EE 1301: Introduction to Computing Systems

IoT Laboratory #1

Introduction to Particle Photon

Created by: David Orser, Kia Bazargan, and John Sartori

Many thanks to the students, teaching assistants, and faculty that work to continually improve this document. Together we make better labs!

Please send comments and suggestions to orser@umn.edu

Background

Microcontrollers provide the first line of machine intelligence in today's everyday devices. For example, a Keurig coffee pot contains a microcontroller. This device takes input from the user (buttons), water level sensors, temperature sensors, clocks, and operates a series of pumps, motors, and valves to produce a cup of coffee when required. Everything from coffee pots to quadcopters have at least one microcontroller embedded in them. The microcontroller transforms everyday devices into **smart devices**.

> A **smart device** senses events in the world around it, makes decisions based on its programing, and controls actuators to respond to those events.

The big idea behind the Internet of Things (IoT) is adding communication capabilities to items that we use every day. Smart devices are evolving to perform communications through wifi or cellular connectivity. Much like our entertainment, business, and our personal lives have changed significantly because of mass communication networks, adding communication opens up entirely new possibilities in smart devices. For example, a regular parking meter for a parking spot cannot tell you whether a car is parked in the spot unless you physically go to the spot and look. On the other hand, an internet-connected parking meter could report to you over the internet whether a car is parked in a spot, saving you the trouble of traveling to the spot and looking. Taking this one step further, a network of internet-connected parking meters could report parking availability for **every parking spot in the city**. A cellphone app looking at the information reported by the parking meters could tell you what is the closest available parking spot to your current location, saving you the trouble of trolling the streets to find a place to park. This is the power of the internet of things.

In this lab we'll be utilizing a microcontroller with an embedded WiFi interface as our communications capability. This will allow you to interact with your device, communicate with the outside world (or other devices), and program your device wirelessly.

Purpose

The purpose of this lab is to familiarize you with your new Particle microcontroller, perform first time configuration of your new Particle microcontroller, create a basic "Hello World!" program, and provide you with a basic knowledge of the Particle Photon microcontroller.

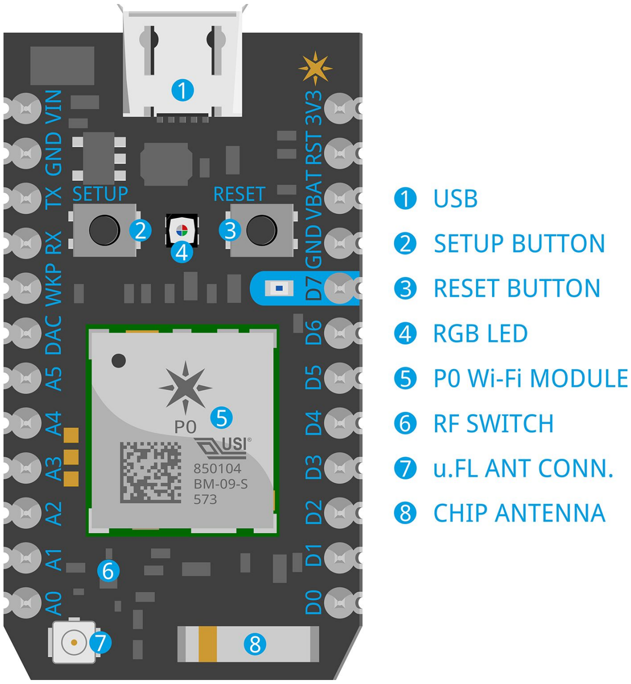
Supplemental Resources

https://www.particle.io
http://docs.particle.io/photon/
Photon - Common Issues (ECE maintained Photon debugging help)

Orientation

The Photon operates on power over USB (similar to how you charge your cell phone). It has several interface components described in the diagram to the right. The most important of these are:

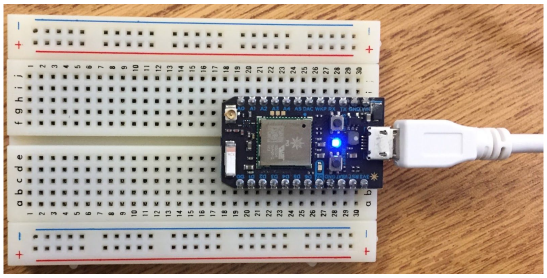
- #2 the Setup button
    - Used to put the device into listen mode (over USB or Wifi)
- #3 the Reset Button.
    - Used to reboot the Photon
- #4 the RGB status LED
    - The color of this LED shows the status of the Photon
    - See appendix “Device Modes” for more details



There is a nice overview of the Photon available online at: http://docs.particle.io/photon/

First-time Setup

This section describes how to get your Photon attached to our WiFi network and how to “claim” your photon. “Claiming” your Photon is the process by which you lock your Photon to your online account at particle.io. This will secure your Photon so that it only responds to you or someone you provide with a security key.



First off, go ahead and plug in your Photon to the white breadboard (with lots of holes) as shown in the figure on the right. Note that we have oriented the breadboard in such a way that the blue line is on top. Plug the small end of the USB cable into the Photon. NOTE: do not force the USB cable into the socket of your Photon board: you might be trying to connect it upside down and might break the socket. Do NOT plug in the other end of the USB cable to your computer yet.

This section is split into 3 parts:

- USB-based setup - choose the Linux, Mac, or Windows section
- Hardware setup
- Claiming your Photon

> NOTE: The above links are for first time setup in a classroom, discussion section, or lab section. If you are getting an early start with your Photon <u>on your own</u>, please check out the appendix [First-time Setup Outside of Lab]()

USB-based Setup (Linux)

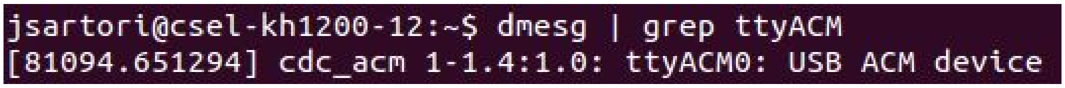
(If you are using OSx (Mac) or Windows, skip this section)

1. Connect to your Photon via serial terminal
    a. With your Photon disconnected from the computer, enter the following command in the terminal.

       ```
       $ dmesg | grep ttyACM
       ```

       dmesg prints out the kernel message buffer in a linux system. USB serial ports that the Photon connects to are listed in this message buffer as USB ACM devices. “dmesg” by itself prints a lot of other information that is not relevant for our purposes. Hence, we pass (or in linux terms, “pipe”) the output of dmesg to “grep”, which just prints out lines in the output of the dmesg command that contain the string “ttyACM”.

       The output of the dmesg command will look something like the following.

       ```
       jsartori@csel-kh1200-12:~$ dmesg | grep ttyACM
       [81094.651294] cdc_acm 1-1.4:1.0: ttyACM0: USB ACM device
       ```

       If the computer has any USB serial devices connected, they will be listed. If there are no USB devices connected, the output of the dmesg command may be blank. Take a note of how many ttyACMx devices are listed. In general serial ports in Linux are listed as /dev/tty*, where * is something like S0, S1, ACM0, ACM1 etc.

    b. Connect your Photon to your computer by plugging the larger side of the USB cable to your computer’s USB port. Your device should startup and begin quickly flashing blue. If your device is not brand new, it may need resetting, see the Appendix - “resetting your device”.

    c. Enter the same dmesg command into your terminal again:

       ```
       $ dmesg | grep ttyACM
       ```

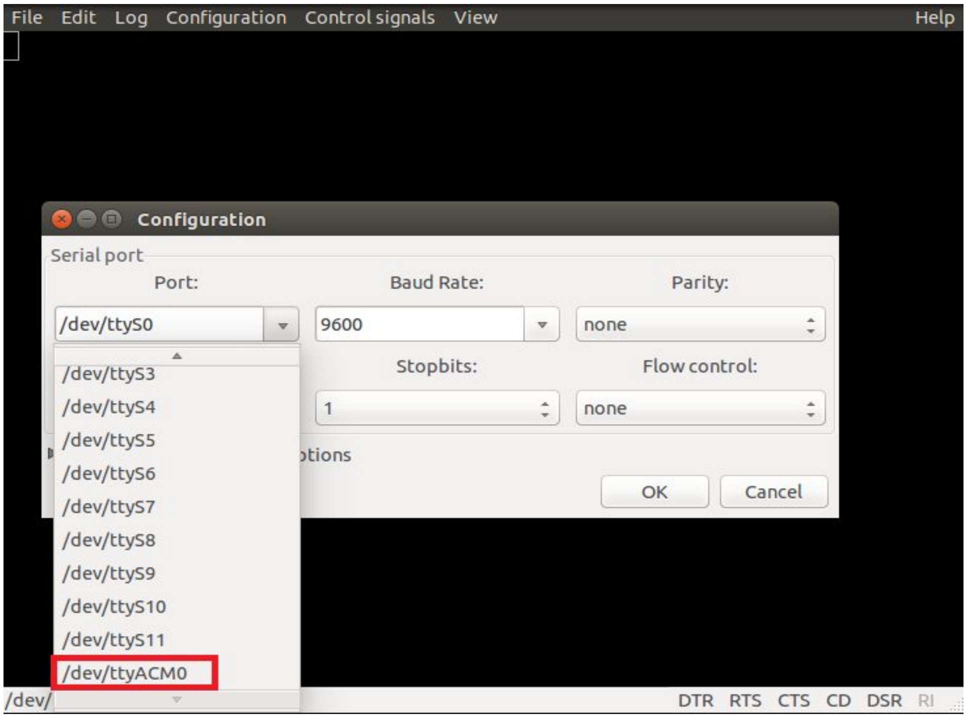
       This time, your Photon should be listed as a new USB ACM device with ttyACMx different from the ones listed when you executed the command before.

Remember the new “ttyACMx” that appears when you plug in your Photon. That’s the serial port your Photon is connected to. In the example screenshot above, the Photon is connected to /dev/ttyACM0.

2. Open gtkterm by entering the following command in the terminal.

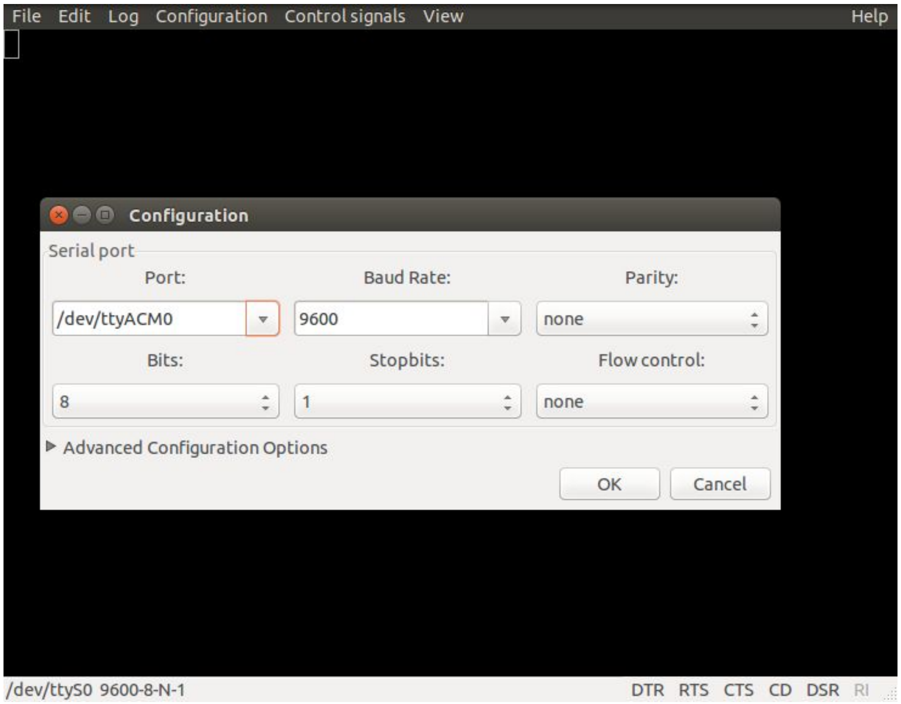
```
$ gtkterm
```

Now, click on Configuration > Port. Select the port that you noted down in the previous step. In our case this is /dev/ttyACM0.



Port: /dev/ttyS0
Baud Rate: 9600
Parity: none
Stopbits: 1
Flow control: none

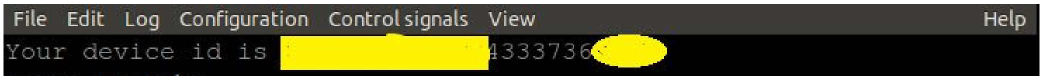
Click OK to connect. Now you can interact with your Photon through the serial port.


File Edit Log Configuration Control signals View Help
Configuration
Serial port
Advanced Configuration Options
OK
Cancel
DTR RTS CTS CD DSR RI

Port: /dev/ttyACM0
Baud Rate: 9600
Parity: none
Bits: 8
Stopbits: 1
Flow control: none

/dev/ttyS0 9600-8-N-1

3. Now we are ready to configure your Photon and setup WiFi credentials. In gtkterm, type "i" to get the Photon's unique ID. Copy down your ID number some place (hint: select the ID with your mouse and press Ctrl-C to copy to the clipboard. Open a text editor and press Ctrl-V to paste. Save it somewhere you can get it later.)



```
Your device id is [redacted]4333736[redacted]
```

4. Skip the next sections on Mac and Windows Setup and go to "Configuring your connection"

USB-based Setup (Mac)

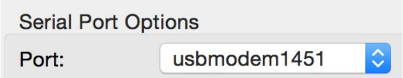
(If you have a Windows PC, skip this section)

1.) Download a serial port client or terminal program
   a.) CoolTerm is a lightweight terminal program for Mac OS X. There are other terminal programs as well as a method to utilize the command-line as a terminal. You are welcome to use anything that works for you, but we'll only cover CoolTerm here.
   b.) Goto http://freeware.the-meiers.org/
   c.) Select and download the latest version of CoolTerm
   d.) Extract the ZIP file (CoolTerm_Mac.zip)
   e.) Drag the CoolTerm application to the Applications Folder (usually located in the left navigation pane under "Favorites")

   f.) To run the program for the first time you will need to open the applications folder in the finder and right-click and “Open” CoolTerm. You will likely be prompted to confirm you wish to run this program. Please do so.

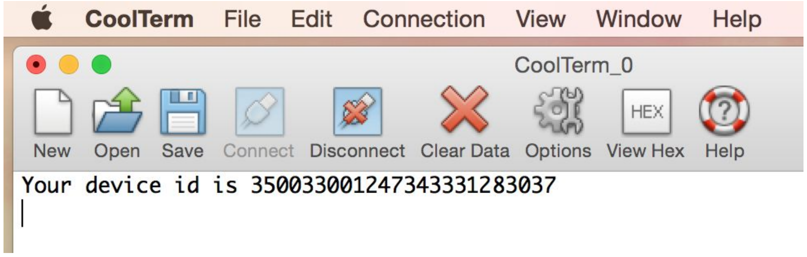
2.) Connect your Photon to your computer by plugging the larger side of the USB cable to your computer’s USB port. Your device should startup and begin quickly flashing blue. If your device is not brand new, it may need resetting, see the Appendix - “resetting your device”.

3.) Connect to your Photon by serial terminal
   a.) Start up CoolTerm
   b.) Click on Options
   c.) Select the port to be “usbmodem…”



Serial Port Options
Port: usbmodem1451

   If you don’t see the usbmodem option, click on the refresh button, wait a minute, and potentially re-plug your USB cable.
   d.) Click on Connect
   e.) Type the letter “i” into the window it should respond with a device ID



```
Your device id is 350033001247343331283037
```

   f.) Your ID number would be different from the example above. Copy down your ID number some place (hint: select the ID with your mouse and press Cmd-C to copy to clipboard. Open textedit or similar text editing software and press Cmd-V to paste. Save it somewhere you can get it later.)

4.) Skip the next section on Windows Setup and go to [“Configuring your connection”](#)

(For additional information on CoolTerm see: https://learn.sparkfun.com/tutorials/terminal-basics/coolterm-windows-mac-linux)

USB-based Setup (Windows)

1.) Students using KHK 4-138 Computers and *all* Windows 10 and later computers, you may skip this step. If you are for some odd reason still running Win7,8,etc. You will need to install a serial port driver from this link:
https://s3.amazonaws.com/spark-website/Particle.zip
More information here:
https://community.spark.io/t/installing-the-usb-driver-on-windows-serial-debugging/882

2.) Obtain PuTTY or a similar serial port client. (If you already have PuTTY on your machine skip this step.)

   a.) PuTTY is an extremely lightweight terminal program. It is capable of connecting to serial ports, telnet, and SSH protocols. It is available as a stand-alone executable, there is no need to install this program, just copy the executable into a location where you won't lose it. Just double click to run.

   Website: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

   Under "For Windows on Intel x86" → Next to "PuTTY" → click on "putty.exe"

   OR use this Direct Link: http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

3.) Connect your Photon to your computer by plugging the larger side of the USB cable to your computer's USB port. Your device should startup and begin quickly flashing blue. If your device is not brand new, it may need resetting, see the Appendix - "resetting your device".

4.) Check that your device was installed correctly

   a.) Go to Device Manager (WindowsKey → Type "Device Manager", press Enter)

   b.) Select the twiddle (▷) "Ports (COM & LPT)"

   If you don't see Ports, click on View → Show hidden devices.

   c.) Write down the serial port name IMPORTANT: note the COMx number that you see behind your "Photon". For example it could be COM3, COM6, etc.

   d.) If the device has a yellow exclamation point

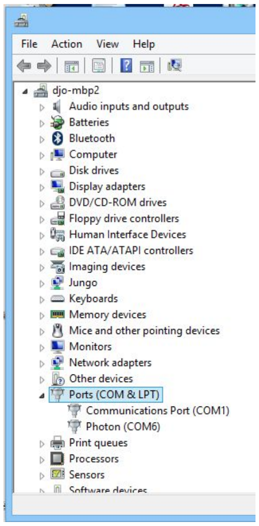
      i.) Right-Click on "Spark Photon…" → Click on "Update Driver Software…"

      ii.) Select "Browse my computer…" and select the folder to which you extracted "spark.zip"

      iii.) If it still fails to install correctly, plug and unplug your Particle.

      iv.) If it still fails to connect, talk to your TA.

   e.) Note down the serial port name (usually "COM6" or similar)



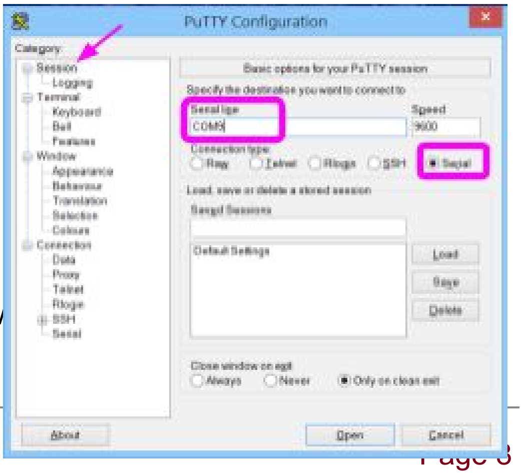
- djo-mbp2
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Floppy drive controllers
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Imaging devices
  - Jungo
  - Keyboards
  - Memory devices
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
  - Ports (COM & LPT)
    - Communications Port (COM1)
    - Photon (COM6)
  - Print queues
  - Processors
  - Sensors
  - Software devices

5.) Connect to your Particle Photon by serial terminal

   a.) Start Up PuTTY[1].

   b.) Click on "Session", which is the top item on the left pane list.

   c.) Select the "Serial" button as shown in the figure below.



Specify the destination you want to connect to

Serial line: COM6

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

[1] Remember that you placed "putty.exe" in a folder. Now "putty.exe" (or "putty" if your computer doesn't show file

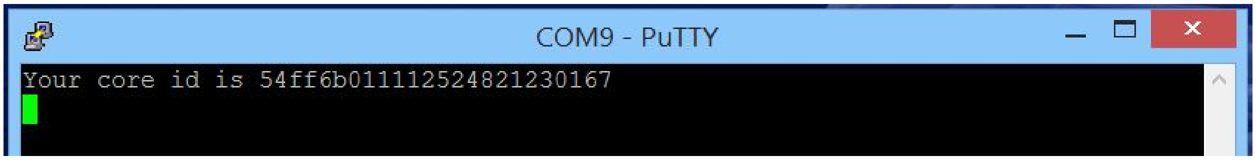
Copyright 2018

   d.) Under “Serial line”, type the COM port name as you saw in step 4-b above (e.g., if you saw COM3 in front of “Spark Photon” in Step 4-b, you should type “COM3” without the double quotes under “Serial line”. Click on the “Open” button.

6.) A blank black window will open up.

7.) Type the letter “i” to verify your device is correctly connected.

The device will respond with its Device ID, something like:



```
Your core id is 54ff6b011112524821230167
```

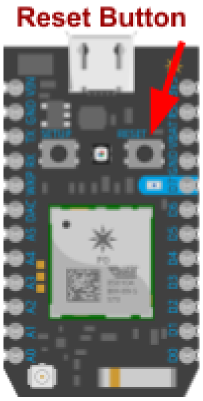
8.) Your ID number would be different from the example above. Copy down your ID number some place (hint: select the ID with your mouse and press Ctrl-C to copy to clipboard. Open notepad.exe and press Ctrl-V to paste. Save it somewhere you can get it later.)

Configuring Your Connection (Linux, Mac, and Windows)

While still connected to the USB port of your computer, continue to the following steps

1.) Press “w” to configure your wireless
   a.) Set the SSID = “CSE-IoT” (without quotes)
   b.) If prompted select **WPA2** security and **AES+TKIP** (both usually option #3)
   c.) Password = “**Go!Gophers**” (without quotes)

2.) Your photon **may** now reboot on its own. If you wait ~10 seconds and don’t see “Particle <3 you!”, look at your Photon. Either:
   a.) The light flashed blue a couple times and your photon is blinking blue again. In this case press the RESET button and continue below.
   b.) The light is blinking some combination of green-slow, green-fast, cyan-slow, cyan-fast then it is trying to connect. Wait a full 2 mins, then contact your TA if it doesn’t change.
   c.) If it is flashing magenta. Your photon is trying to complete a firmware update, do not interrupt this stage! Just wait!
   d.) The light is slowly “pulsing” cyan. Congratulations your Photon is connected to the network! Continue on to the lab section “Claim Your Photon”. If you’re photon doesn’t pulse cyan (connected) and isn’t flashing magenta (firmware update), press RESET one more time and wait an additional 2 mins, then contact your TA for assistance.



Allowing Your Device to Connect to Additional Wifi Networks

If you want to setup your Photon to connect to more wifi networks (e.g., your home network), then you can connect it to your computer’s USB port, and press and hold the SETUP button for about three seconds. The Photon will enter the “listen mode” again (more details can be found in the section called Appendix - Resetting Your Device into Listen Mode in this document). You can connect to it using your terminal program, press “w” and enter additional wifi credentials.

What Just Happened?

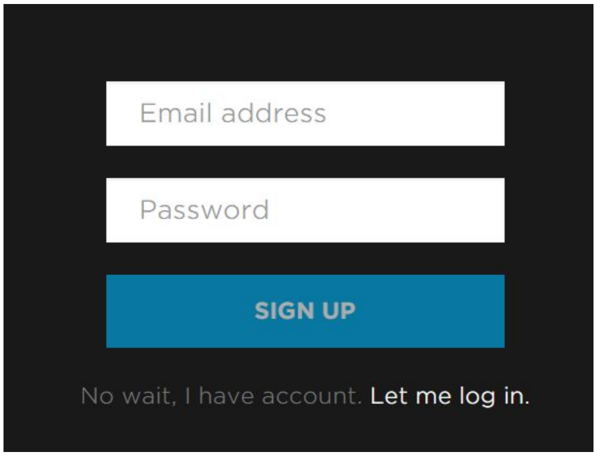
Just to make sure you don’t think of any of this as black magic, here’s the summary of what we just did. The Photon board can do many things, including communicating with sensors and actuators in its environment, AND talking to other entities on the web. We have not utilized any of that yet. All we did was to ask the Photon to listen to the USB port. We sent a “command” by pressing the “i” key, and it responded by sending us its ID in the text format. The terminal client (CoolTerm or PuTTY) on our computer knew how to decipher the electric signals coming back on the USB cable and translate it to the message you saw on the screen.

As of now, your Photon board cannot talk to anyone on the web. In Lab 2, we will configure the board to use sensors and actuators. In Lab 3, we will explore how to listen to commands, how to send data, and take actions over the internet. Next, in this lab, we will setup and explore the online integrated development environment (IDE) for the Photon.

Claim Your Photon

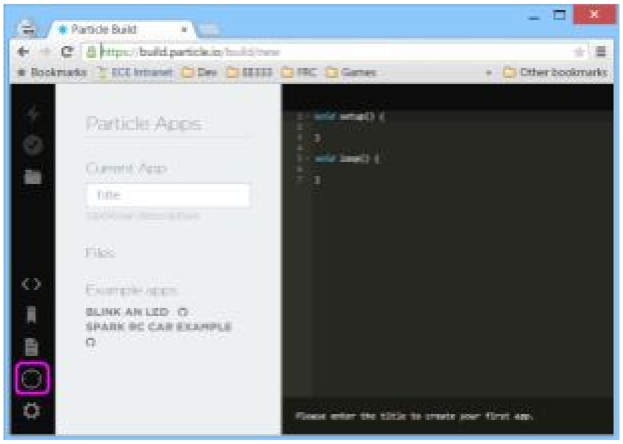
In this section we will create a new user account at Particle.IO and claim your Photon. Doing this locks the unique ID of the Photon to your user account. This prevents other people from overwriting your code and taking control of your Photon.

1. Go to the website: https://build.particle.io/build (we recommend you bookmark this page.) You should see something like the following prompt the first time you connect.



No wait, I have account. Let me log in.

2. This prompt is asking you to create a new account. Type in your UMN (or other) email address and create a new password (better not to use your UMN password here for security reasons).
3. Once you fill in the form, the Particle web-based development environment will appear.
4. Select the “Devices” button ⌖



```
void setup() {

}

void loop() {

}
```

Please enter the title to create your first app.

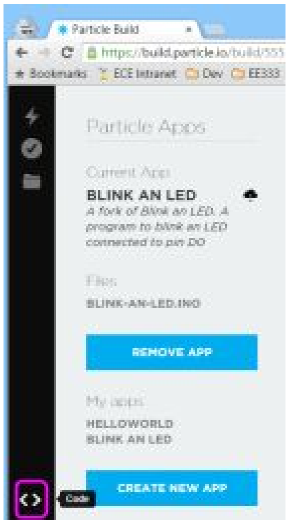
5. Click “Add New Device”.
6. Type in your giant hex string device ID recorded previously; for example “54ff6b011112524821230167”.
7. Click “Claim a Photon”.
8. Name your Photon, your name should be short, unique, and memorable. For example your X.500 short name, followed by “Photon” (e.g., OrserPhoton).

No More Wires!

From this point on, we will be using the USB connection to the computer only as a power source. In fact, if you disconnect the USB cable from your computer and plug it into a power source (e.g., a battery like the one on the right), the rest of the steps in this lab will work just fine. The designers of the Photon have programmed it so that it regularly checks-in with the website particle.io to see if it has any messages (more on this later).

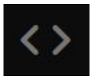
The ID number you entered claims your Photon to your account. This allows secure communication between the particle.io servers, your IDE, and the Photon. Using this secure channel we can send and receive commands, and even reprogram the Photon, all wirelessly!

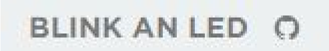
“Hello World!”



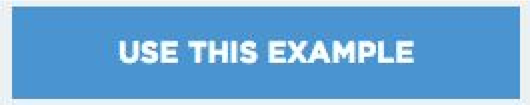
In this section we will create a very basic program to test the connectivity and functionality of your new Photon. We write the program on the particle.io website, and ask the servers on the site to send the program to our Photon board.

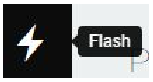
1.) Goto particle.io/build (or continue from above)

2.) Select the Code  button

3.) Click  . This brings up a pre-written program to make a small blue LED on the Photon board flash in a pattern of 1-second on, 1-second off, 1-second on, and so on.

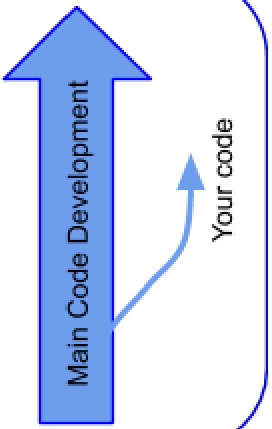
4.) Click



5.) Click  . In microcontroller terminology, “flashing” a program means loading its step-by-step instructions (i.e., compiled program code) to the flash memory on the device. You are familiar with this concept: you flash the chips inside your smartphone every time you upgrade it to the newest version of its operating system.

Fork (software)

In software engineering terminology, a “fork” happens when a developer copies the source code of an existing piece of software and starts independant development. Sort of like cloning yourself, and then adding an extra eye to make you look even better!



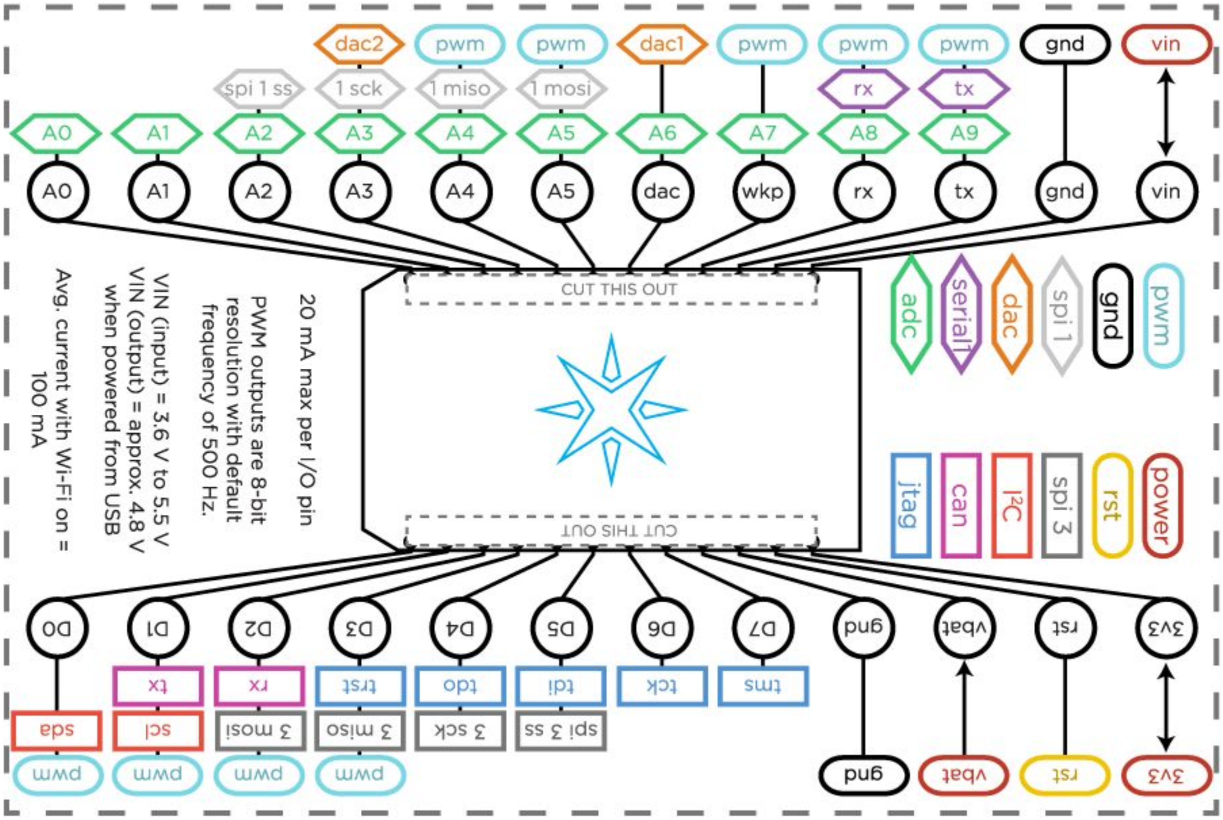
6.) Watch your Photon. It should flash Magenta (red and blue), then cycle through several more colors, until it reaches Breathing Cyan again. The different colors of the status LED are detailed in the appendix.

7.) Your small Blue LED should be blinking slowly (1 sec on, 1 sec off)
8.) Modify your code so the LED blinks faster (on and off each for 0.25 secs).

STOP and READ this!
Do not cut-n-paste code for these labs! PDFs and many other text sources can contain hidden characters. When pasted into the build.particle.io text editor these hidden character will cause compilation errors. You have been warned.

Appendix - Photon Pinout Diagram

Many pins in the Photon can be utilized in a variety of ways. The following pin diagram may be useful when deciding which outputs to use on a project.



20 mA max per I/O pin

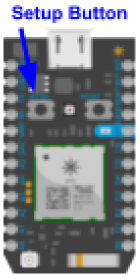
PWM outputs are 8-bit resolution with default frequency of 500 Hz.

VIN (input) = 3.6 V to 5.5 V
VIN (output) = approx. 4.8 V when powered from USB

Avg. current with Wi-Fi on = 100 mA

More details available at:
http://docs.particle.io/photon/photon-datasheet/#3-3-pin-out-diagrams

Appendix - Resetting Your Device into Listen Mode

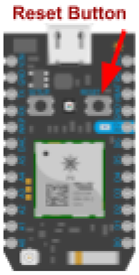


Listen mode is indicated by a flashing blue (not cyan) light. This indicates the Photon is awaiting configuration instructions.

1.) Hold the SETUP button for 3-6 seconds, wait for the light to flash blue

2.) If this doesn't work, press and release RESET, then press and hold the SETUP for 3-6 seconds.

3.) If your Photon is still unresponsive at this point, contact your TA.



Appendix - Clearing WiFi List

Every time you enter new wireless network credentials it adds them to a list inside the Photon. The Photon goes through this list every time it boots up and tries each connection until it finds one that works. This can take up to 10 seconds per network! If you enter incorrect information 10 times, you will have to wait 2 minutes to connect each time your Photon reboots!

Simply press and hold the setup button for a little over 10 seconds to clear the WiFi credentials list. It should flash blue rapidly and then switch into Listen Mode. You will now need to re-enter the correct Wifi credentials as instructed in the First-time Setup section.

Photon Documentation Page - Wi-Fi Network Reset

Appendix - Debugging Help

For advanced debugging help, including the flash green forever issue: Go to the document: https://docs.google.com/document/d/1dfAUDLg2COK4tPWFwb-x4yWIzfxGjCA-fOTO6G3U2Fk/edit?usp=sharing

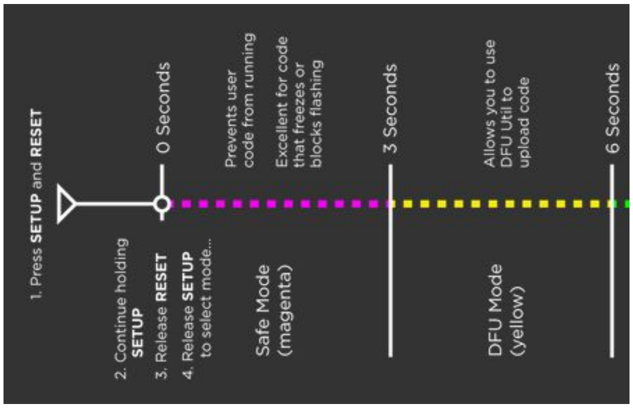
Appendix - Device Modes

Quick and dirty mode list:

(refer to https://docs.particle.io/guide/getting-started/modes/photon/ for more details)
Blinking blue: Listening for Wi-Fi setup
Solid blue: Getting Wi-Fi info from app
Blinking green: Connecting to the Wi-Fi network
Blinking cyan: Connecting to the Particle Cloud
Blinking magenta: Updating to the newest firmware
Breathing cyan: Connected!

Reset diagram of Photon mode selection

To reset your photon, press and hold the SETUP and RESET buttons, release RESET...wait, then...release SETUP when the following colors show up to access that mode:


1. Press SETUP and RESET
2. Continue holding SETUP
3. Release RESET
4. Release SETUP to select mode...
0 Seconds
Prevents user code from running
Excellent for code that freezes or blocks flashing
Safe Mode (magenta)
3 Seconds
Allows you to use DFU Util to upload code
DFU Mode (yellow)
6 Seconds

Appendix - (Optional) First-time Setup Outside of Lab

Should you be interested in getting a jump start on using your Photon in a location where the EE-Labs network is not available, this section and the Particle website will get you started.

Requirements

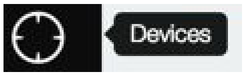
- Particle Photon
- USB Cable for Power Only
- Android or iOS based cell phone
- Access to a 2.4 GHz WPA2-PSK wireless network (UofM, UofM_Secure, etc. do not work!)

Note: You will want to make sure only one new Particle Photon is available on your local wireless network at one time.

To setup your new Particle Photon, follow the setup instructions available at http://docs.particle.io/photon/start/

Glossary of Frequently Used Terms

- **Serial Port** - a communication physical interface through which information transfers in or out one bit at a time --Wikipedia; Examples of serial ports are USB and RS-232
- **Terminal** - A text terminal, or often just terminal (sometimes text console) is a serial computer interface for text entry and display. --Wikipedia; Examples of terminal programs in modern computer are PuTTY or CoolTerm
- **Microcontroller** - A microcontroller (sometimes abbreviated μC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. --Wikipedia
- **Firmware** - In electronic systems and computing, firmware is a type of software that provides control, monitoring and data manipulation of engineered products and systems. --Wikipedia
- **Forking a program** - In software engineering, a project fork happens when developers take a copy of the source code from one software package and start independent development on it, creating a distinct and separate piece of software. --Wikipedia
- **Flashing a microcontroller** - Flashing involves the overwriting of existing firmware or data in programmable memory (ie., EEPROM) present in an electronic device with new data.
- **Sensor** - In the broadest definition, a sensor is an object whose purpose is to detect events or changes in its environment, and then provide a corresponding output. --Wikipedia
- **Actuator** - An actuator is any device that makes physical changes in the real world (from LEDs to motor.)
- **API** - In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types. --Wikipedia
- Device ID (and how to get yours) - In reference to your Photon: Click on

  ,
- Access token (and how to get yours) - In reference to your Photon Click on