Smart Toilet Light
Created by Tony DiCola

Last updated on 2018-08-22 03:53:18 PM UTC
## Guide Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Contents</td>
<td>2</td>
</tr>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Hardware</td>
<td>4</td>
</tr>
<tr>
<td>Parts</td>
<td>4</td>
</tr>
<tr>
<td>Assembly</td>
<td>4</td>
</tr>
<tr>
<td>3D Printed Case</td>
<td>6</td>
</tr>
<tr>
<td>Software</td>
<td>10</td>
</tr>
<tr>
<td>Arduino Sketch</td>
<td>10</td>
</tr>
<tr>
<td>Adafruit IO</td>
<td>11</td>
</tr>
<tr>
<td>IFTTT Integration</td>
<td>14</td>
</tr>
<tr>
<td>Zapier Integration</td>
<td>19</td>
</tr>
</tbody>
</table>
Overview

The smart toilet light project brings IoT to your bathroom. IoT, or Internet of Toilets, is the popular industry buzzword to describe a connected social network of intelligent toilets. With the smart toilet light you can create a light that illuminates your toilet bowl based on data from the internet. For example your toilet can shine red if it's going to rain, or glow yellow if it's sunny and clear outside. Not only is the smart toilet light a useful aid to find the toilet in the dark, but it's a real source of information to help you plan for your day!

This project is built with a [Feather Huzzah ESP8266](http://adafru.it/2821) WiFi microcontroller and a [8mm NeoPixel LED](http://adafru.it/1734). You can 3D print a case and a LED clip to hold the hardware near the toilet bowl. By connecting to [Adafruit IO](https://adafru.it/fH9) the ESP8266 can listen for color and animation commands that are sent from other services like [If This Then That](https://adafru.it/mxF) (IFTTT), [Zapier](https://adafru.it/mxE), and more. You'll have your toilet pulsing and flashing different colors from the internet in almost no time!

Before you get started you'll want to familiarize yourself with the following guides:

- [Adafruit Feather Huzzah ESP8266](https://adafru.it/kD3)
- [Adafruit NeoPixel Uberguide](https://adafru.it/ja1)
- [Adafruit IO](https://adafru.it/kSb) & [Adafruit IO Basics](https://adafru.it/iDX)

In addition you'll want to be familiar with how to solder--for this project you'll need to solder headers to a board, and wires to a LED. This is a relatively easy soldering project even for a beginner, but be sure to read the [guide to excellent soldering](https://adafru.it/dxy).
Hardware

Parts

To build this project you'll need the following parts:

- **Feather Huzzah ESP8266** ([http://adafru.it/2821](http://adafru.it/2821)). You can use other ESP8266 boards like the simpler Huzzah ESP8266 ([http://adafru.it/2471](http://adafru.it/2471)), but you'll need more parts like a serial to USB cable to talk to them. Pick up a set of female Feather headers ([http://adafru.it/2830](http://adafru.it/2830)) to make connecting wires to the Feather easier.
- **NeoPixel Diffused 8mm Through-Hole LED** ([http://adafru.it/1734](http://adafru.it/1734)). You can use any NeoPixels but I recommend the large 8mm through-hole LEDs for their large size and brightness. The 3D printed LED holder for this project is designed for the 8mm NeoPixel.
- **5V power supply** ([https://adafru.it/e5A](https://adafru.it/e5A)). The easiest way to power the board is with a 5V power supply plugged into the USB port. You could use a battery like a lipoly pack, but I don't recommend putting a lithium battery near where it can get wet or fall into water. **Be careful, electronics and water don't mix! Always plug devices into a GFCI outlet and do not place them where they can fall into water.**
- **Soldering tools** ([http://adafru.it/136](http://adafru.it/136)) & wire ([https://adafru.it/dya](https://adafru.it/dya)). You'll need to solder headers to the Feather and solder wires to the NeoPixel.

Assembly

Start by following the Feather Huzzah ESP8266 guide ([https://adafru.it/kD3](https://adafru.it/kD3)) to assemble and test the ESP8266 board. **Don't continue until you're sure the board is working and you can upload sketches with the Arduino IDE!**

Next solder wires to the NeoPixel. Use a long enough length of wire to place the LED on the rim of the toilet bowl and keep the Huzzah board safely away from the toilet. You'll need to solder wires to the **data in**, **5V** and **ground** pins (the data out wire is unused):

![NeoPixel Connections](image)

It helps to tin each wire and pin with solder first, then hold them together with helping hands and apply the soldering
iron to flow the solder. Don't forget to slip on some heat shrink tubing too!

Connect the NeoPixel to the Feather Huzzah ESP8266 as follows:

- **NeoPixel data** in to Huzzah ESP8266 pin #2.
- **NeoPixel 5V** to Huzzah ESP8266 3.3V power. You can instead use the VUSB pin if you're powering the board from its USB port. Don't power the NeoPixel off the 3.3V power if you're using more than 1 or 2 pixels! Consult the NeoPixel Uber Guide (https://adafru.it/ja1) if you have any questions about how to power NeoPixels.
- **NeoPixel ground** to Huzzah ESP8266 ground.

If you're having trouble getting bare wires to stay inside female headers, tin the wires with solder until they're thick enough to firmly stay in place.
That's all there is to assembling and wiring the hardware for this project!

3D Printed Case

If you have access to a 3D printer you can print a case for the Feather Huzzah ESP8266, and a bracket to hold the LED to the side of the toilet bowl. These small parts should print on almost any 3D printer and are fine to print with PLA.

The LED bracket is designed to hold a 8mm LED and fit a toilet bowl with a 40mm wide rim. It will probably work for slightly larger and smaller bowls, but you might need to adjust the part or design your own if your toilet is different. You can find the source for this part on Tinkercad (https://adafru.it/mxd) and adjust it as necessary.

Download the STL files for these parts from their home on Thingiverse (https://adafru.it/mzB) below:

https://adafru.it/mzC
https://adafru.it/mzC

The parts should print without any need for support. For PLA I used 20% infill and 0.02mm layer height with good results, but adjust as needed for your filament, printer, etc.
The Feather Huzzah ESP8266 should snap into place without need for screws, etc. The case can optionally accommodate a slide switch however it is unused in this project.
For the LED holder thread the wires through the side and hole in the top, then pull them through to put the LED into place in the hole. See the photos on the left for more details.

Once assembled your smart toilet light should look something like the following:
Place the clip on the rim of the toilet and make sure it is secure. You can use tape or other means of adhesive if the clip is not held firmly in place (be careful not to use anything too permanent though!). The Feather Huzzah ESP8266 should be placed somewhere secure and near a power source.

Be careful with electronics and water! Don't place anything near water which could cause harm if it were to get wet or fall into water. Always plug devices into a GFCI outlet, and ensure devices are secure and not able to fall into water!
Software

Arduino Sketch

The software for this project is an Arduino sketch that will connect to Adafruit IO (https://adafruit.io/fH9) and listen for color and animation data from a feed. You can even setup integration with IFTTT (https://adafruit.io/mxF), Zapier (https://adafruit.io/mxEx), etc. to change the feed and toilet light color when something happens. For example change the toilet color based on your local weather!

Before you setup the software for this project make sure you've followed the Feather Huzzah ESP8266 guide (https://adafruit.io/kD3) and can successfully program the board. You will also want to install the following libraries with the Arduino library manager (https://adafruit.io/fCN):

- Adafruit NeoPixel Library (https://adafruit.io/aZU)
- Adafruit MQTT Library (https://adafruit.io/fp6)
- Adafruit SleepyDog Library (https://adafruit.io/fp8)

Then download the Arduino sketch for the project from its home on GitHub (https://adafruit.io/mxf) by clicking the button below:

https://adafruit.io/mxA

Unzip the sketch and load it in the Arduino IDE. You should see it look something like the following:
There are a few #define values at the top of the sketch that you will need to change:

- **WLAN_SSID** - Set this to the name of your WiFi access point.
- **WLAN_PASS** - Set this to the password for your WiFi access point.
- **AIO_USERNAME** - Set this to your Adafruit IO username (you can find it on [http://accounts.adafruit.com](http://accounts.adafruit.com)).
- **AIO_KEY** - Your Adafruit IO key. You can find this by clicking the View AIO Keys button on the Adafruit IO settings page ([https://adafruit.io/fVT](https://adafruit.io/fVT)).

There are other #define values below that you can optionally change, however I recommend keeping them at their defaults unless you know your hardware is setup in a different way.

Upload the sketch to your Feather Huzzah ESP8266 board. Then open the serial monitor in Arduino and set it to 115200 baud. You should see the board print information as it connects to the WiFi network and then Adafruit IO. For example:

Make sure you see the sketch connecting to your WiFi network and Adafruit IO (the MQTT Connected! message). If you see an error connecting to WiFi check that you have the right SSID name & password. If you see an error connecting to MQTT/Adafruit IO check you have the right username and Adafruit IO key set and try again.

Once the sketch is running and connecting to Adafruit IO you're ready to start controlling it from the web!

**Adafruit IO**

To control the animation and color of the light you can use [Adafruit IO](https://adafruit.io/fH9). If you aren't familiar with Adafruit IO you'll want to read through the following guides first:
By default the software for this project will look for a feed named `toilet-light` in your Adafruit IO account. You can send simple text commands to this feed to change the color and animation of the light.

The easiest way to change and test the light is to directly send a color value to the feed. From the Adafruit IO feeds list for your account find the feed named `toilet-light`. If you don't see a feed named `toilet-light` then use the Create Feed button to create one (name it `toilet-light`).

Click the toilet-light feed to go to its feed page:

![Toilet-light feed](https://learn.adafruit.com/smart-toilet-light)

Click the blue + icon in the lower right to manually set the value of the toilet-light feed. This will bring up a dialog where you can enter a new feed value.
Enter the value **SFF0000** (those are zeros, not the letter O), then press **Save Changes**. After a moment you should see your toilet light LED turn red!

The value that you set the toilet-light feed is a command that can control the color and animation of the LED. You can send the following types of commands:

- **Srrggbb** - Set the LED to a solid color specified by the hex rr, gg, bb values (red, green, blue). For example the hex RGB color FF00AA (red = 255, green = 0, blue = 170) could be set with the command: **SFF00AA**
- **Prrggbbff** - Pulse the LED from full color to off repeatedly using the specified hex rr, gg, bb values (red, green, blue), and at the specified hex frequency ff. Frequency is specified as a hex byte value (0-255) and is the number of times the light should pulse from off to on and back off again in a 10 second period. For example to pulse the LED red every 2 seconds you would send: **PFF000005**
- **R** - Animate the LED with a rainbow color cycle. There's no parameter or other information to send, just set the feed to the value **R**!

Try setting the toilet-light feed to different solid, pulsing, and rainbow cycle values. That's all there is to controlling the light from Adafruit IO! Continue on to learn about interesting ways to drive the light, like from weather triggers on IFTTT and Zapier.
IFTTT Integration

If This Then That (https://adafruit.ioe) is a great internet service that simplifies home automation. IFTTT can listen to services like GMail, Dropbox, Twitter, etc. and take actions like controlling lights in your home or even sending data to Adafruit IO. Using IFTTT you can trigger your toilet light to change color in many interesting ways. For example you can have IFTTT change the color of your toilet light based on the current weather in your area.

To setup IFTTT to show weather on the toilet light you'll first need to sign up for an IFTTT account. Then create a new recipe:

Click ‘this’ to pick what action will trigger the toilet light. Search for weather and you'll see IFTTT's native weather trigger channel (the blue cloud and yellow sun on the right):

Click the weather trigger channel to select it and see the available triggers. Note that you might need to first add the weather trigger channel and configure it to your location before you can continue.

For this example you want the Current condition changes to trigger which will fire when the weather in your area changes to rain, snow, clouds, or clear conditions. However read the other triggers to see what other weather actions you might use with the toilet light.
After selecting the **Current condition changes to** trigger you can now pick the specific condition that this trigger will action on. For example to change the light when it's raining pick the **Rain** condition in the **Current condition** drop down:

Now click **Create Trigger** to continue on to the ‘**that**’ configuration for this recipe:

Click ‘**that**’ and you'll see the available action channels. Search for or select the **Adafruit** channel. Note that you might need to do some one time setup to authorize IFTTT the first time you select Adafruit's channel.
Select Adafruit’s channel and pick the **Send data to Adafruit IO** action:

Now select the **toilet-light** feed in the **Feed name** drop down (if you don’t see the feed then make sure you’ve setup and changed the toilet light LED color by following all the steps in the previous page).

For the **Data to save** value you can specify a command to change the toilet light color. For example the command **PFF000005** will pulse the toilet light red every 2 seconds. Use any command like changing to a solid color, pulsing other colors, or animating a rainbow cycle—check out the [previous page](https://adafruit.it/mxD) for the command syntax.
Click **Create Action** to finish setting up the recipe.

Give the recipe a descriptive name and then click **Create Recipe** to finish and enable it. When the weather changes to rain you should see the toilet light pulse red!

Try setting up more recipes to change the light when weather conditions change to snow, clouds, or clear skys:
That's all there is to setting up the toilet light with IFTTT! You can hook up almost any action to control the toilet light. For example have it flash when you have too many unread emails or missed a phone call. The possibilities are almost endless!
Zapier Integration

Zapier (https://adafruit.it/mxE) is another great service for automating and connecting other online services. Just like IFTTT (https://adafruit.it/mxF) you can use Zapier to trigger on actions and send data to Adafruit IO. For example you can setup Zapier to change the toilet light color based on the weather.

First you will need to create an account on Zapier’s website (https://adafruit.it/mxE). Then login and click Make a Zap:

In the list of triggers tha appears search for weather and select the Weather by Zapier trigger.

Choose the Will it Rain Today? trigger for a simple trigger to change the toilet light color if it will rain.
Follow the steps on the page to pick the right latitude and longitude for your location.

After clicking continue Zapier will confirm the latitude and longitude you chose, and test that it can access weather data. Click Fetch & Continue to move on.
After the test runs it should succeed and you can click **Continue** to pick the action that will be taken when Zapier detects it will rain.

In the list of actions search for **Adafruit** to find the **Adafruit IO** action. Choose the action and you'll see it has a **Send Value to Feed** option:
After clicking **Save + Continue** you'll need to setup your Adafruit IO account with Zapier. If you've already setup the account then choose it from the list, otherwise click **Connect a New Account** and go through the authorization process.

Again click **Save + Continue** to move on to set the feed and feed value. For the **Value** pick a command to change the toilet light color, like **PFF000005** to pulse the light red every two seconds. You can use any toilet light feed command, like setting a solid color or rainbow cycle, see the command syntax on the [previous page](https://adafruit.it/mxD).

For the **Feed** pick the **toilet-light** feed that powers your toilet light.
Click **Continue** and Zapier will move on to test the Adafruit IO connection. Click **Create & Continue** to perform the test.

After the test runs you're almost done! Click **Finish** to move on and give the zap a name.

Name the zap and then click **OFF** to toggle the Zap ON! Your toilet light should now pulse red when Zapier performs a
rain check every morning. Cool!

Explore other triggers on Zapier's site to see more interesting ways to control your toilet light. For example have it flash when you have too many unread emails or missed a phone call. The possibilities are almost endless!