon retain whatever extra Wifi creds were added?

Tony

^ | ▾ • Reply • Share ›



Subscribe



Add Disqus to your siteAdd DisqusAdd



Do Not Sell My Data

#DIY #Particle Photon #Electronics #Maker #IOT

© 2021 Patrick Blesi

Powered by Jekyll using the Clean+Simple theme