Para Manana

1. Bring laptop if you have one
2. Download Particle App
3. Register for free Particle account (can do this through the app)
4. Download Blynk App
5. Bookmark https://github.com/jlosaw/SDQcreates


ENVENTYS
PARTNERS
PRODUCT DESIGN
BOOTCAMP
DAY 2: IoT
February, 2020

OUR PROGRAM

- What is IoT
- History of IoT
- How Easy is It
- What Makes A Good IoT Product
- Lab Time

ENVENTYS PARTNERS

What Is IoT?

IOT

3 Layers

Application/Network/Perception

Application Layer
(The Device)

Network Layer
(The Transport)

Perception Layer
(The App)

Application

What you are doing

- Sensors (Temp, humidity, light, buttons, etc.)
- Actuators (motors, lights, speakers, etc.)

Network

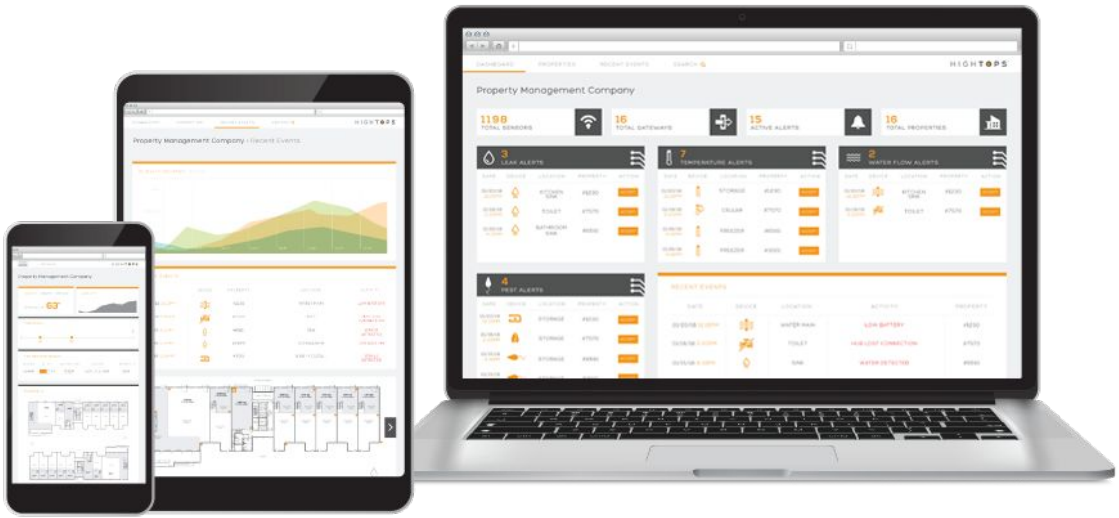
How you transfer the data

- Ethernet
- WiFi
- Bluetooth
- Zigbee

Perception

The App or dashboard

- Web
- Mobile
- Tablet



All Together

ENVENTYS PARTNERS

History of IoT

Kevin Ashton

Coined the term Internet of Things in 1999 while working at Proctor and Gamble

- “The Internet of Things is about empowering computers...so they can see, hear and smell the world for themselves.”

128.2.209.43

Coke is IoT!

- In 1982, engineers at Carnegie Mellon connected a Coke machine to their network to monitor the temperature and inventory of the machine.

Carnegie Mellon University

Toast

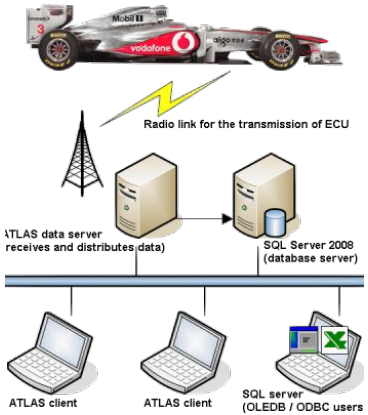
1990. Simon Hackett and John Romkey connect the first commercial product to the internet (not a computer)

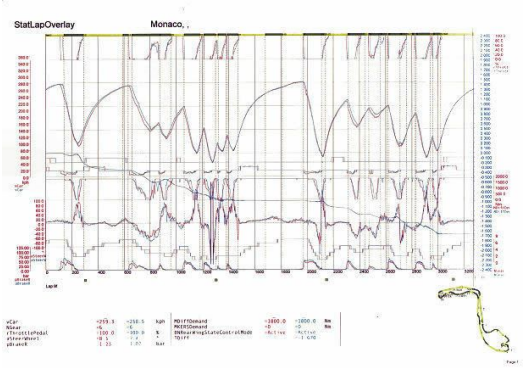
- Could be remotely turned on and off. Darkness of toast controlled by on-time
- In 1991 they added a crane to take the toast out.

Racing

Formula 1 Cars become mobile IoT labs

- First car to pitwall transmission in the late 80s
- By the 90s, bi-directional wireless data achieved

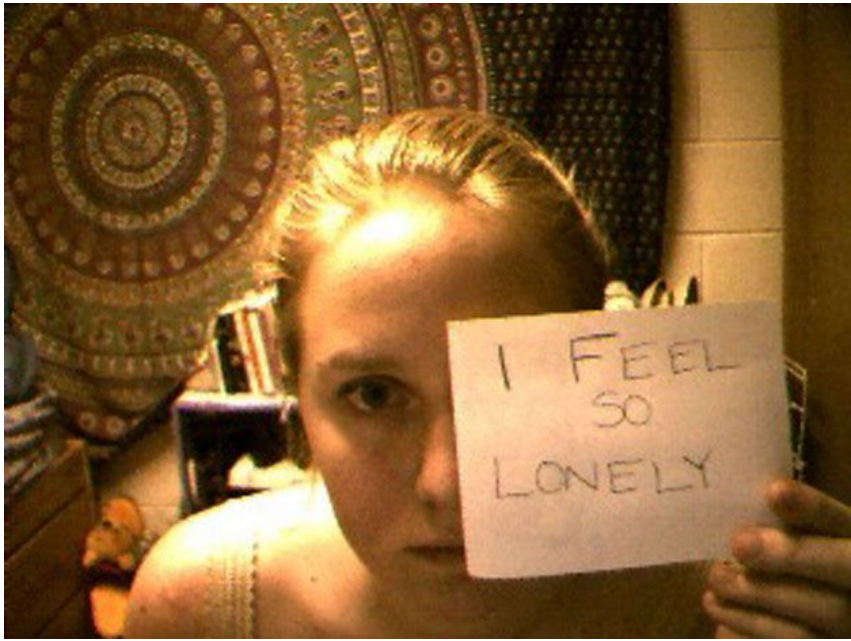




JenniCam

In 1996 Jenni Ringley uses her webcam to become the first internet celebrity using IoT.

- Broadcast her entire life in 15s increments
- World-wide attention lands her on Letterman



In the Year 2000!

LG Internet Digital DIOS IoT Refrigerator Debuts.

Took 55 people and a budget of ~$45 million

- Measured temperature and freshness of food
- Integral webcam

Ambient Orb

Tracked stock prices and displayed color based on price

- MIT project that was launched in 2002.
- Still in production

Fitness

IoT gets buff

- 2006. Nike+ connects to iPod to measure running stats
- 2009 Fitbit announces their first fitness tracking device

Smartphones

Our necks have not been the same since

- 2007. First gen iPhone released
- A pocket full of possibilities

Smart Home

The call is coming from inside the house!

- 2011. Nest gen 1 launches
- Slew of smart home devices follow
- Amazon Echo released in late 2014

How Easy Is It?

Super Easy

You can be on the web in minutes

- Developer kits make it very easy to push data to the cloud
- Many boards are compatible with Arduino code and libraries
- Sample code is plentiful

Fire on Demand

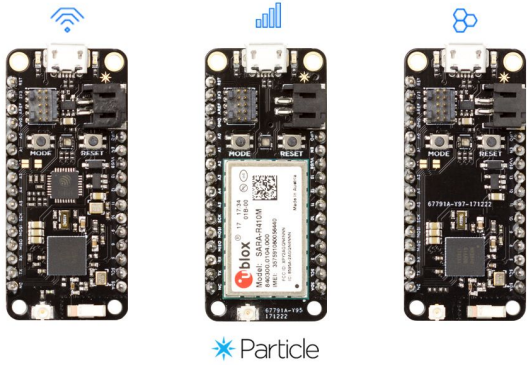
No developer required

- Services like Blynk make it very easy to build a custom app for Iot hardware
- Integral solution that requires very little code for the device

And It’s Cheap

Less than a venti mocha latte

- Wifi+ BTLE in a $2.50 module
- Developer hardware for under $20
- Cellular data plans for $3/month



ENVENTYS PARTNERS

What Makes A Good IoT Product?

What Makes a Good IoT Device

Smart, Adaptable, Human centered, and Solve a Real Problem

- Strong Need
- Positive Cost/Benefit
- Robust architecture
- Easy to use

Strong Need

Don’t do it just because you can

- Think of the Why, before the “I”

*From https://iout.rehabagency.ai/

Benefit has to outweigh the cost

IoT versions of the same device can cost 5-10X their analog components

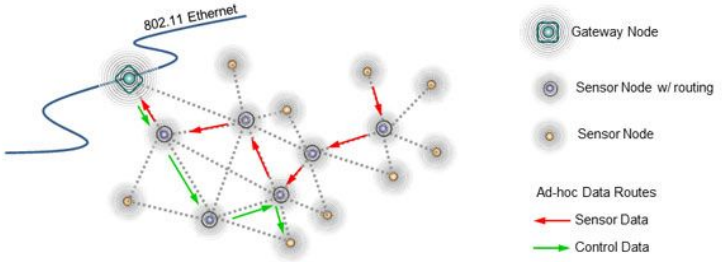
- Not to mention the time required to set them up

$215.99	$59.95
$99.97	$12.60
$179.94	$39.99
$48.50	$12.99
$49.99	$0.00
$594.39	**$125.53**
RD$31,772	**RD$6,686**

Robust Architecture

Reliability is key to adoption

- Communication needs to work all the time
- Data has to be accurate
- Actuators need to work
- Adaptive networks



Easy to Use

Everyone wins when the UI matches the need

- Clear and intuitive UI
- Quality electromechanical assembly
- Easy installation

ENVENTYS PARTNERS

Easy-IoT Platforms

Goldilocks quality

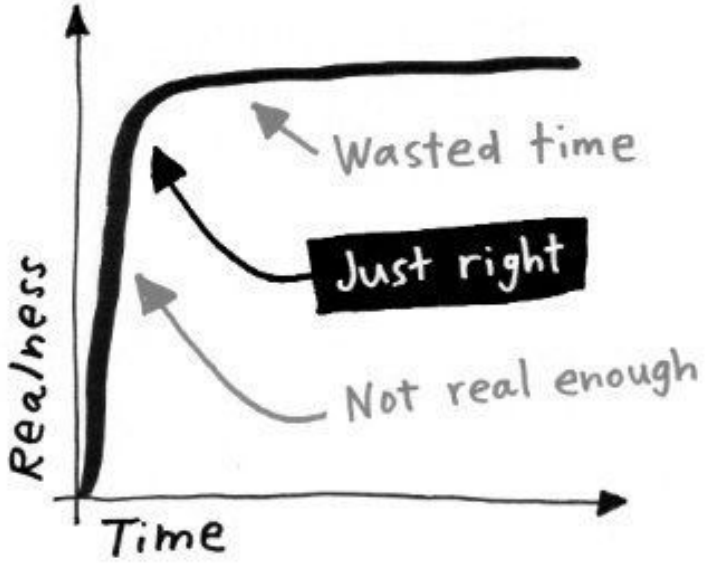


Image from *Sprint* by Jake Knapp

Goldilocks IoT

We can use pre-built tools like Tinker and Blynk to do minimal code with the maximum impact.

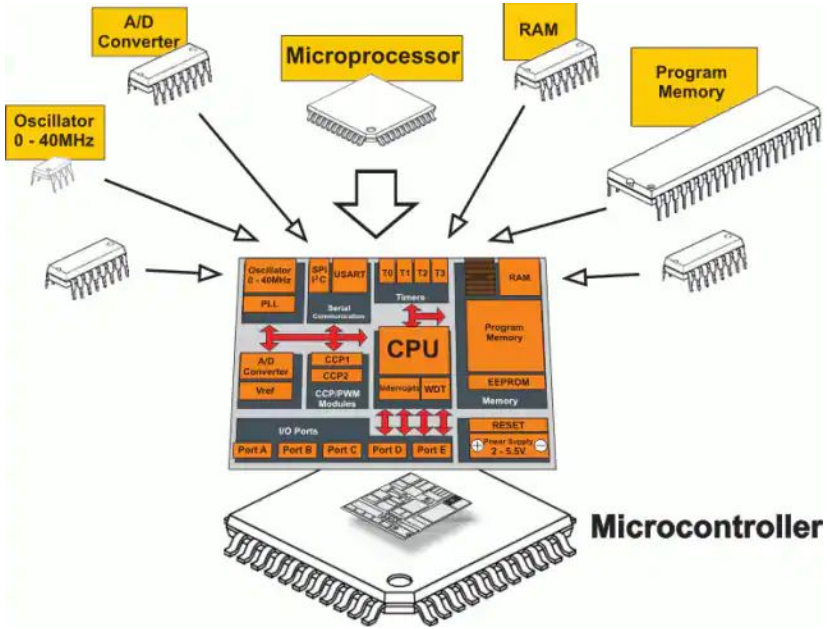
ENVENTYS PARTNERS

Microcontroller Basics

What is a microcontroller?

Mini computer

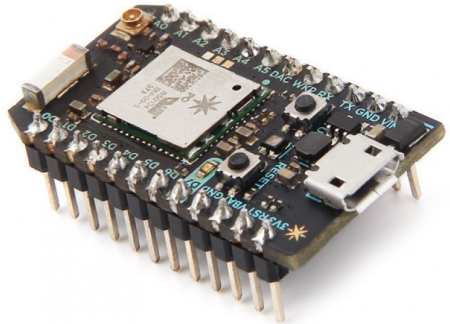
Processor, memory, and I/O in one integrated circuit



https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller

Development Board

Microcontroller that is expanded out onto a PCB that that can easily be used for prototyping. Often too large for incorporation into a consumer product, but useful for prototyping.

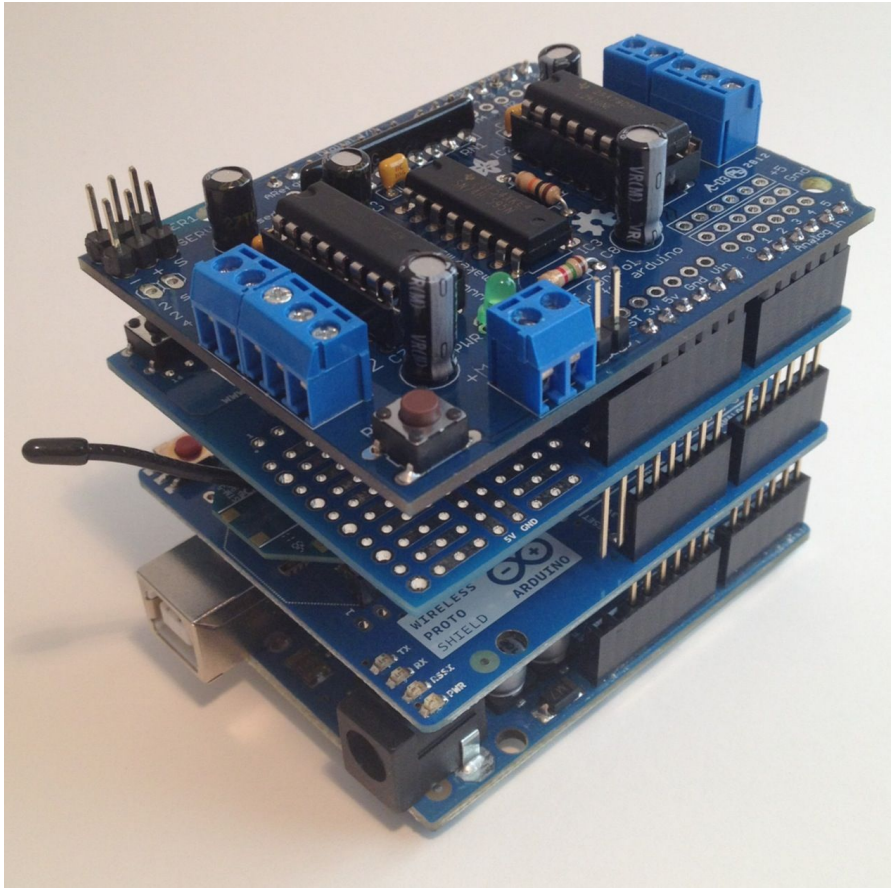


GreenIQ

Exception that proves the rule

Arduino

Arduino was first major platform that brought microcontrollers to the masses. Allowed the use of stackable “shields” to simplify wiring.

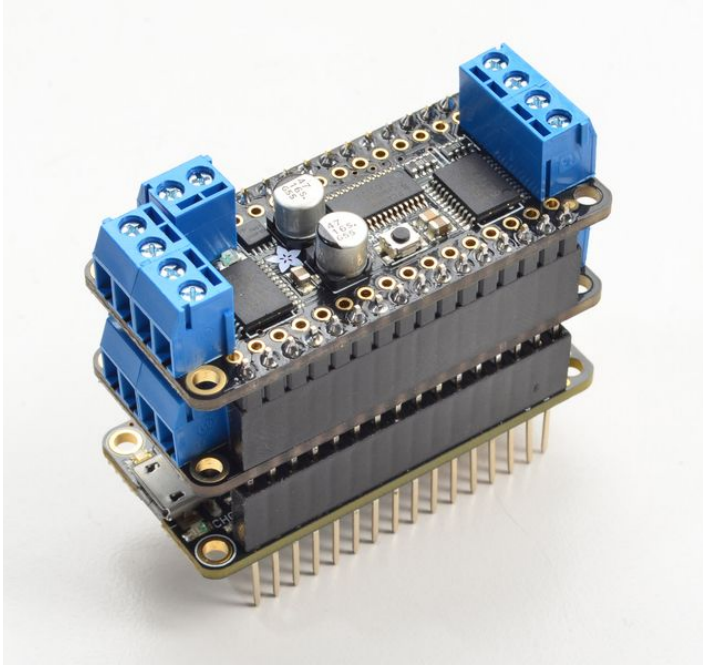


Feather

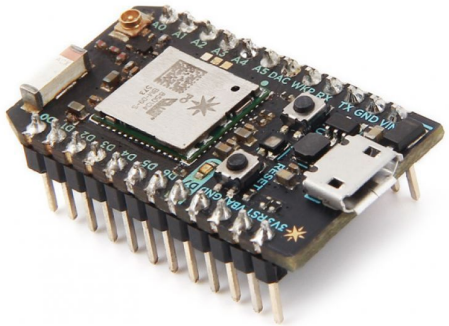
Feather platform is becoming the new standard for dev board footprints.

Most IoT dev boards are feather compatible.

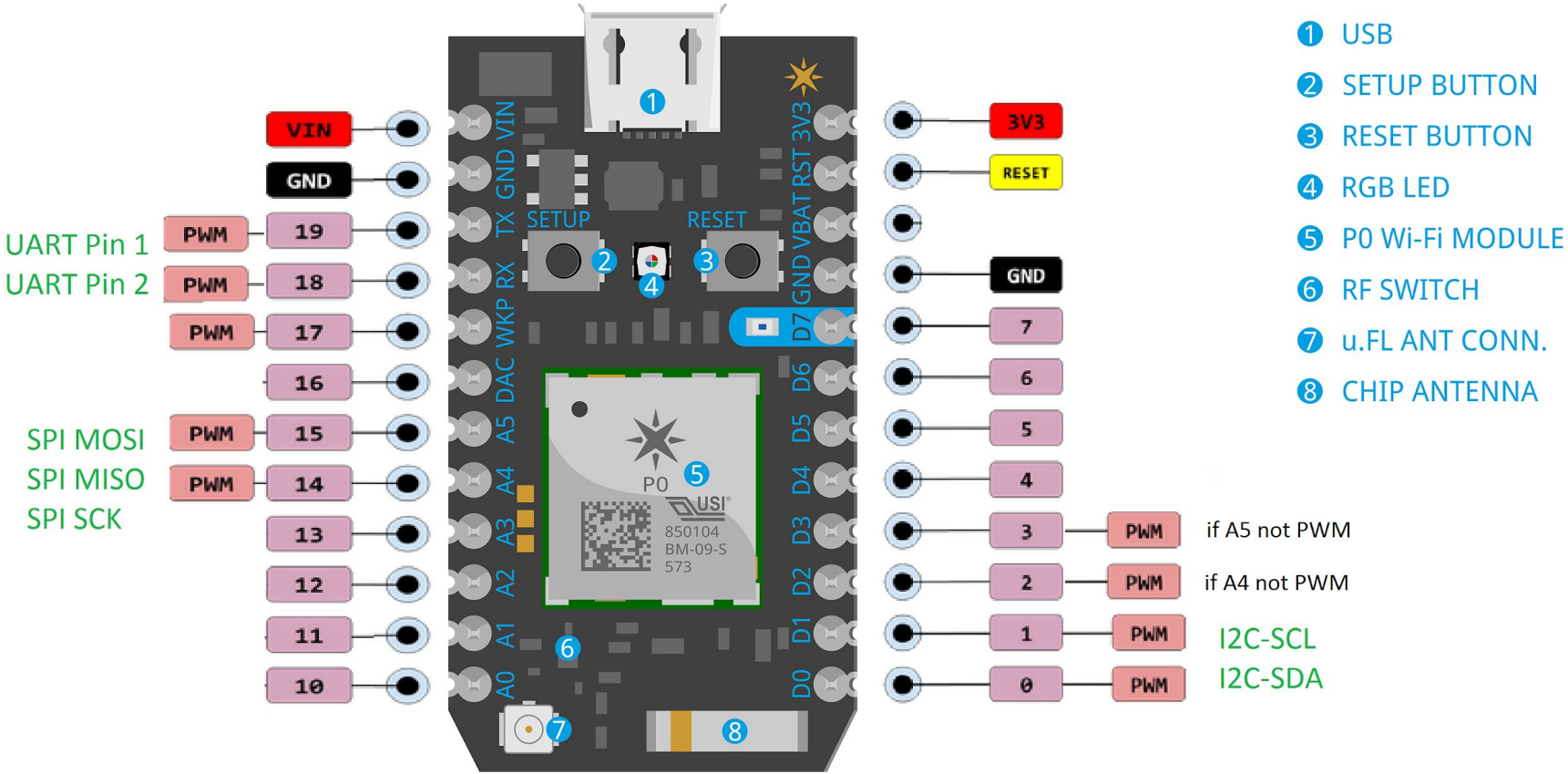
The Photon is not, but their new sister boards are.



Particle Photon



- Particle PØ Wi-Fi module
  - Broadcom BCM43362 Wi-Fi chip
  - 802.11b/g/n Wi-Fi
  - STM32F205RGY6 120Mhz ARM Cortex M3
  - 1MB flash, 128KB RAM
- On-board RGB status LED (ext. drive provided)
- 18 Mixed-signal GPIO and advanced peripherals
- Open source design
- Real-time operating system (FreeRTOS)
- Soft AP setup
- FCC, CE and IC certified


1 USB
2 SETUP BUTTON
3 RESET BUTTON
4 RGB LED
5 P0 Wi-Fi MODULE
6 RF SWITCH
7 u.FL ANT CONN.
8 CHIP ANTENNA
VIN
GND
UART Pin 1
UART Pin 2
PWM 19
PWM 18
PWM 17
16
SPI MOSI
SPI MISO
SPI SCK
PWM 15
PWM 14
13
12
11
10
SETUP
RESET
P0
850104
BM-09-S
573
3V3
RESET
GND
7
6
5
4
3 PWM if A5 not PWM
2 PWM if A4 not PWM
1 PWM I2C-SCL
0 PWM I2C-SDA

One More Thing

Arduino

- Go to https://www.arduino.cc/en/main/software
- Download latest version of Arduino IDE
- Will use this only for the serial monitor

zAnd download Github Repo

https://github.com/jlosaw/SDQcreates

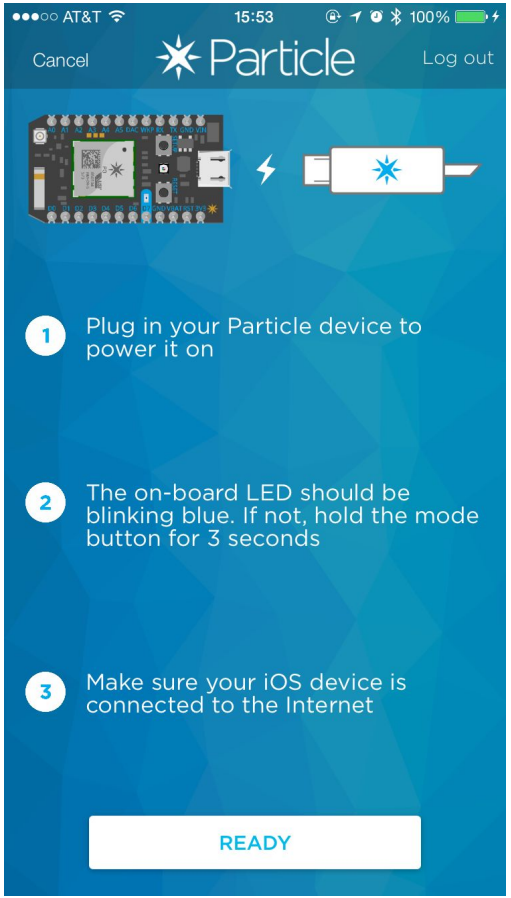
ENVENTYS PARTNERS

Photon Connection and Programming

Particle App

Claim device. Download and follow Setup

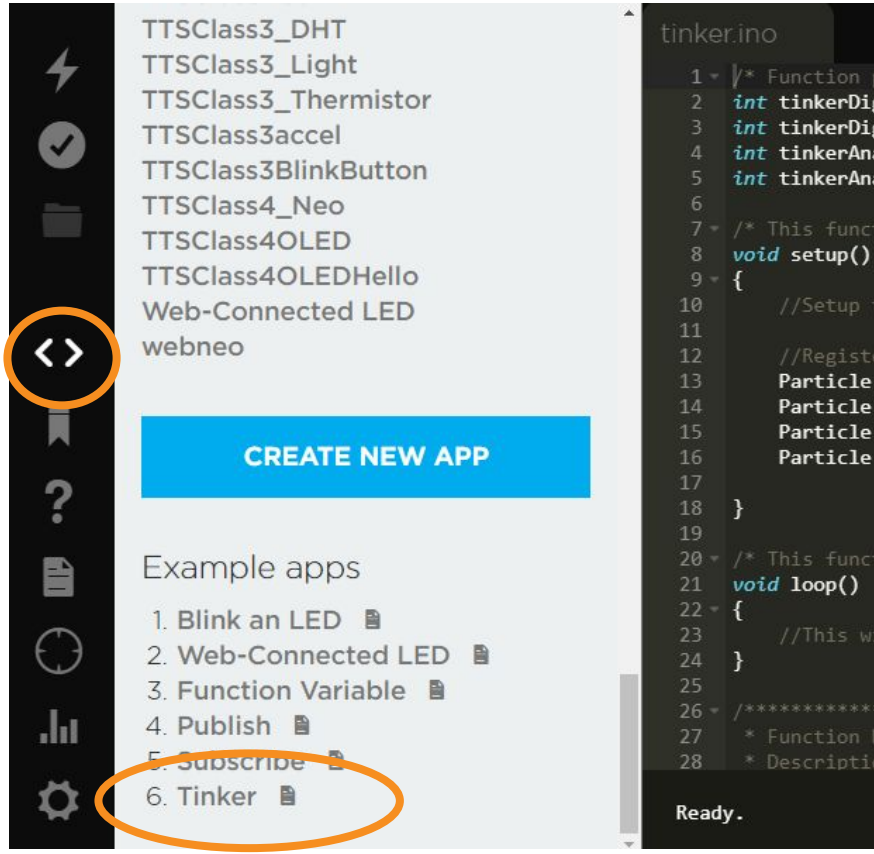
- Setup (run once)
- WiFI
    - ID: EVENTO XOLUCTRONIC
    - PW:EVENTO2020
- Put device in “listening mode” and re-connect
- Photon will hold up to 5 sets of Wifi credentials, so you should not have to reconnect next class



Firmware Update

Photons need to be upgraded to latest firmware

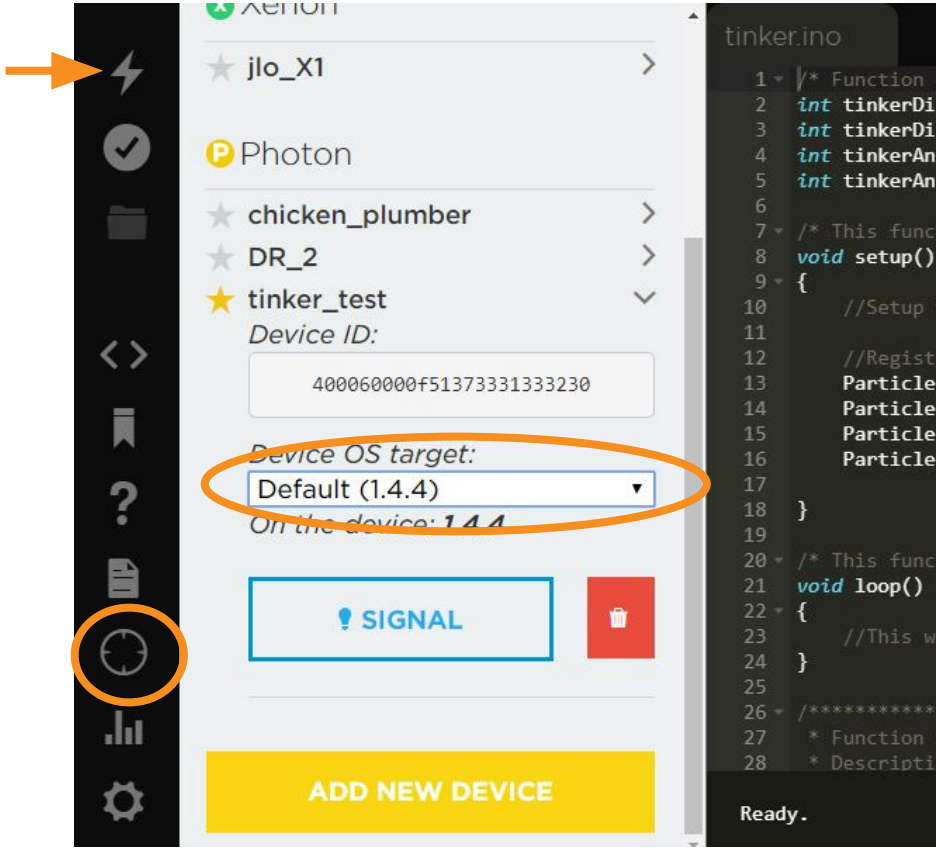
- Open WebIDE from Particle.io
- Click code icon and select “Tinker”



Firmware Update

Step 2

- Select “Devices from the sidebar menu
- Set “Device OS target” to latest stable release (1.4.4)
- Click the lightning bolt icon to update firmware and flash the code



ENVENTYS PARTNERS

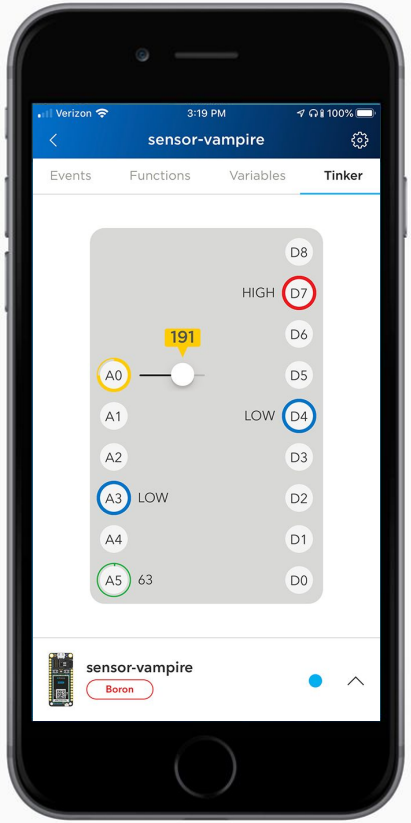
Tinker

Tinker Functions

Can use Particle app to control any of the Photon pins with a single touch.

4 Functions:

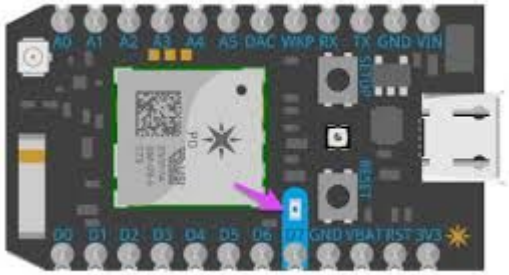
digitalWrite
analogWrite
digitalRead
analogRead

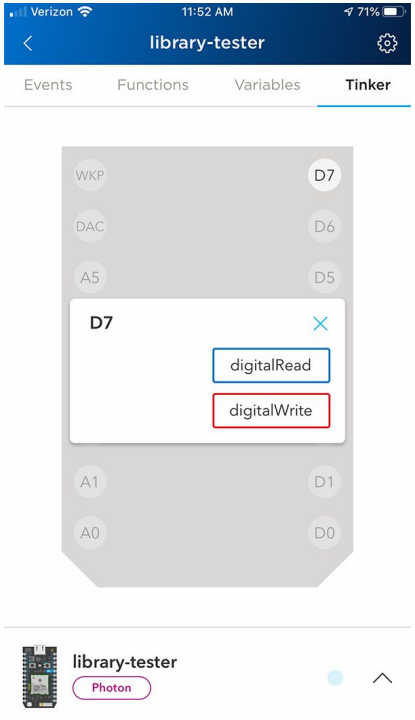


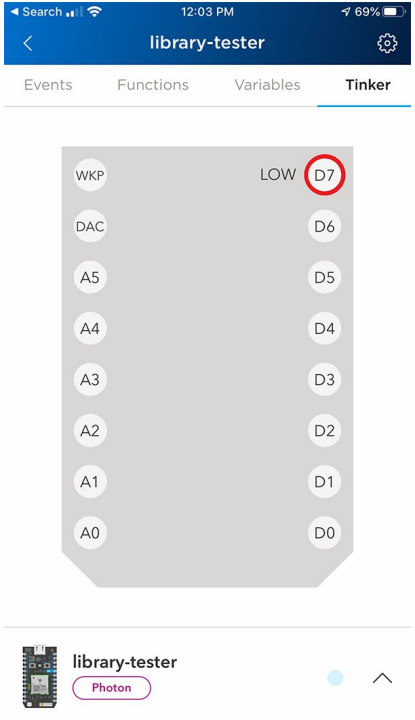
Try it Out

Blink the on-board LED

1: Choose your device.
2: Click the “Tinker” tab
3. Click D7 and set to “digital Write”
4. Toggle the pin to see on-board LED turn on and off



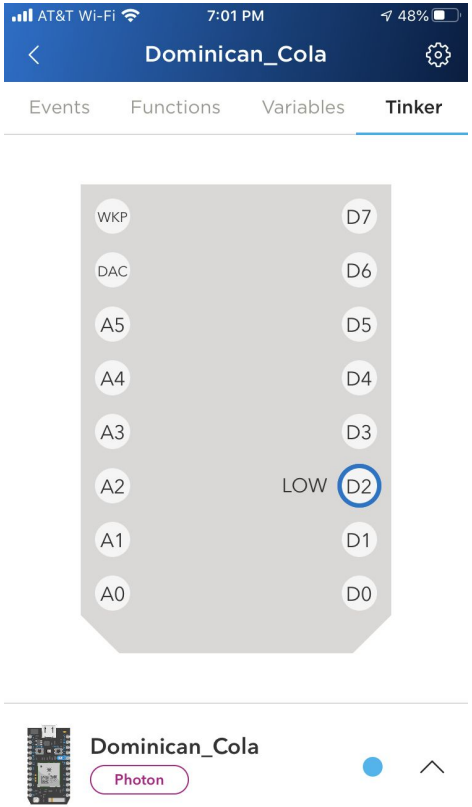




Prototype Button

Read High/Low state of a pin

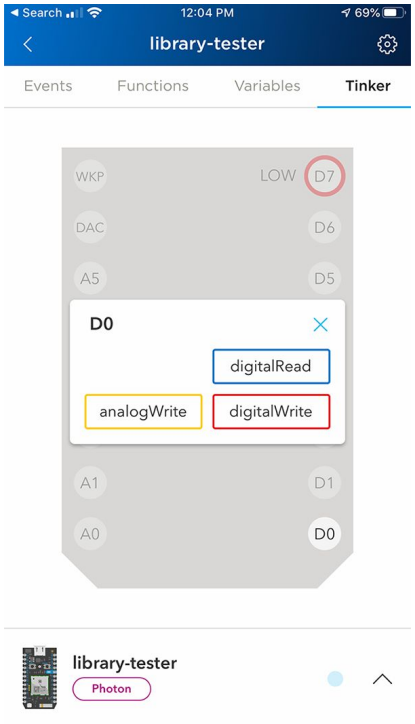
1: Set D2 to “digitalRead”
2: Note that the state is “LOW”
3: Run jumper wire from”3V3” pin to to D2
4: Touch D2 and see how it reads “HIGH”

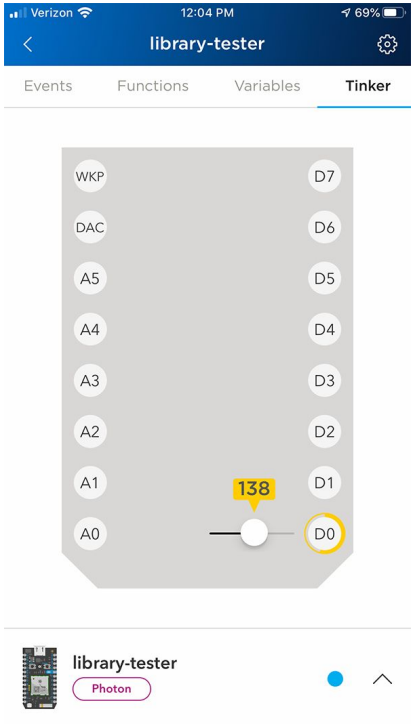


LED Dimmer

Analog Control

1: Connect LED from D0 to Ground
2: Change D0 to “analogWrite” in Tinker
3. Change slider to see LED change brightness

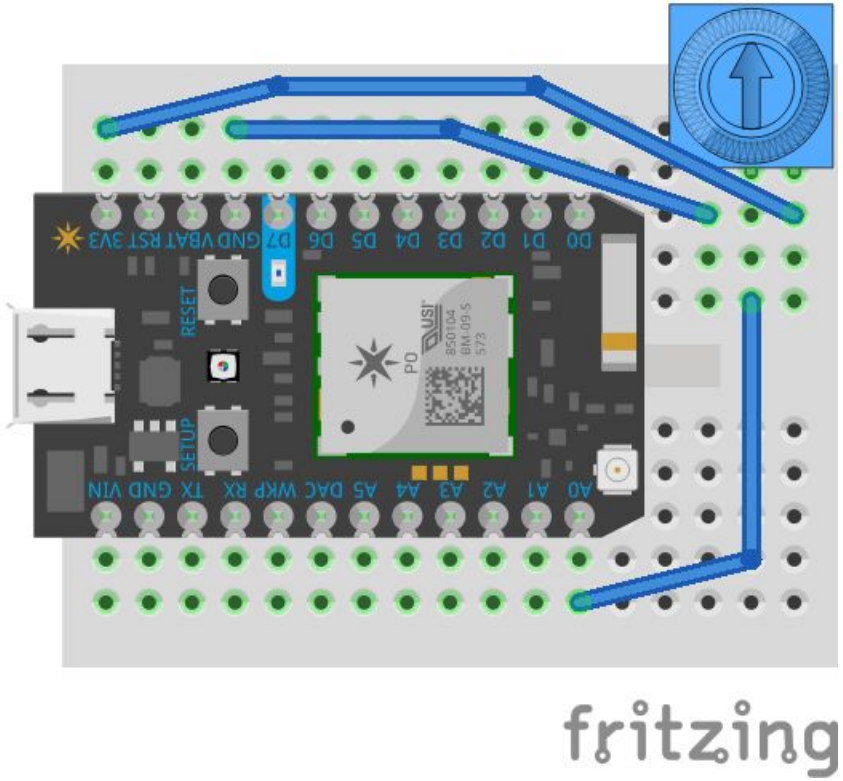




Read Potentiometer

Read a sensor

1: Connect potentiometer to A0
2: Change A0 to “analogRead” in Tinker
3. Turn the dial to change the values



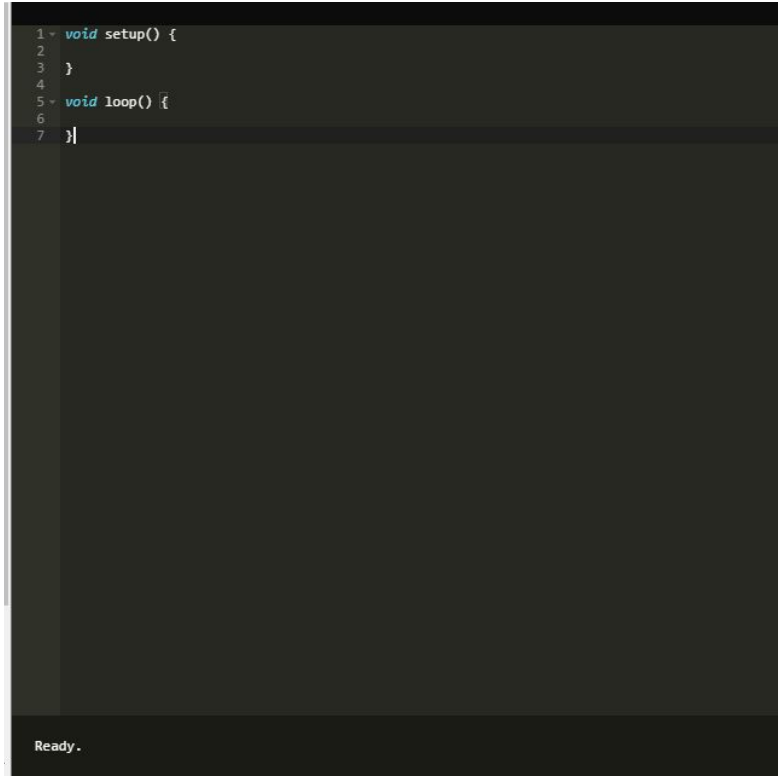
ENVENTYS PARTNERS

Programming The Photon

Modified C/C++

3 Main Parts

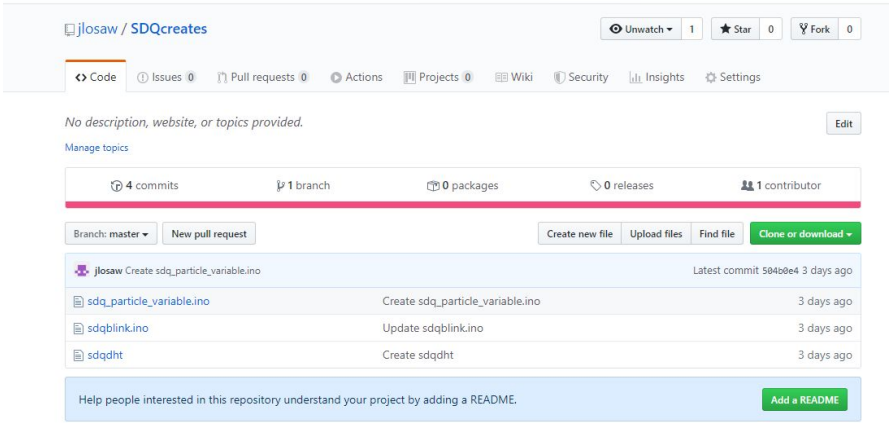
- Declarations
- Setup (run once)
- Loop (run forever)
- Subroutines



GitHub

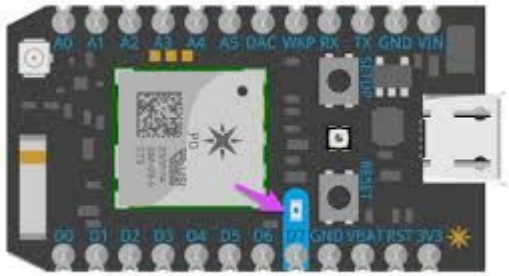
Website with sample code

- https://github.com/jlosaw/SDQcreates
- Sample files for all of our experiments

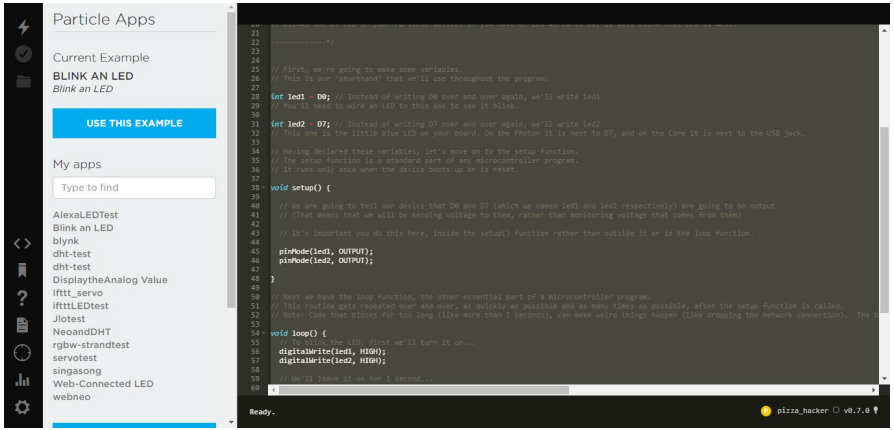


Blink an LED

The “Hello World” of microcontrollers



- Often used for troubleshooting and debugging
- Photon has on-board LED connected to pin D7
- Download **sdqblink.ino** and run code

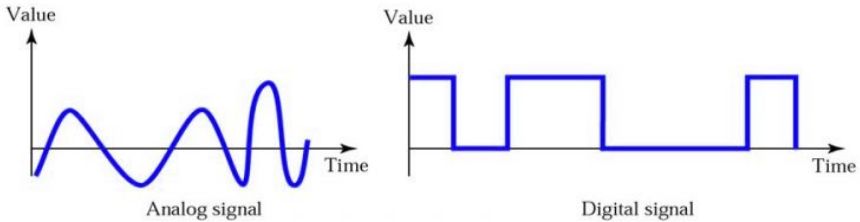


Sensors

Types of Sensors

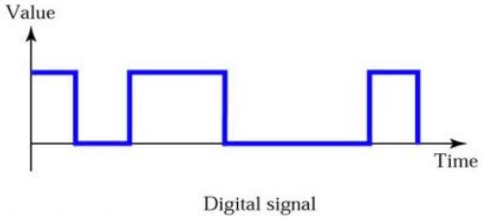
What kinds of signals

- Analog
- Digital
- Resistive
- Current
- Amplified



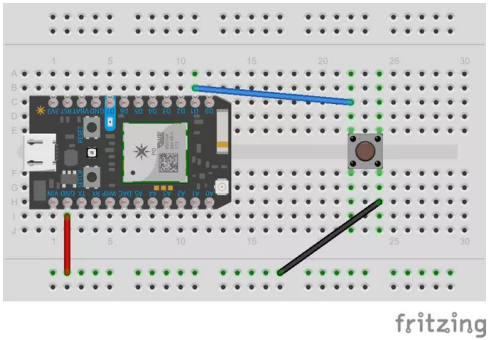
Digital

High or Low



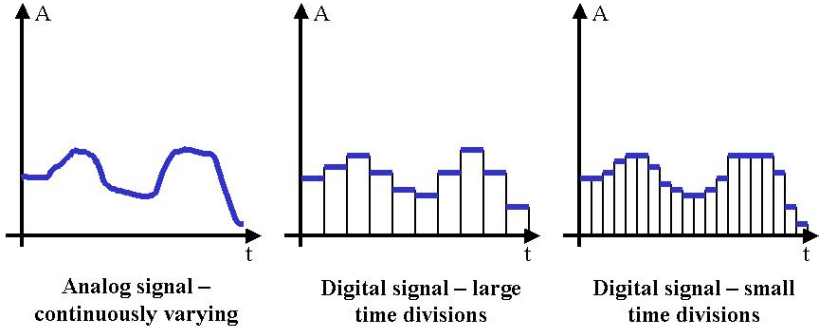
Digital signal

- Voltage high or low, or discrete
- Great for measuring on/off states
- Example: Limit switch on a 3D printer

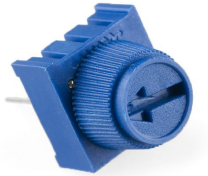


Digital

Discrete values

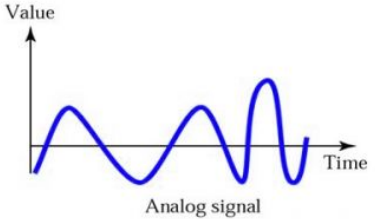


- Accuracy depends on the bit rate of the processor
- Example: 0-100 degree temp sensor with a 10 bit D/A converter reads in increments of roughly .1 degrees (100 degrees/ 1024 values)

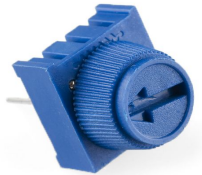


Analog

Signals that are continuously variable



- Voltage output continuously variable with time
- Example: Potentiometer and a gas pedal



Analog to Digital Conversion

How many values can we read?

- Digital is base 2. 0 or 1 are the only values
- For example, 8 bit is $2^{8} = 256$
- The Photon is 12 bit $2^{12} = 4096$

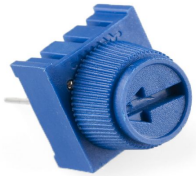
8 bit

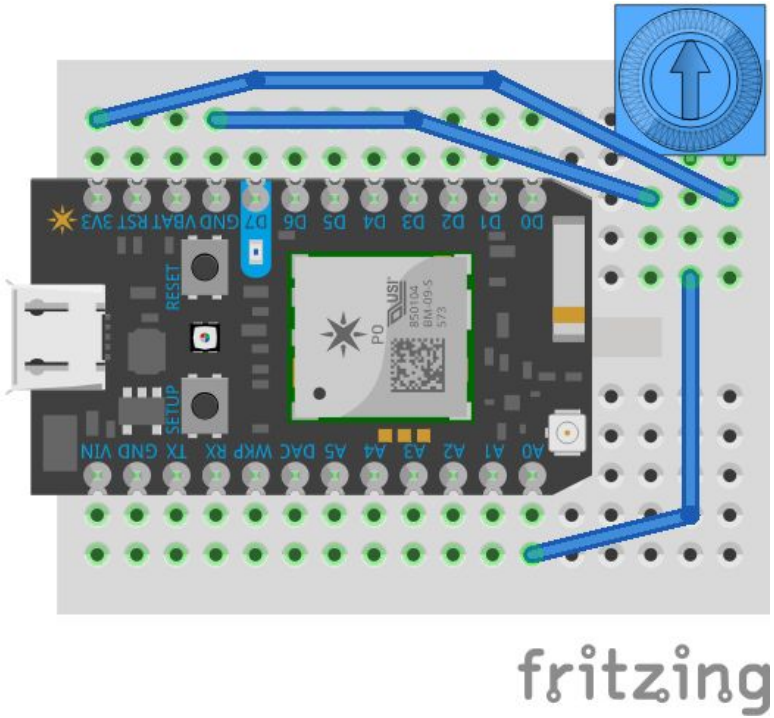
16 bit

256 bit

Potentiometer

Get the analog value from the pot

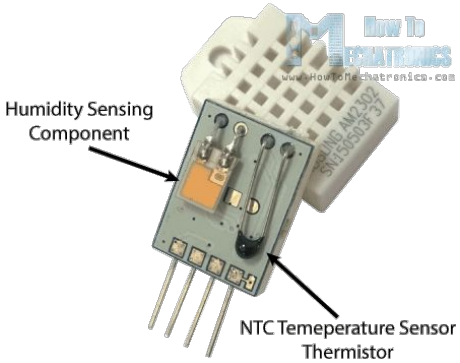


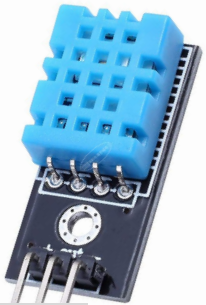


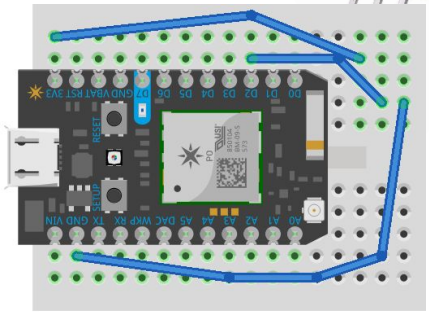
- Download Arduino program
- Download **sdq_potentiometer.ino**
- Hook up the pot to A0 on the breakout board
- Remember, V=IR. We are varying R so the voltage will change
- The voltage output is infinitely variable, but why is output not 0- infinity

DHT11

A more common sensor for temp and also get humidity





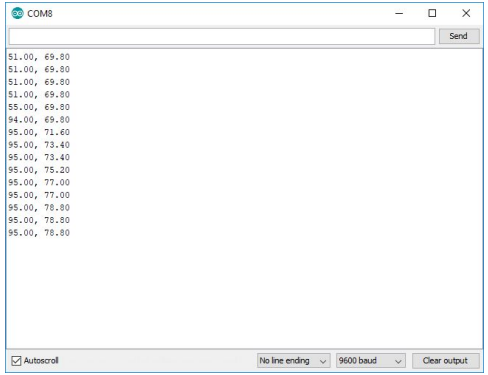


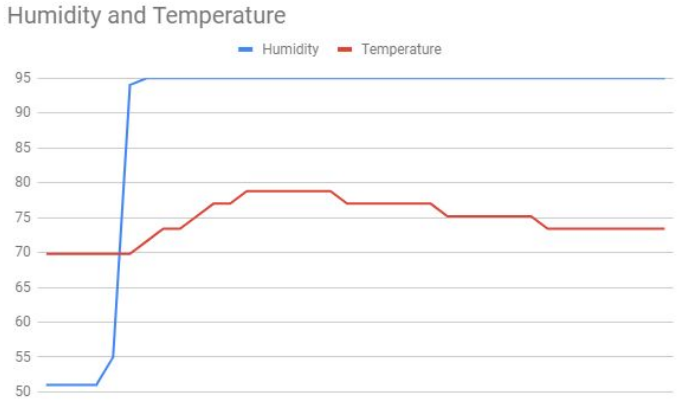
- Inside is an NTC thermistor and an electrode that changes resistance with humidity
- Download **sdqdht.ino**
- Use #include "Adafruit_DHT_Particle.h" library
- Data is aggregated and sent over 1 data wire
- Libraries for the sensor parse the data
- Setup the serial monitor from Ardunio IDE to watch the data stream

Pro Tip

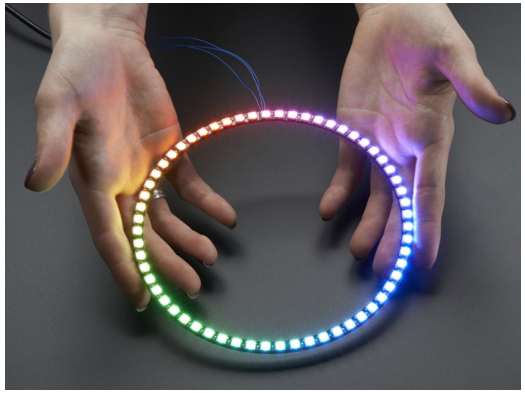
Format serial output for easier plotting

```
Serial.print(h);
Serial.print(", ");
Serial.println(f);
```





Addressable LEDs



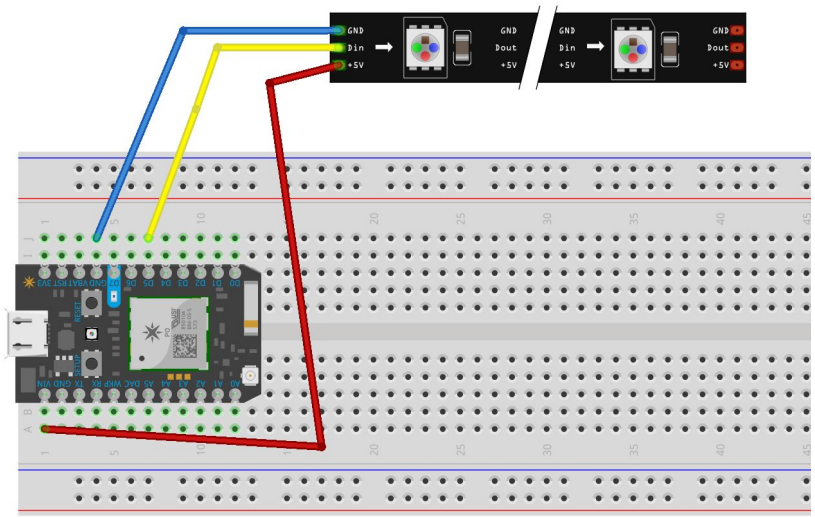
Control each pixel in a chain or grid

- Each LED has a small control chip that takes commands from the controller, displays its own value and then passes the next instruction down the line
- Allows for more complicated effects and customized output
- Comes at a price
- Two main types
    - Neopixels (3 wire)
    - Dotstars (4 wire)

Double Rainbow

Test out some samples from the library

- Use hookup wire to connect ring to data line to D5 on the board.
- Need 5V so hookup power to Vin
- Download and run **sdq_addressableLED.ino**
- Don't forget to re-add the Neopixel library
- Push the sample code to the board and note the effects



ENVENTYS PARTNERS

Particle Platform

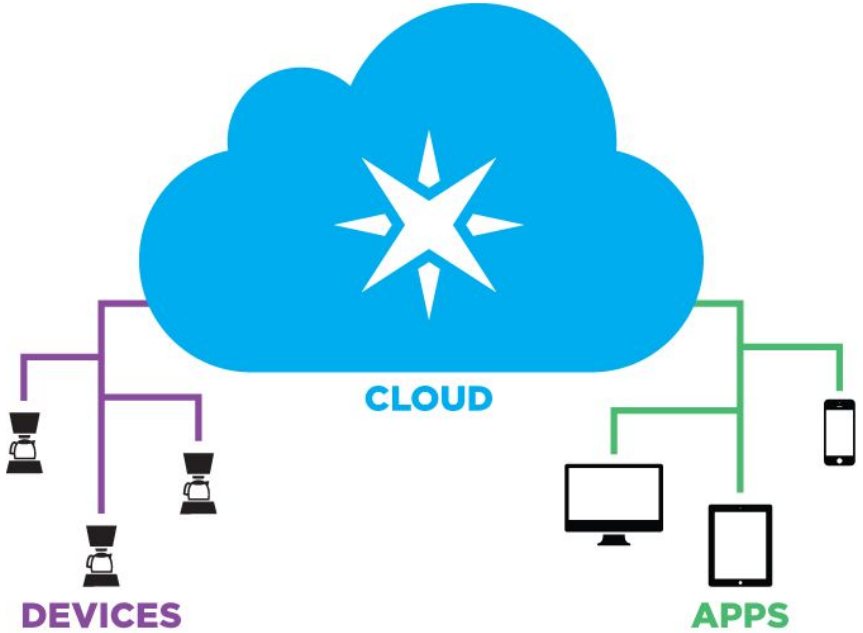
ENVENTYS PARTNERS

Cloud Functions

Particle Cloud

4 Ways to interact with the Photon over the web

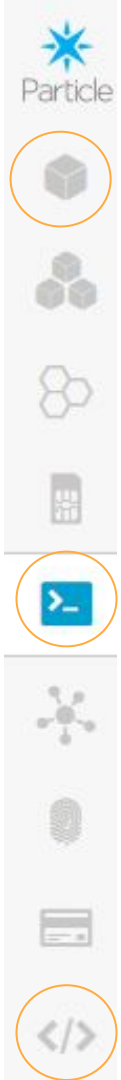
- Particle Functions
- Particle Variables
- Particle Publish
- Particle Subscribe
- Reference docs are here: https://docs.particle.io/reference/device-os/firmware/photon/
- There are more, but this is what we are going to focus on

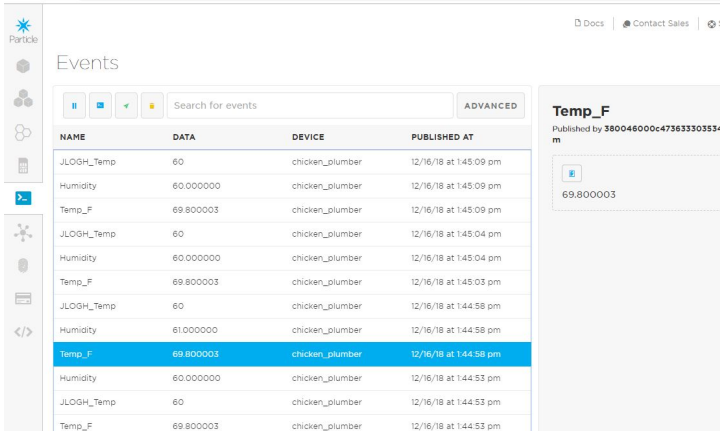


Particle Console

Your dashboard to see what is going on with your device

- **Devices**
- Products
- Networks (for mesh setups)
- Sim Cards (for LTE chips)
- **Events**
- Integrations
- Authentication
- Billing
- **Web IDE**





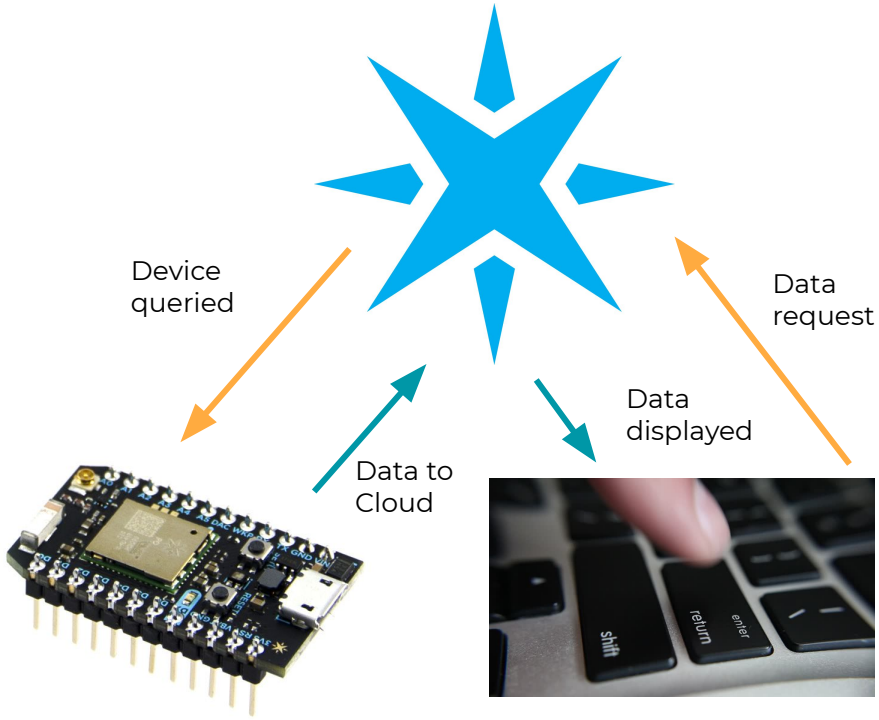
ENVENTYS PARTNERS

Particle.variable()

Particle Variable

Query data from a device through the cloud

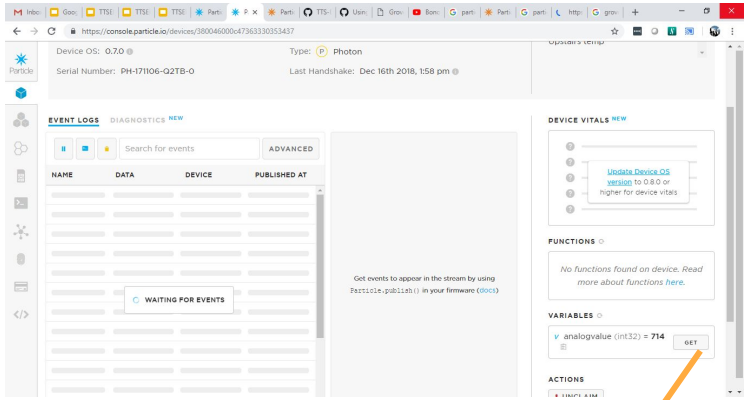
- Opens a pipeline between the cloud and a device
- Does not constantly upload data to a dashboard
- Only allowed 20 variables and no more than 12 characters
- There are three supported data types:
    - INT
    - DOUBLE
    - STRING (maximum string length is 622 bytes)

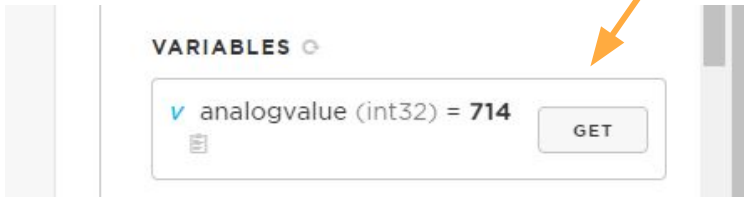


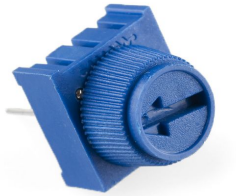
Particle Variable

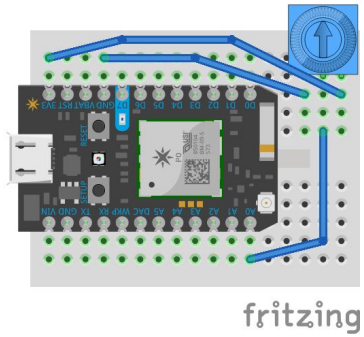
Query data from console to get the value of the rotary pot

- Open **sdq_particle_variable.ino** from the repo
- Plug rotary pot into A0
- Upload code
- Login to your Particle Console
- Click the “Get” button on your variable to query the sensor and retrieve analog value





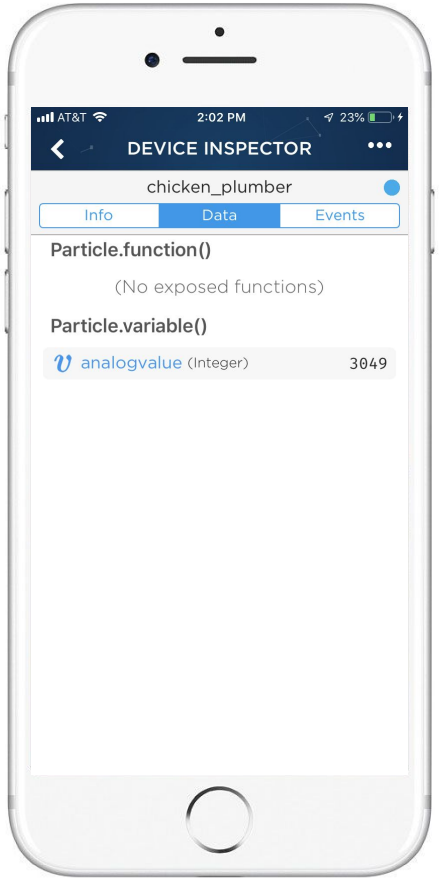




Particle Variables with App

Make the query with your phone

- Open Particle app
- Navigate to your Photon
- Tap the “v” under the variable to retrieve the data



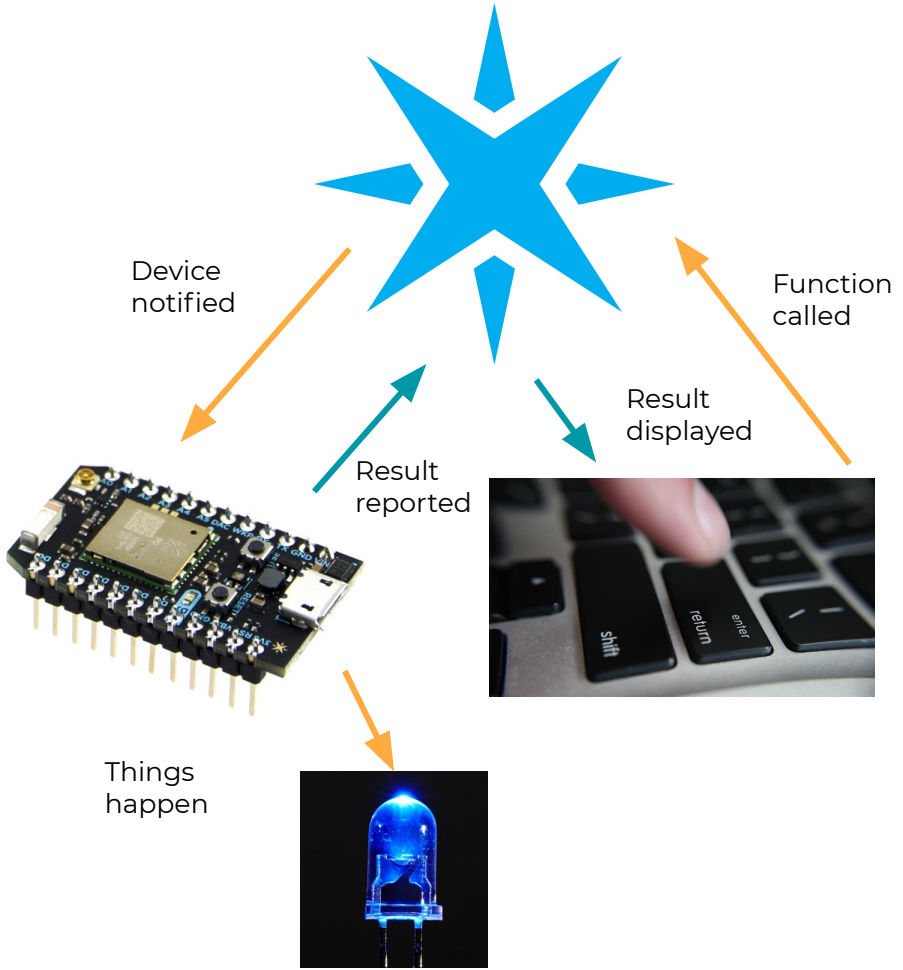
ENVENTYS PARTNERS

Particle.function()

Particle Function

Triggers a Particle device to run a routine

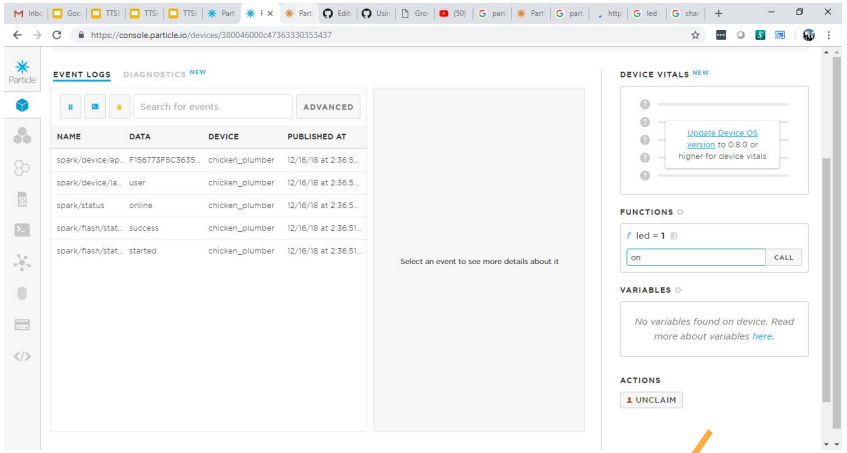
- Good for triggering an actuator
- Only allowed 15 variables and no more than 12 characters per name
- A cloud function is set up to take one argument of the String data type. This argument length is limited to a max of 63 characters.
- Returns an integer value based on the result
    - A non recognized input returns -1

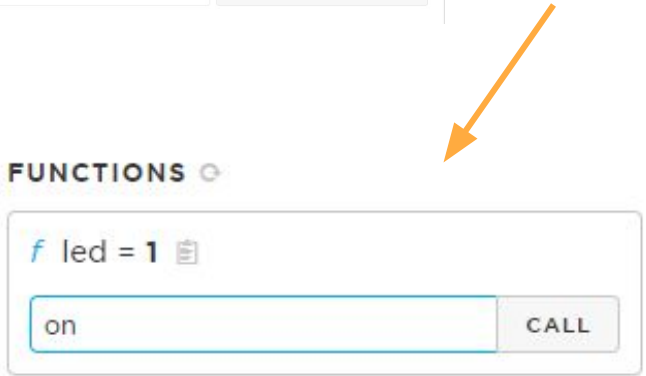


Particle Function

Blink LED on command

- Open **sdq_LED_function.ino** from the repo
- Upload code
- Login to your Particle Console
- Type “on” or “off” into the function box to control LED

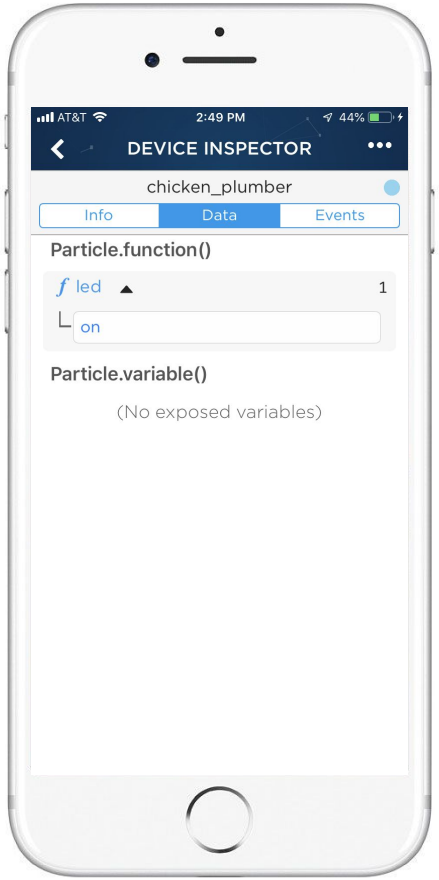




Particle Functions with App

Make the command with your phone

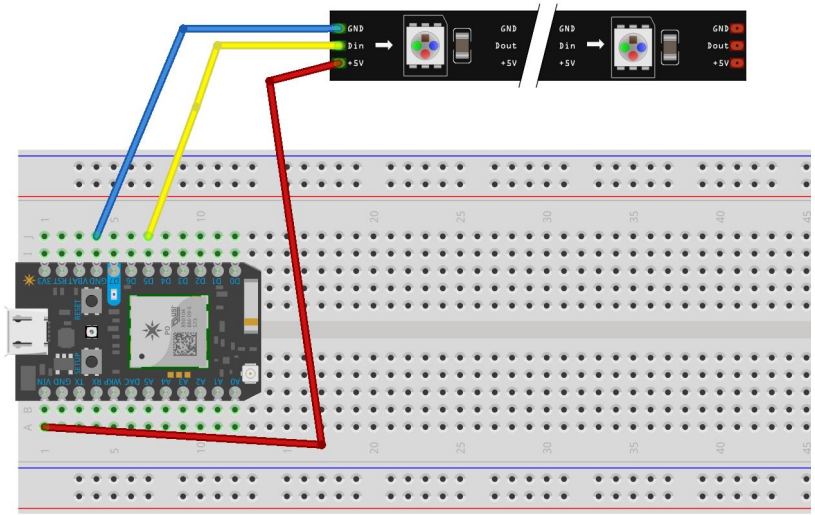
- Open Particle app
- Navigate to your Photon
- Type in the command to drive the output



Make it Rain

Let’s hook up the neopixel ring and have some more fun

- Open **sdq_function_neo.ino** from the repo
- Hook up neopixel to D5
- Use “rain”, “color”, and off commands
- Class challenge: add a 4th function to drive a different effect



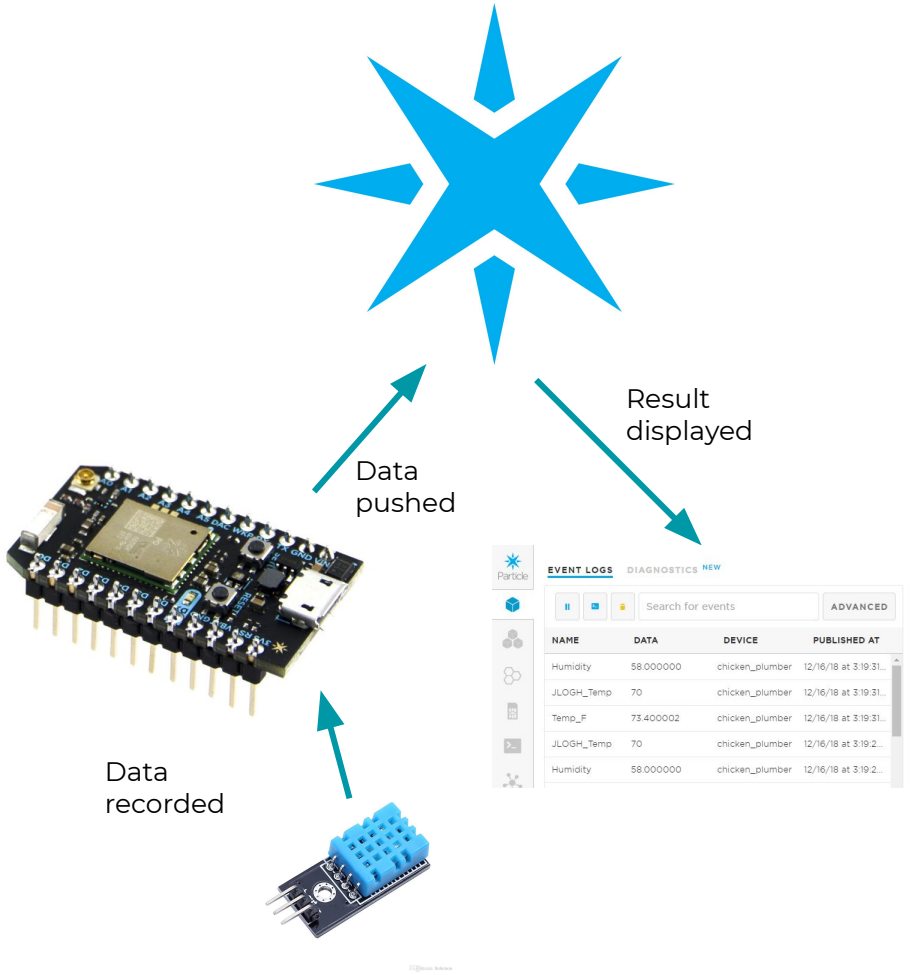
ENVENTYS PARTNERS

Particle.publish()

Particle Publish

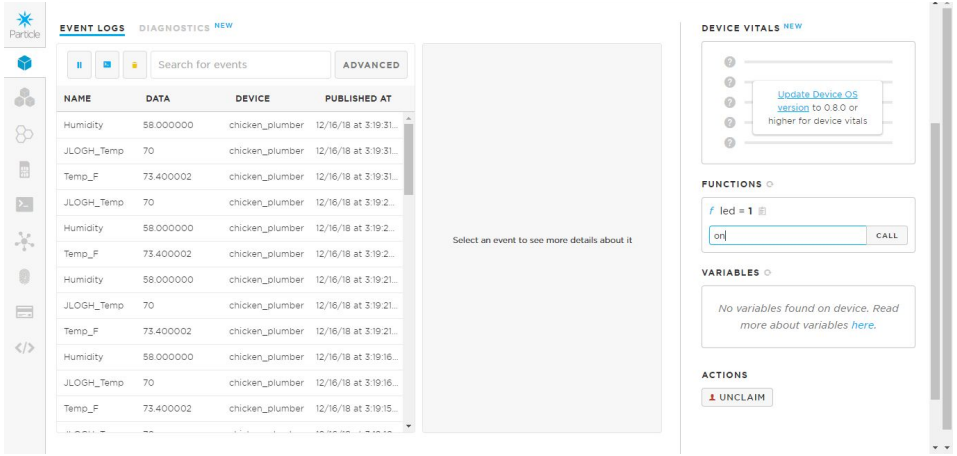
Can push data continuously to the console

- Uses just a single line of code
- Event is forwarded to all registered listeners
- Pushes the value out of the device at a time controlled by the device firmware
- Publish events can be public or private
- Publish events can also be pulled if requested
- Always returns a string, so numeric data needs to be formatted on Photon firmware before pushing

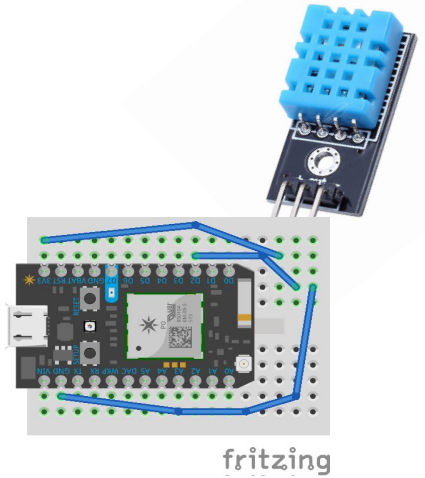


Particle Publish

Watch the data stream in



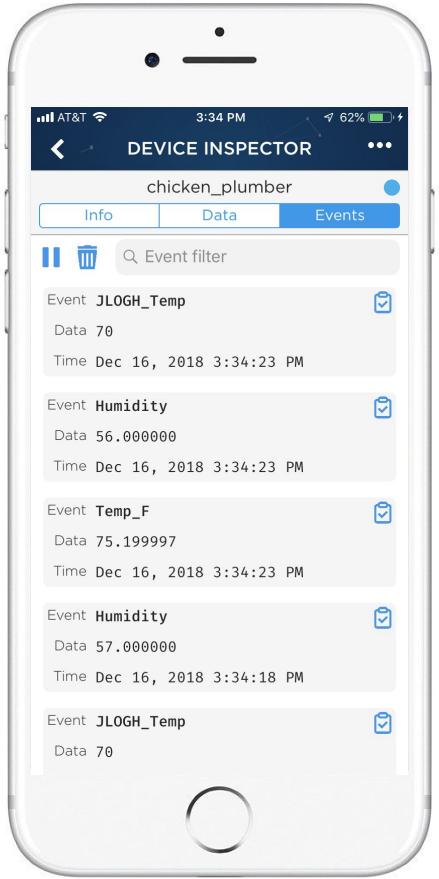
- Open **sdq_temp_publish.ino** from the repo
- Plug DHT11 into D2
- Add the following code after the serial print
    - //Publish the temp data to the console
    - Particle.publish("Temp C",String (t),60,PRIVATE);
    - Particle.publish("Humidity",String (h),60,PRIVATE)
- Upload code
- Login to your Particle Console
- Watch data stream in



Particle Publish with App

Watch your data live from anywhere

- Open Particle app
- Navigate to your Photon
- Select the “Events” tab and watch the data



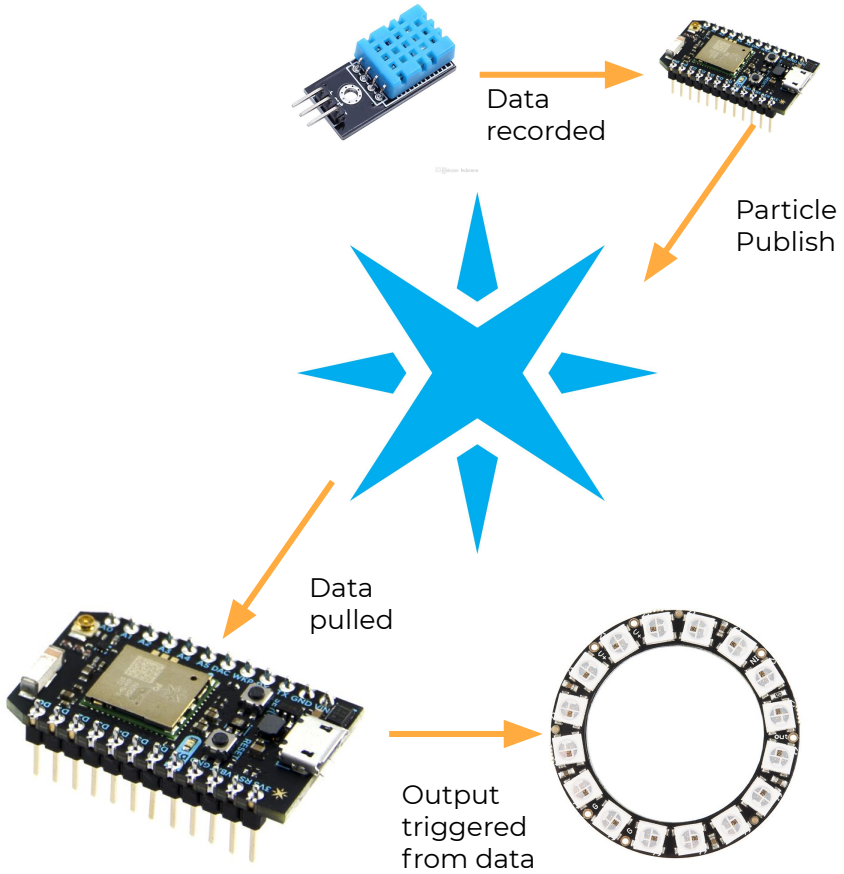
ENVENTYS PARTNERS

Particle.subscribe()

Particle Subscribe

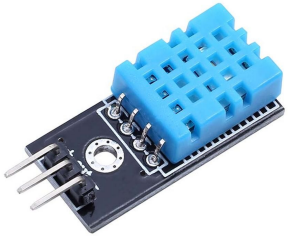
Listen in for events published to the cloud

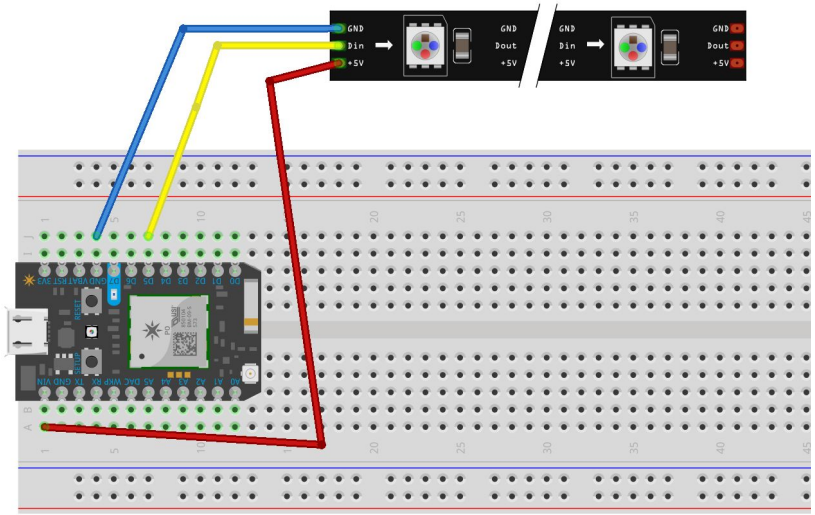
- Turns the Photon into a listening device that looks for events published to the particle cloud
- You can call Particle.subscribe() a maximum of 4 times; after that it will return false.
- Events are forwarded to all registered listeners so you can have many listening devices
- Need 2 devices to pull this off



Particle Subscribe

Listen in to another device





- Open **sdq_temp_subscribe.ino** from the repo
- Plug neopixel ring into D5
- Upload code
- Watch it light up when Jeremy's device reports temperature

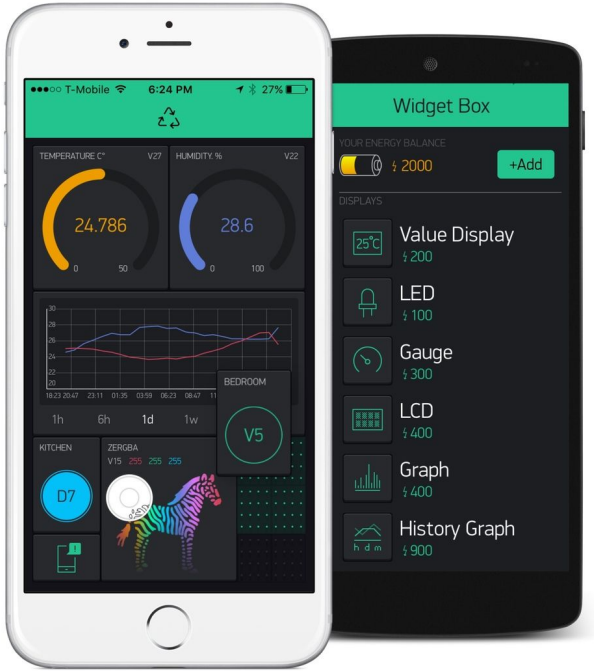
ENVENTYS PARTNERS

Easy-IoT Platforms

Blynk

Blynk

But what if I want more control? Blynk can unlock the power of the Photon.



Blynk Codebase

Authorization Token, Wifi Credentials, BOOM TOWN!

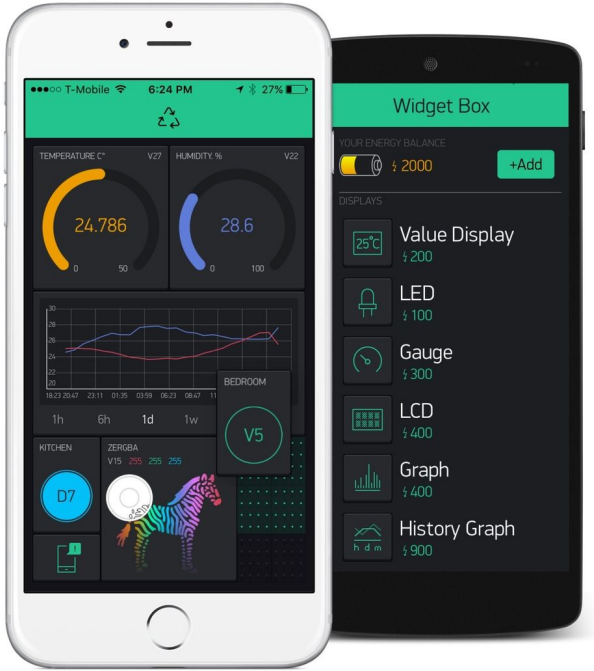
bareminimumblynkcode.ino

```
1  // This #include statement was automatically added by the Particle IDE.
2  #include <blynk.h>
3
4  // This #include statement was automatically added by the Particle IDE.
5  #define BLYNK_PRINT Serial
6
7  char auth[] = "[redacted]";
8
9  void setup()
10
11 {
12     Serial.begin(9600);
13
14      delay(5000); // Allow board to settle
15
16     Blynk.begin(auth);
17
18 }
19
20 void loop() {
21
22     Blynk.run();
23
24 }
25
```

Blynk Setup

Get logged in

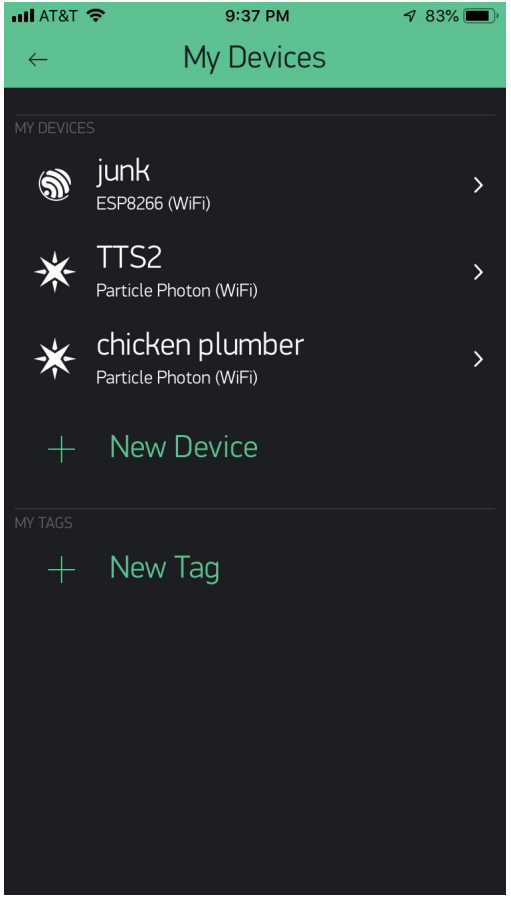
- Download Blynk app from app store
- Create an account
- Note that you get an “energy” balance to add widgets to your app



Add a Device

Connect the Photon to your app

- Click Settings>Devices> New Device
- Scroll down to “Photon”
- Have Blynk email your authentication token (it’s way too long to type in directly)



Load the Code

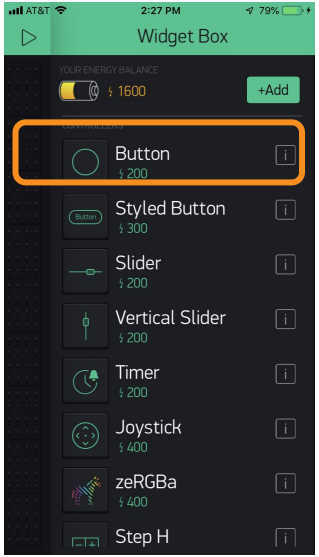
Connect your Photon to Blynk

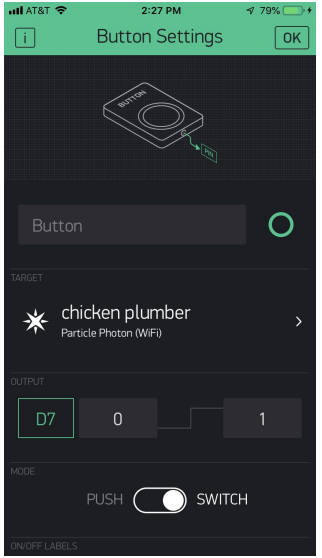
- Upload **sdq_blynk.ino**
- This is all the code you need to start experimenting

```
#include <blynk.h>

#define BLYNK_PRINT Serial //#define
BLYNK_DEBUG


char auth[] =
"2fd02368df2749cbabafb6053834d03f";

void setup()
{
   Blynk.begin(auth);
   delay (1000);
}

void loop()
{
   Blynk.run();
}
```

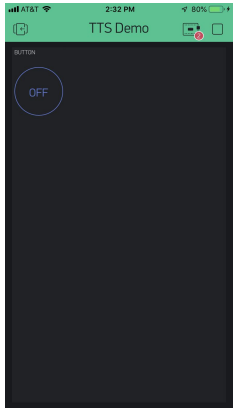
Add a Button

Blink the on-board LED to confirm it works





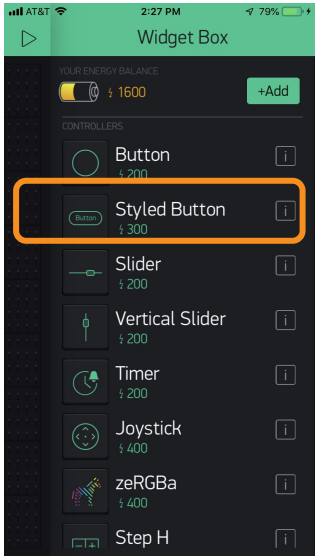
- Click the “nut” icon and select the “button” icon
- Set the output pin to D7
- Click OK to add it to your app
- Click the play button on the upper right to deploy the app
- Push the button and watch the LED light up

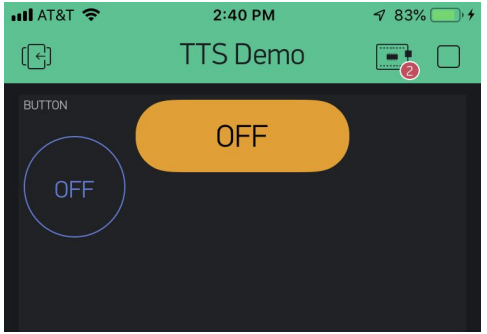


Stylized Button

Expanded options but costs more energy

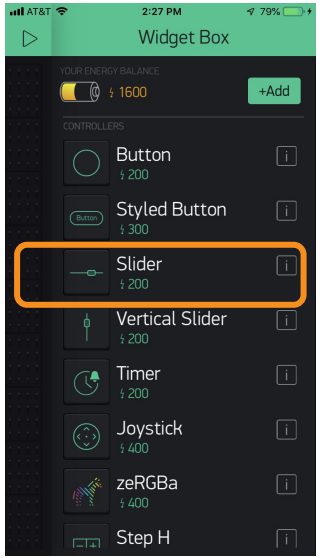
- Have control over the shape, and on/off color, and font size
- Same function as the standard button
- Note that you do not have the ability to choose D7 again until

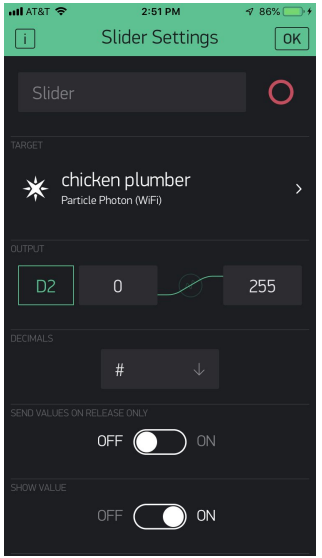


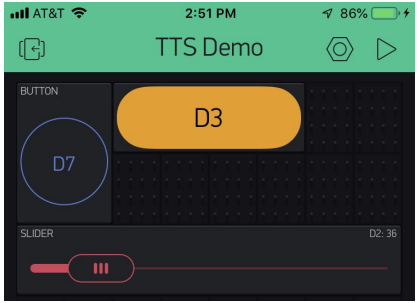


Slider

Send a variable output to an actuator



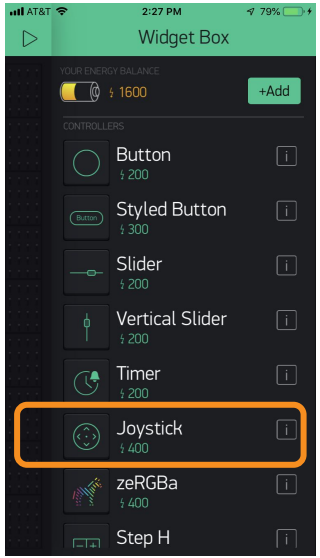


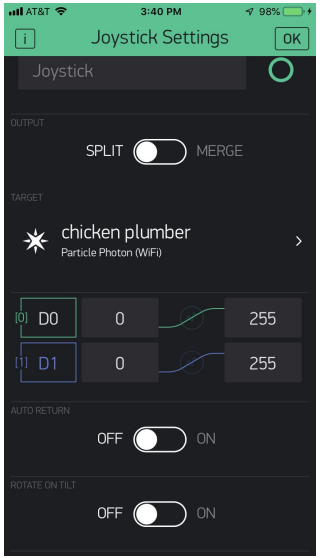


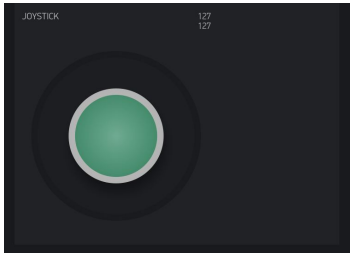
- Add slider widget to your canvas
- Change the output settings to 0-255
- Change “Send Values on Release Only” to OFF
- Set output to digital pin 3
- Add LED to pin 3 on your breadboard
- Watch the LED change brightness

Joystick

Bi-directional control





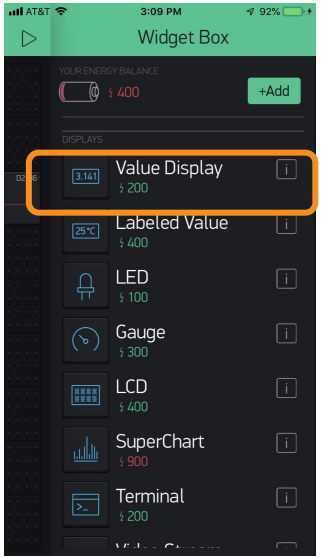


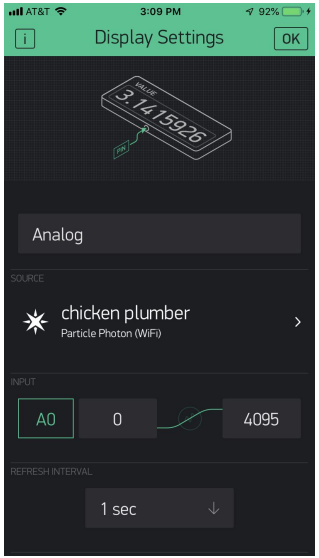
- Add Joystick widget to your canvas
- Change the output settings to D0 and D1 and both from 0-255
- Set auto-return to off
- Install LEDs on pins D0 and D1 on the Photon
- Run the app and move the joy stick widget around to see how the channels mix

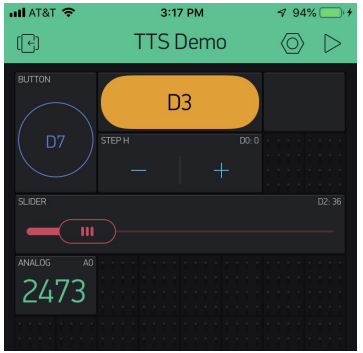
Value Display

Read sensor data

- Add a Value Display widget to your canvas
- Choose pin A0 and set range from 0-4095
- Hook up rotary pot to pin A0 on the Photon
- Move the dial and watch the values change

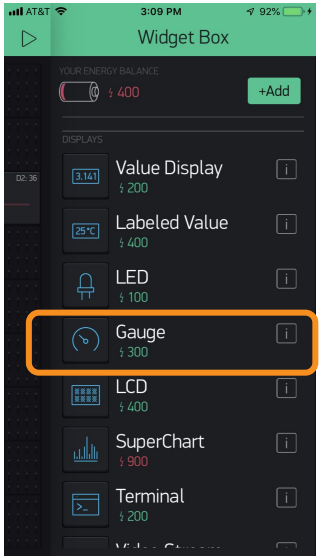


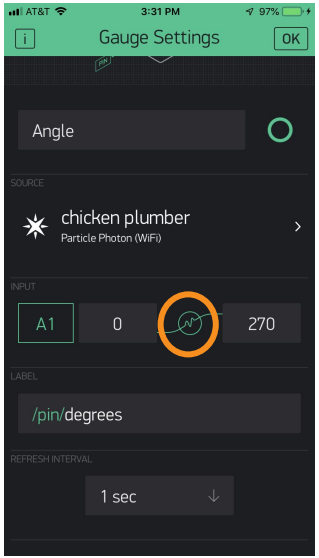


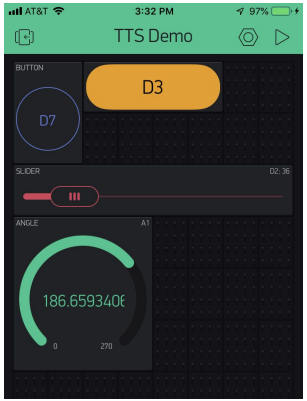


Gauge

Graphical sensor readout







- Add a Gauge Display widget to your canvas
- Choose pin A4 and set range from 0-270
- Click the wavy circle between the min and max values (this will scale the output and give us a calibrated output
- Hook up rotary pot to pin A4 on the Photon breakout board
- Move the dial and watch the values change

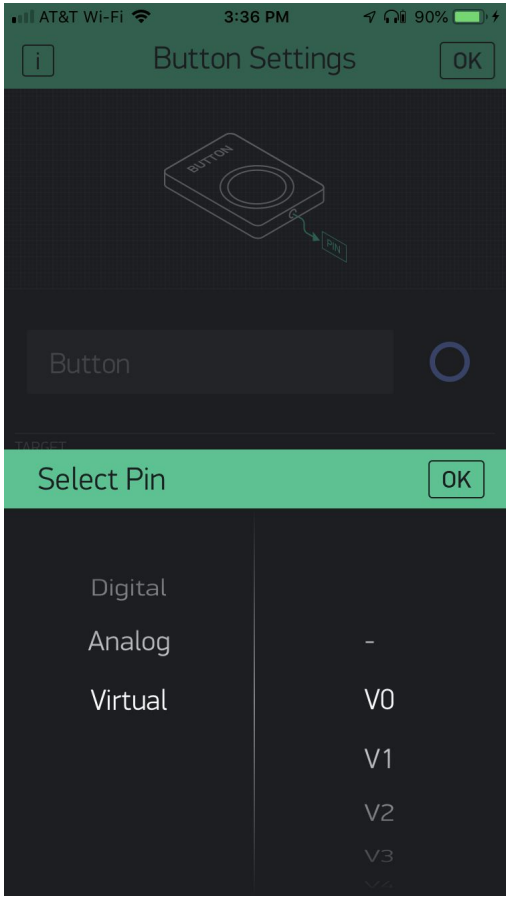
ENVENTYS PARTNERS

Blynk Virtual Pins

Virtual Pins

Add some jazz to your control

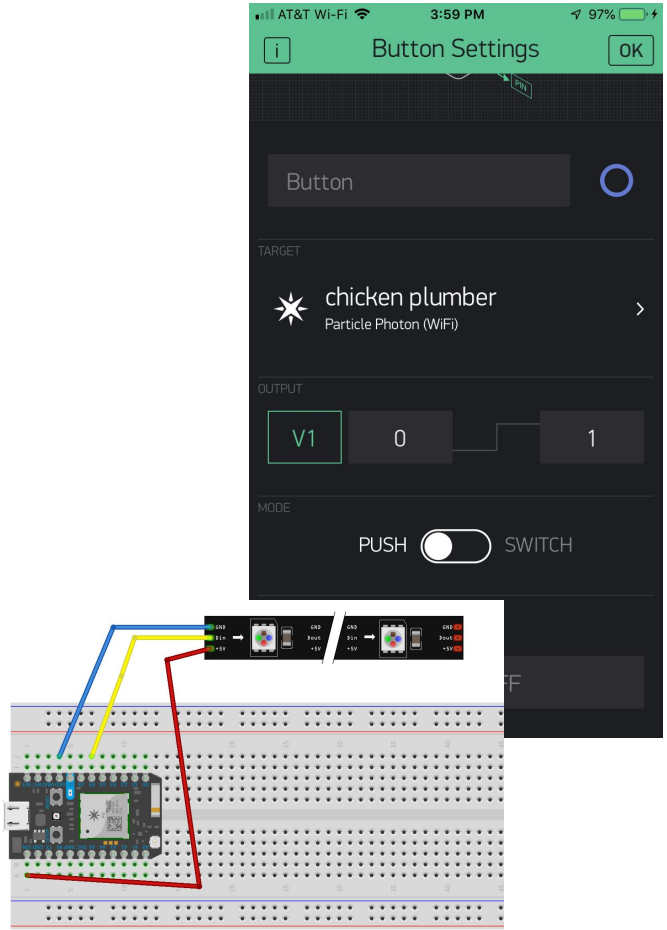
- Virtual pins allow Blynk to trigger advanced functions that can run on the Photon based on input from the app
- Called by simple code BLYNK_WRITE(V1) {} where V1 is the virtual pin
- Can have many many virtual pins in one app



Button to Drive Addressable LED

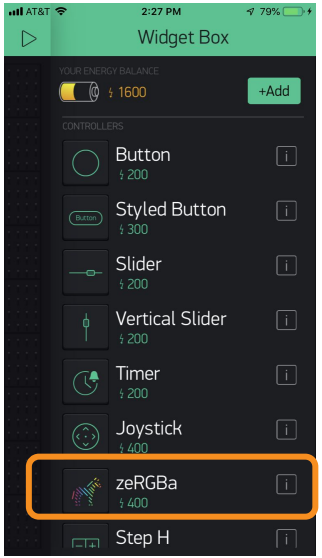
One touch fireworks

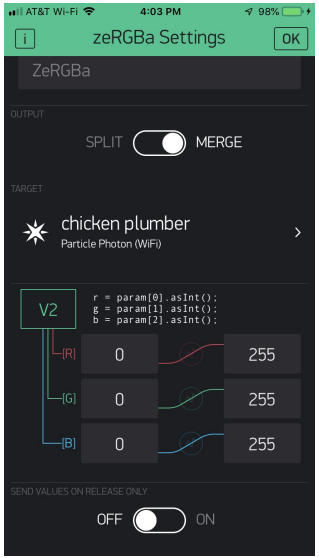
- Change the button on D7 to V1 and change the mode to “Push”
- Upload **sdq_blynk_write.ino** from the Github
- Connect LED ring to pin D6 and give it 5V power
- Start the Blynk app and push the button

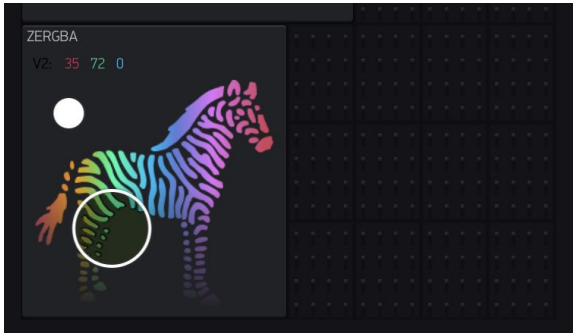


RGB Picker

Add color on the fly





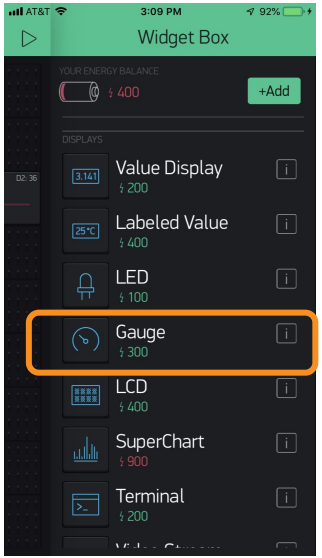


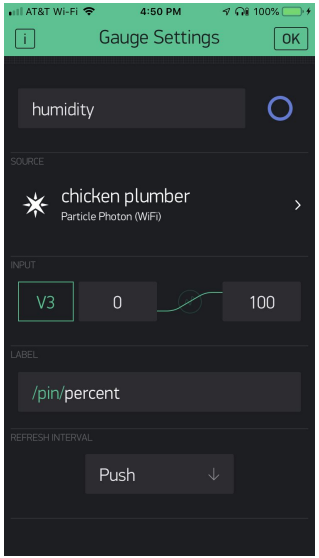
- Add a zeRGBa widget to your canvas
- Select V2 for the virtual pin and select “Merge” for the output option
- Turn “Send values on release only” to OFF
- Deploy widget and play app
- Swipe fingers across the zebra and notice the RGB LED on the Photon change color

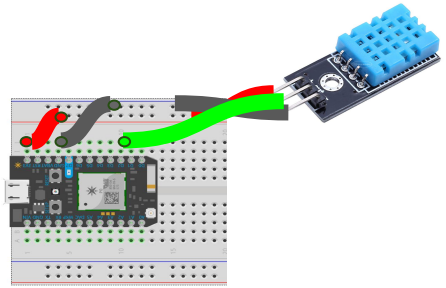
Data Recording

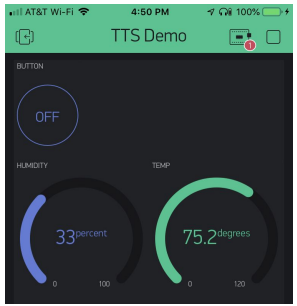
Get DHT data to show in your app

- Push **sdq_blynk_dht11.ino** to your Photon. This pushes DHT data to virtual pins 3 and 4
- Add 2 new gauge widgets to your app. Select pin V3 for the humidity and V4 for the other for temperature



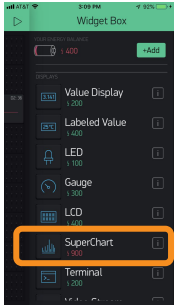


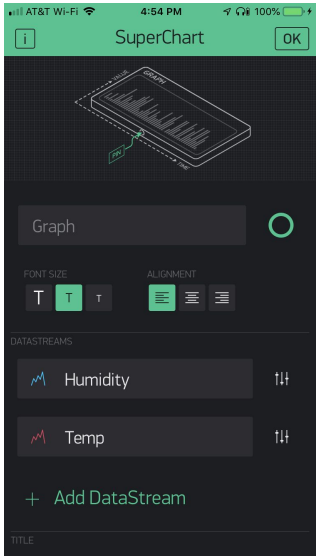


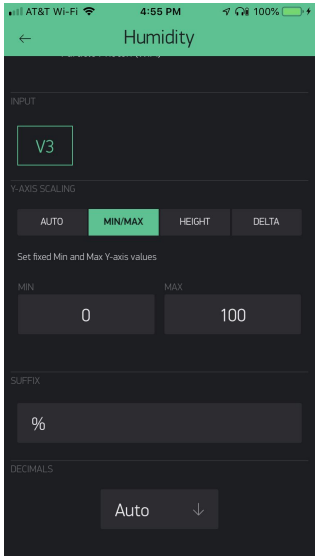


SuperChart

How good is this?







- Add SuperChart to your canvas (you may have to delete a couple of widgets to have enough energy (but it's worth it!)
- Edit the data stream to pull data from virtual pin V3
- Add a data stream and add temperature data from pin V4

SuperChart

Play it and watch the magic

- Watch the values roll in
- Can rotate phone to landscape mode to get a better view
- Export your data in csv format

Thank you.

JEREMY LOSAW
jeremy.losaw@enventyspartners.com
www.enventypartners.com
www.jeremylosaw.com

ENVENTYS PARTNERS