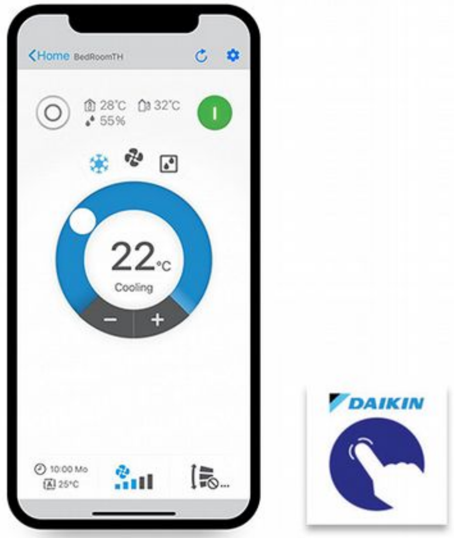
Internet-of-Things for Fun and Profit



A POSTERIORI
Play · Experience · Learn

Before we start...

- We believe in open access to knowledge
- All our slides are shared online for free
- You can print it, share it, modify it, use it to run your own courses
- This current set of slides can be found here

About Us

<u>YONI</u>
Spent 15 years developing software for big banks, now developing the next generation of Makers and Coders.

<u>CORT</u>
Ex-Navy engineer managing big engines, powerful generators, and easily choked toilets. Codes and builds stuff because he's too cheap to buy

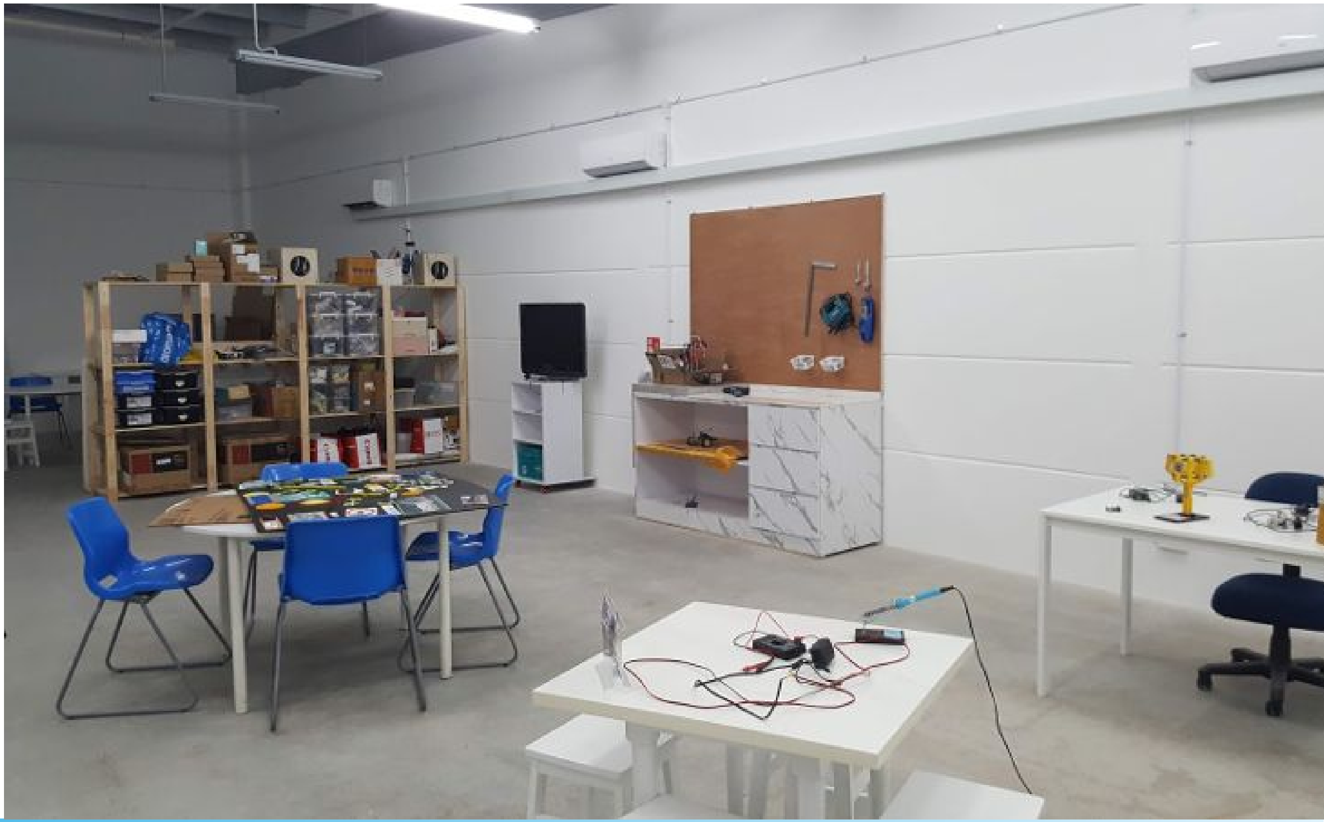
Our Company

A POSTERIORI

Play · Experience · Learn

Our Makerspace

- Suitable for electronics, 3D printing, woodworks, coding and tinkering
- Free for public use
- Located in Mega Woodlands



Our Programmes

- Competition based
  - Eg. FLL Junior (...our team won 1st in 2019!)
- Holiday programmes
  - Robotics, Apps creation, Coding, 3D Designs, Science and Research
- Regular sessions
  - Interest based. Everything from game creation to building a human sized robot.

What is the Internet-Of-Things (IoT)?

A POSTERIORI
Play · Experience · Learn

What is IoT?

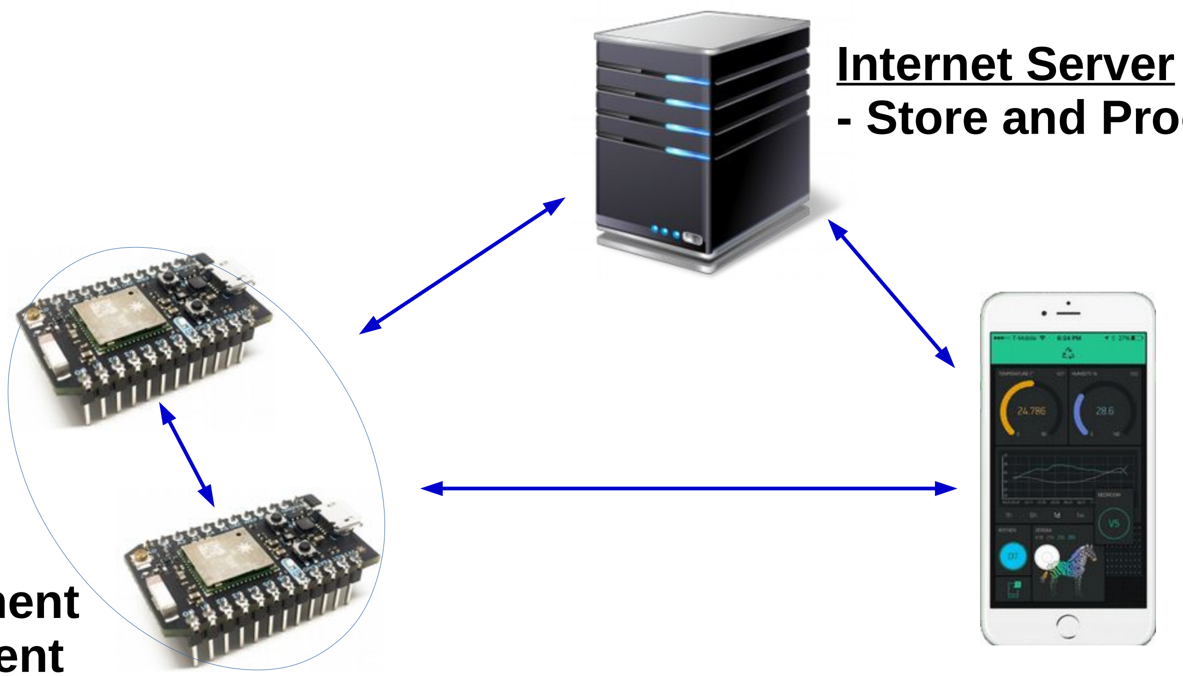
Control lights through phone...

Turn on aircon before reaching home...

Toast your bread remotely (...we don't know why either)

- Make existing or new devices more useful by connecting them to the internet (eg. lights, aircon, door locks, burglar alarm)

How Does IoT Work?



Internet Server
- Store and Process

User Devices
- View and Control
- Typically phones or computers

IoT Devices
- Sense environment
- Control equipment

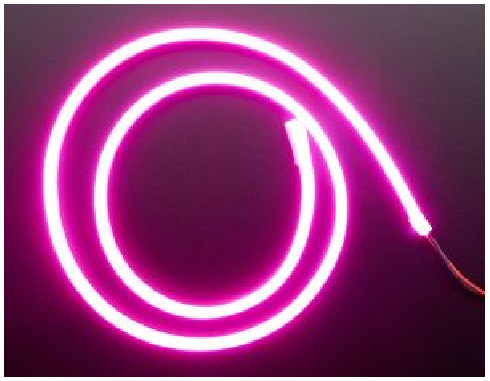
IoT in the Industry

Notify vendor to top-up machine when empty

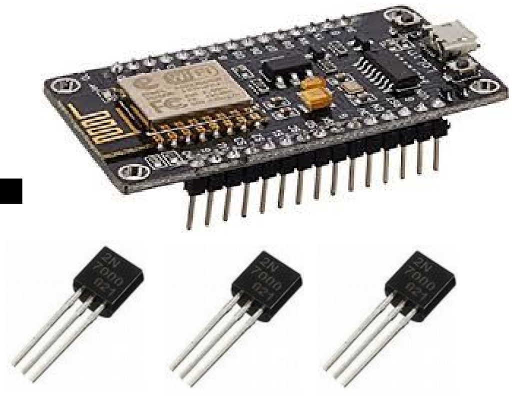
Track water and power usage

Arrange for garbage collection when bin is full

IoT is Easy!



LED Strips
($15 with power supply)



Controller and transistors ($10)

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "xxx";

char ssid[] = "HomeWifi";
char pass[] = "yyy";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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

13 lines of code
(5 mins of your time)

Phone controllable
mood lighting

A POSTERIORI
Play · Experience · Learn

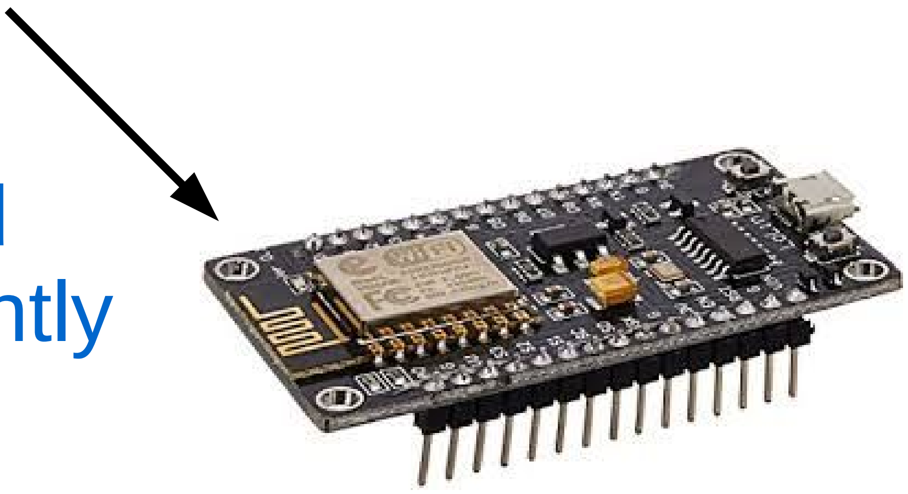
Slides available at:
http://aposteriori.com.sg/other_resources

The ESP8266

A POSTERIORI
Play · Experience · Learn

ESP8266

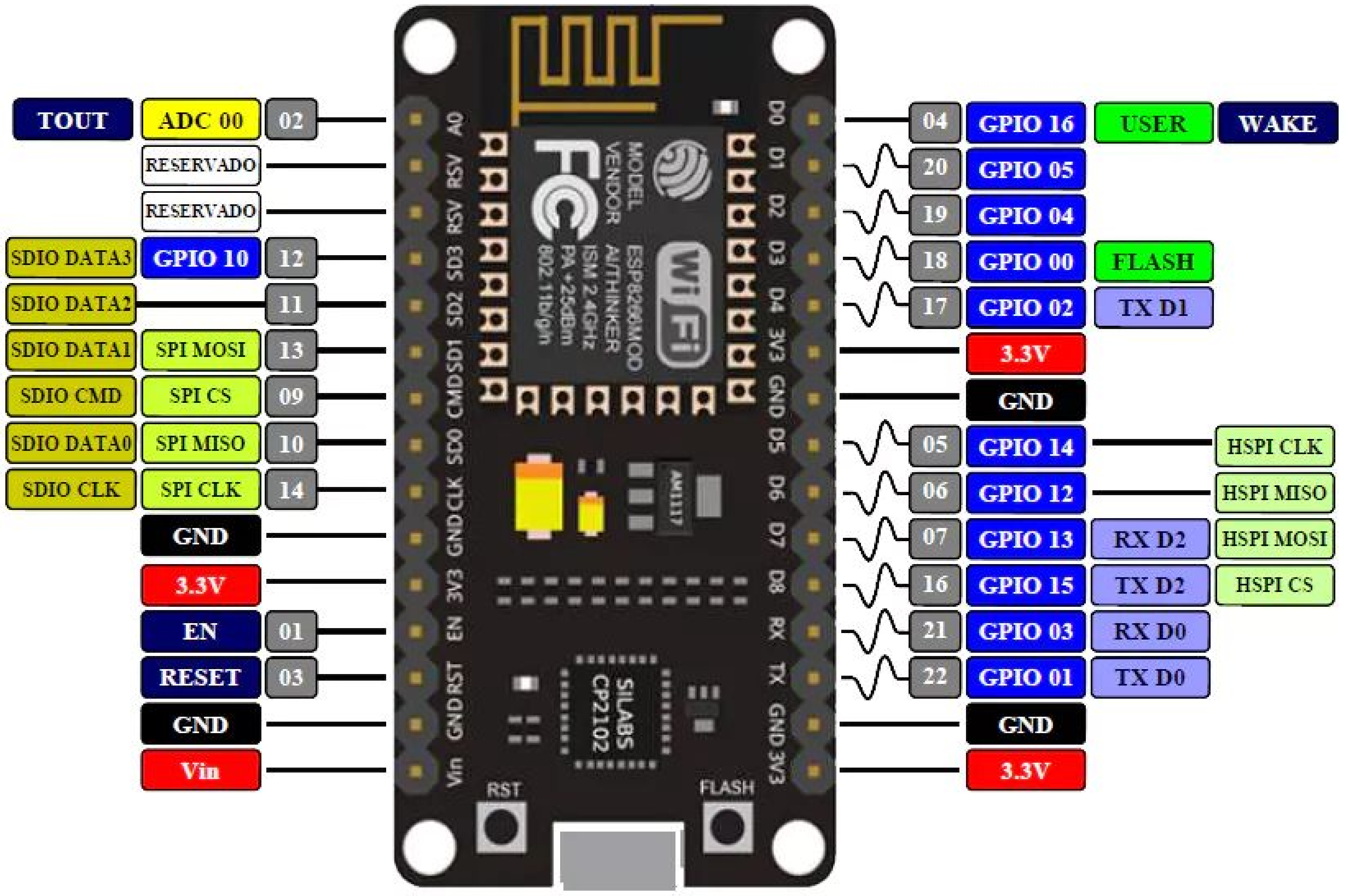
- Many different variations
- We recommend the ones that are described as “Node MCU” (...this one)
- Others are fine too, but specs, programming and connections may be slightly different



What is the ESP8266?

- Microcontroller (similar to Arduino and micro:bit)
- Built-in WiFi (b/g/n)
- 80MHz Processor
- 32KiB instruction RAM and 80KiB data RAM
- Runs on 3.3V (...built-in voltage regulator allows you to use up to 5V)

Pinout


TOUT
ADC 00
02
RESERVADO
RESERVADO
SDIO DATA3
GPIO 10
12
SDIO DATA2
11
SDIO DATA1
SPI MOSI
13
SDIO CMD
SPI CS
09
SDIO DATA0
SPI MISO
10
SDIO CLK
SPI CLK
14
GND
3.3V
EN
01
RESET
03
GND
Vin
A0 RSV RSV SD3 SD2 SD1 CMD SD0 CLK GND 3V3 EN RST GND Vin
D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
RST
FLASH
04
GPIO 16
USER
WAKE
20
GPIO 05
19
GPIO 04
18
GPIO 00
FLASH
17
GPIO 02
TX D1
3.3V
GND
05
GPIO 14
HSPI CLK
06
GPIO 12
HSPI MISO
07
GPIO 13
RX D2
HSPI MOSI
16
GPIO 15
TX D2
HSPI CS
21
GPIO 03
RX D0
22
GPIO 01
TX D0
GND
3.3V

Software Setup

Workshop participants: These have been done for you!

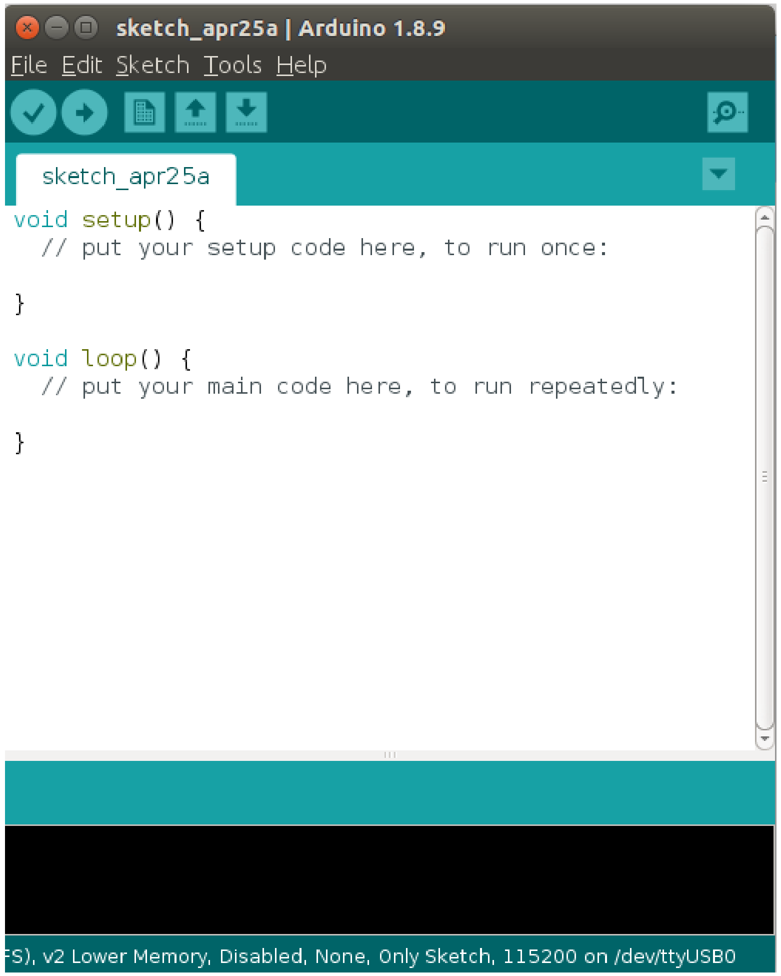
- **Install Arduino IDE:**
  https://www.arduino.cc/en/Main/Software
- **Install ESP8266 libraries:**
  (Follow instructions here: https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/installing-the-esp8266-arduino-addon)
- **Install Blynk libraries:**
  https://github.com/blynkkk/blynk-library/releases/tag/v0.6.1
- **Install CH340 drivers (Not required for Mac and Linux)**
  http://www.wch.cn/download/CH341SER_EXE.html

Exercise 1 : Basic Blynk

- Program the ESP8266 using Arduino
- Connect to the Blynk server
- Create an IoT App using Blynk

Start Arduino IDE

Arduino IDE
An Integrated Development Environment (IDE) used for programming the Arduino microcontroller as well as many other microcontrollers.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

FS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on /dev/ttyUSB0

Before we start…
Go to “Tools > Board” and select “NodeMCU 1.0 (ESP-12E Module)”

Exercise 1: Basic Blynk

This is a multi-line comment. Everything from `/*` ...to... `*/` is ignored.

Single-line comment. Everything that starts with `//` is ignored.

```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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

Download Examples
You can download all code for the exercises from...

https://www.aposteriori.com.sg/other-resources/

No need to copy!

WiFi SSID and Password

• SSID:	Innovate_Workshed
• Password:	1nnoWorkshed

- **Make sure to change these if you're trying it out at home!**

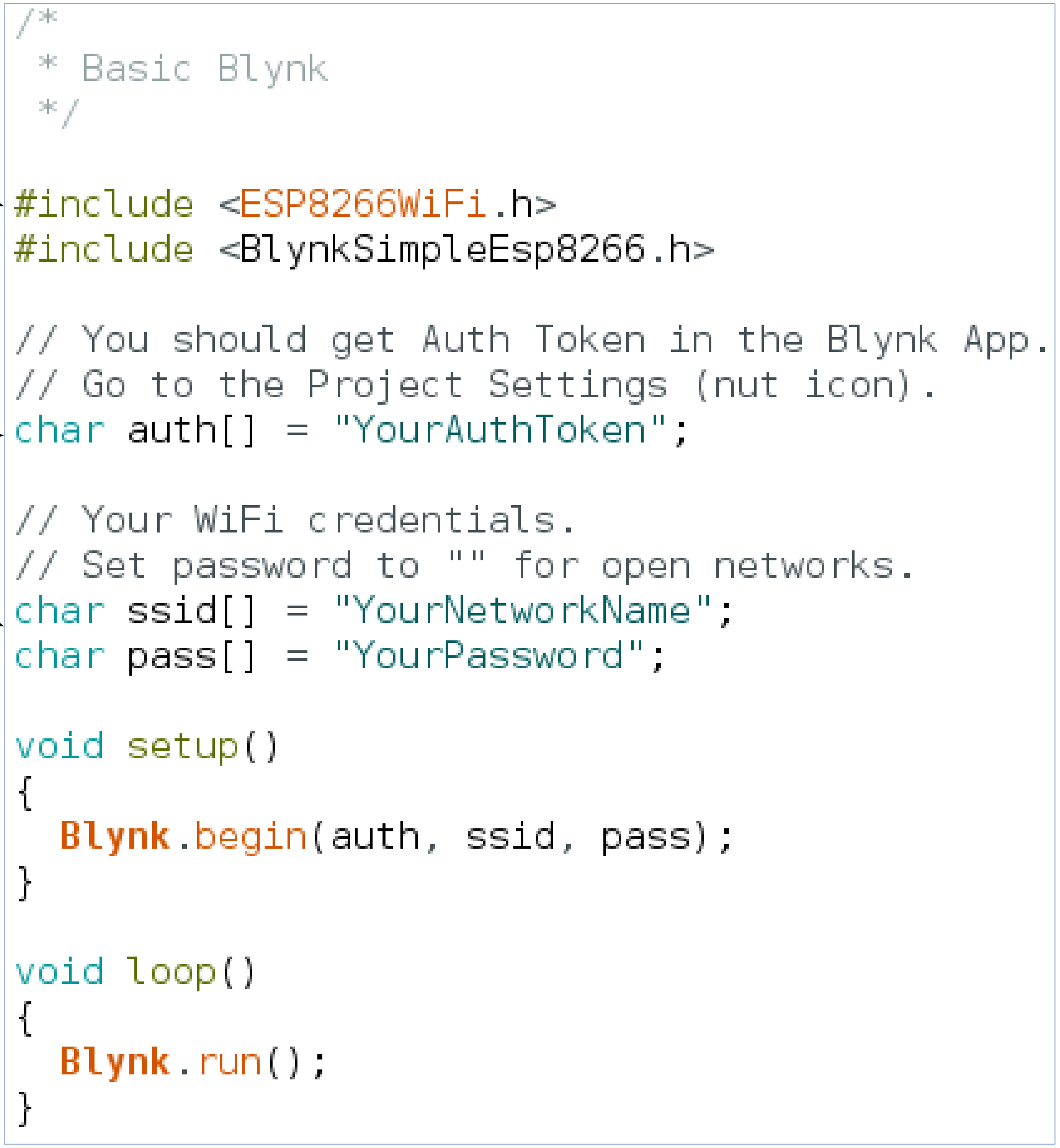
Exercise 1: Basic Blynk

#Include
Add the ESP8266 Wifi library and the Blynk library to your program.

Variables
Variables are used to store values.

Set the WiFi ssid and password correctly. We'll deal with the Blynk token later.

Tips
The "Arduino" programming language is just C/C++ with some added functions. You can refer to any C/C++ tutorials to learn more about the programming language.

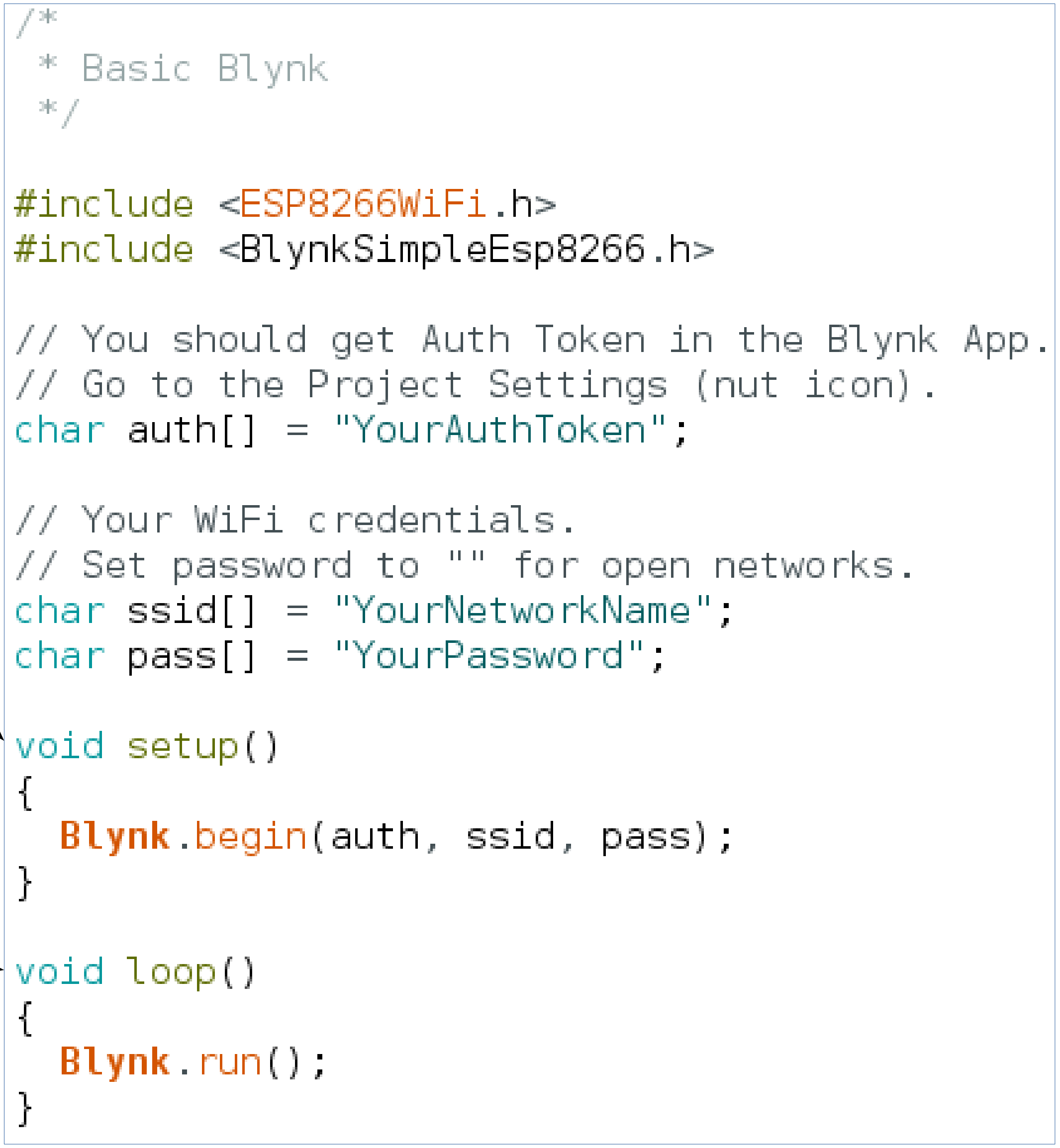


```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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

Exercise 1: Basic Blynk

setup()
The code here will run **once** when the microcontroller is powered up.

loop()
The code here will repeat forever. Every time the program reaches the end of the loop, it'll start again from the top.



```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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

Exercise 1: Basic Blynk

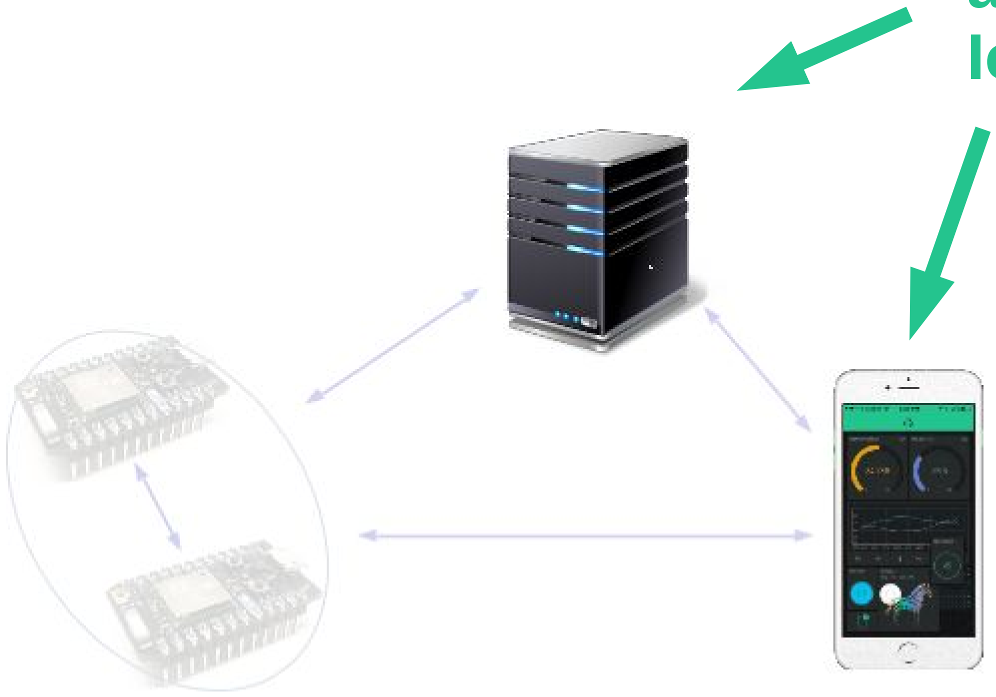
Blynk.begin
Connect your microcontroller to the WiFi network, then connect to the Blynk server.

Blynk.run
Wait for commands from the Blynk server.

```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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

Slides available at:
http://aposteriori.com.sg/other_resources

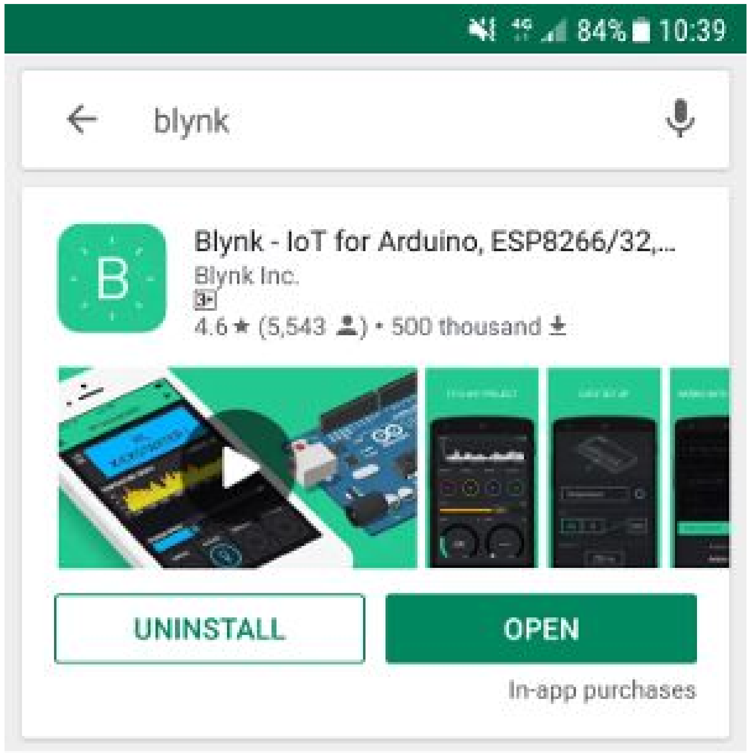
Blynk provides the server and the mobile app for your IoT device.



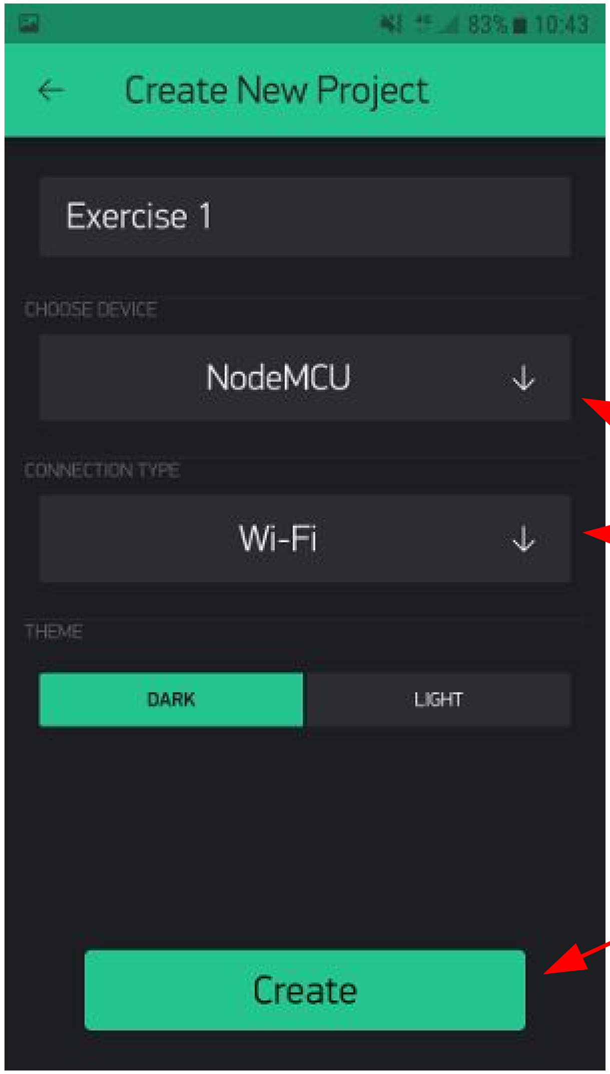
A POSTERIORI
Play · Experience · Learn

Blynk

- Install Blynk
- Available for iOS or Android
- Register and Login


84% 10:39
blynk
Blynk - IoT for Arduino, ESP8266/32,...
Blynk Inc.
3+
4.6 ★ (5,543) • 500 thousand
UNINSTALL
OPEN
In-app purchases

Blynk

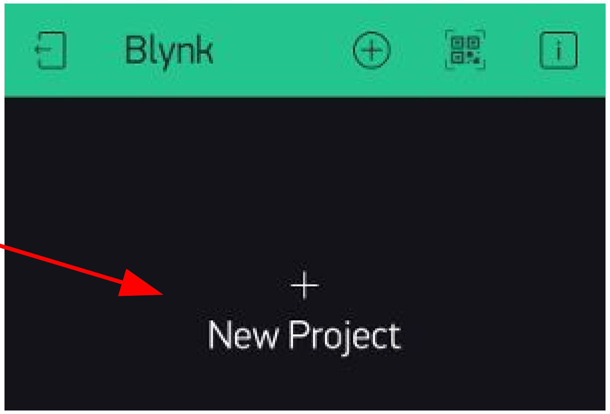


Exercise 1

CHOOSE DEVICE: NodeMCU

CONNECTION TYPE: Wi-Fi

THEME: [✓ DARK] LIGHT



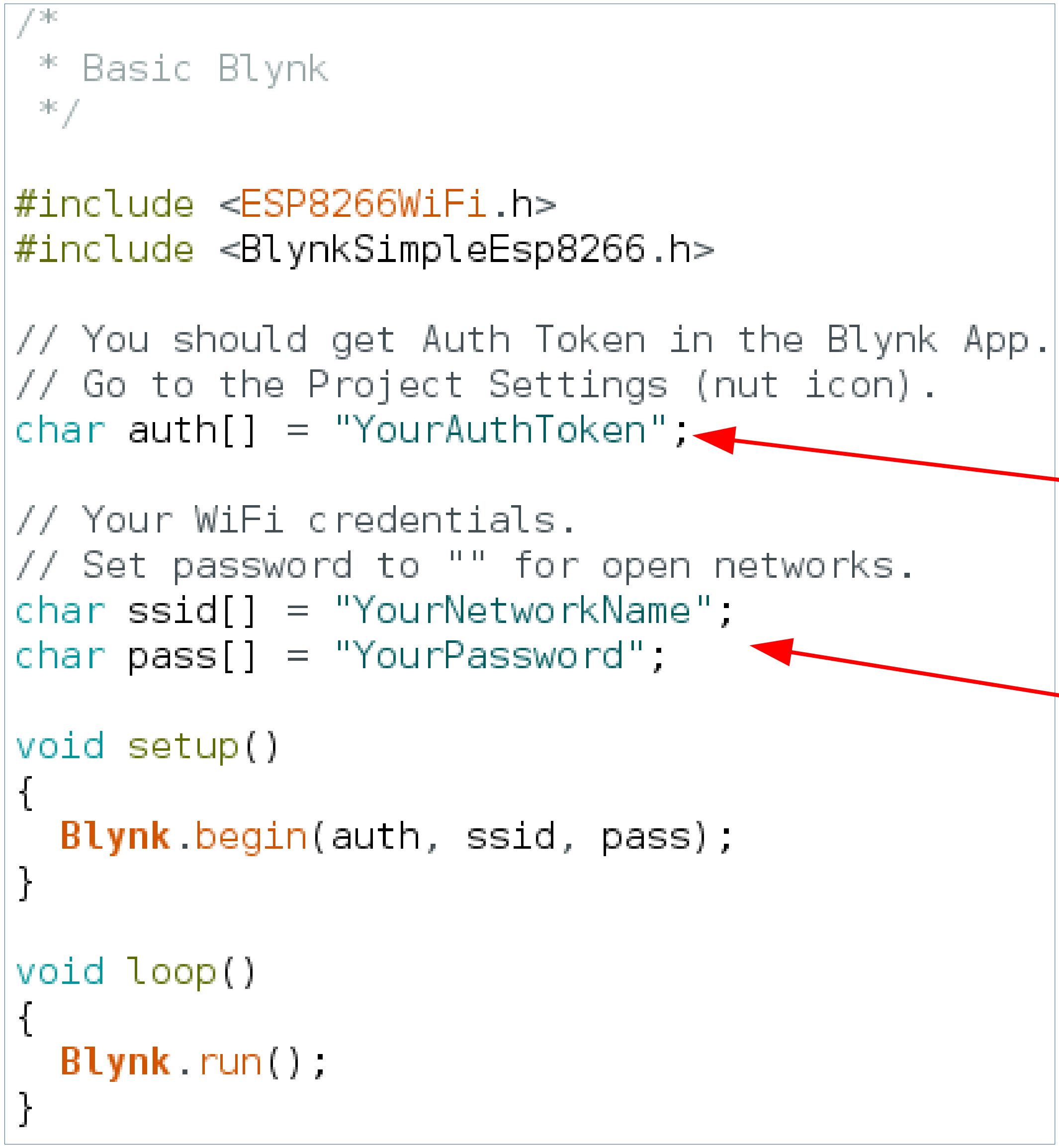
Create a new project

Set “Device” to “NodeMCU” and “Connection Type” to “Wi-fi”

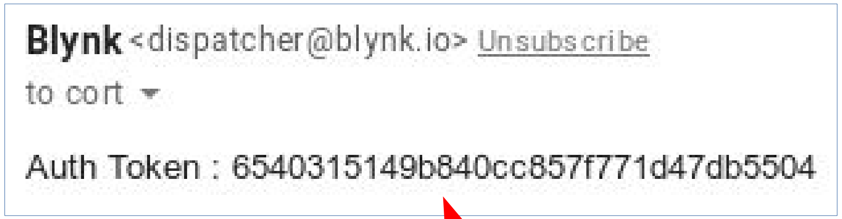
For the project name and theme, choose whatever you like.

Tap “Create” when you’re done

Blynk



```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Blynk.begin(auth, ssid, pass);
}

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



Blynk <dispatcher@blynk.io> Unsubscribe

to cort

Auth Token : 6540315149b840cc857f771d47db5504

Check your email for the Auth Token…

...and put it in here.

Make sure your WiFi ssid and password are correct!

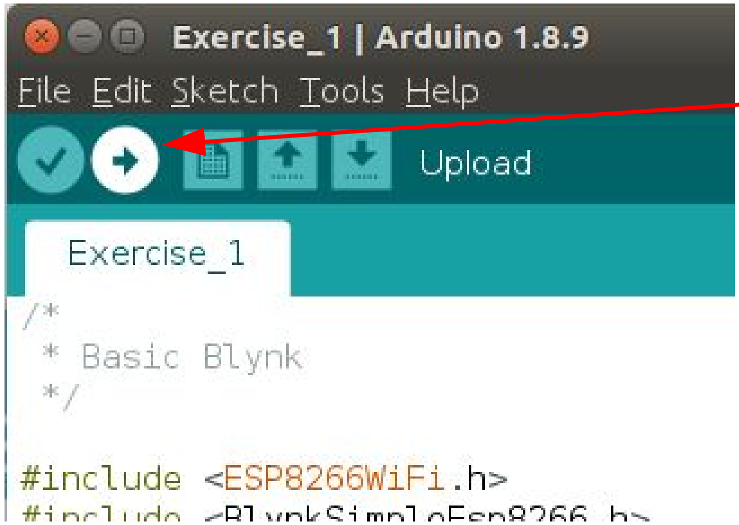
Remember to change them if you're using your device at home.

A POSTERIORI
Play · Experience · Learn

Slides available at:
http://aposteriori.com.sg/other_resources

Blynk

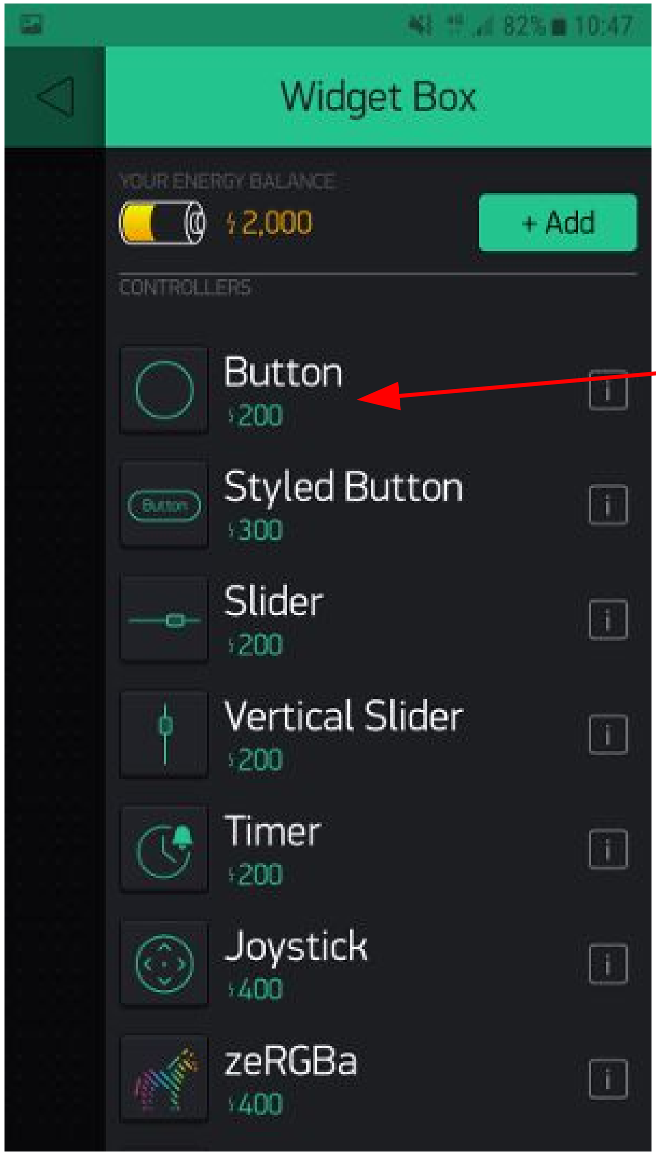
Connect
Connect your ESP8266 to your computer via a USB cable

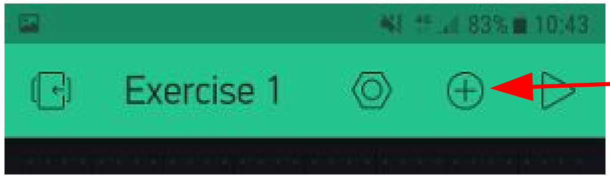


```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

Upload
Click the upload button to upload your program into your microcontroller

Blynk

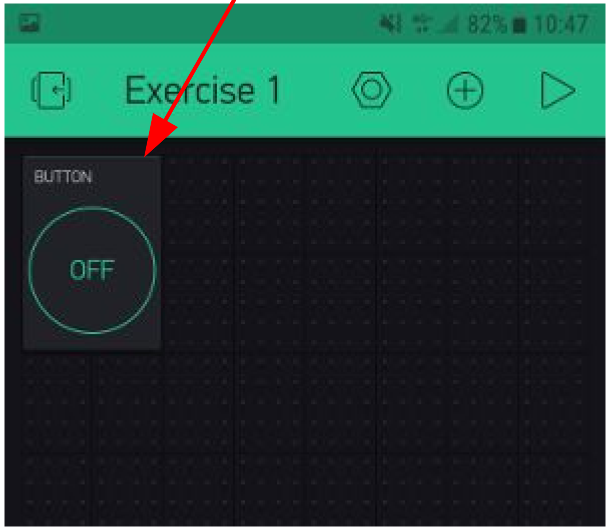




Tap the "+" to open a widget box

Tap this…

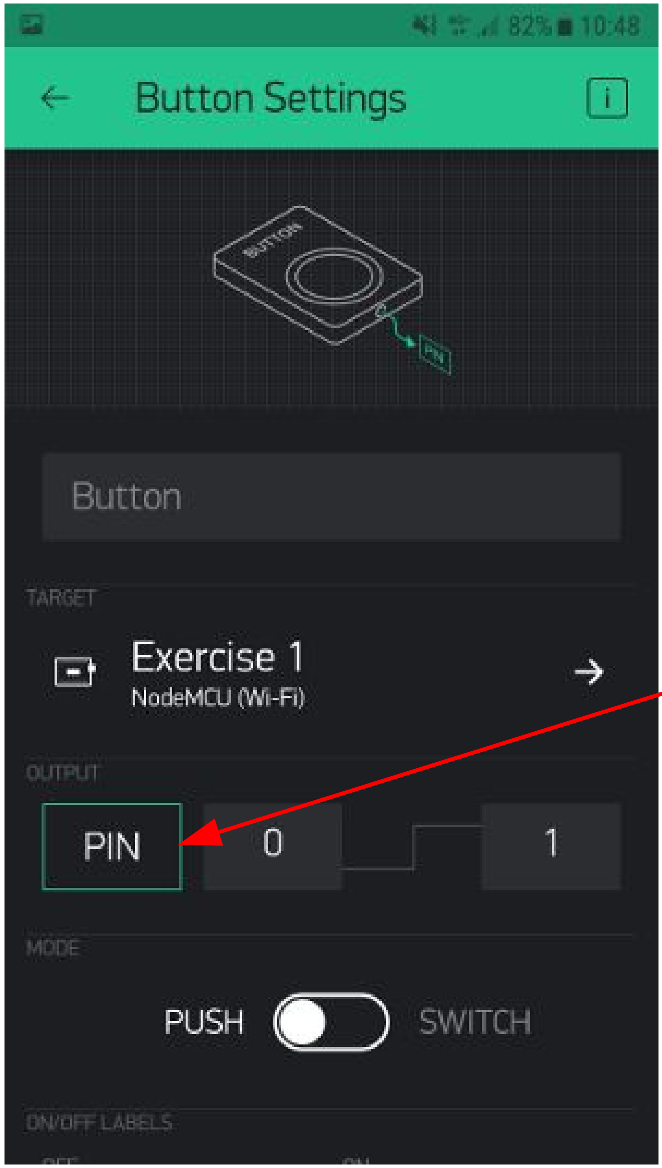
...to add a button to your screen

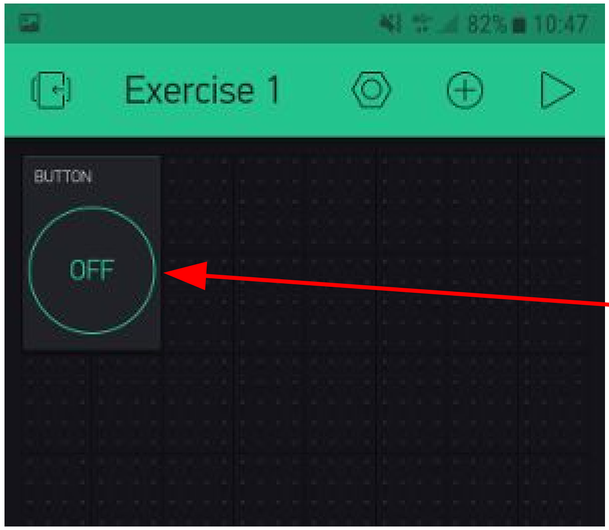


A POSTERIORI
Play · Experience · Learn

Slides available at:
http://aposteriori.com.sg/other_resources

Blynk





Tap your newly added button to configure it

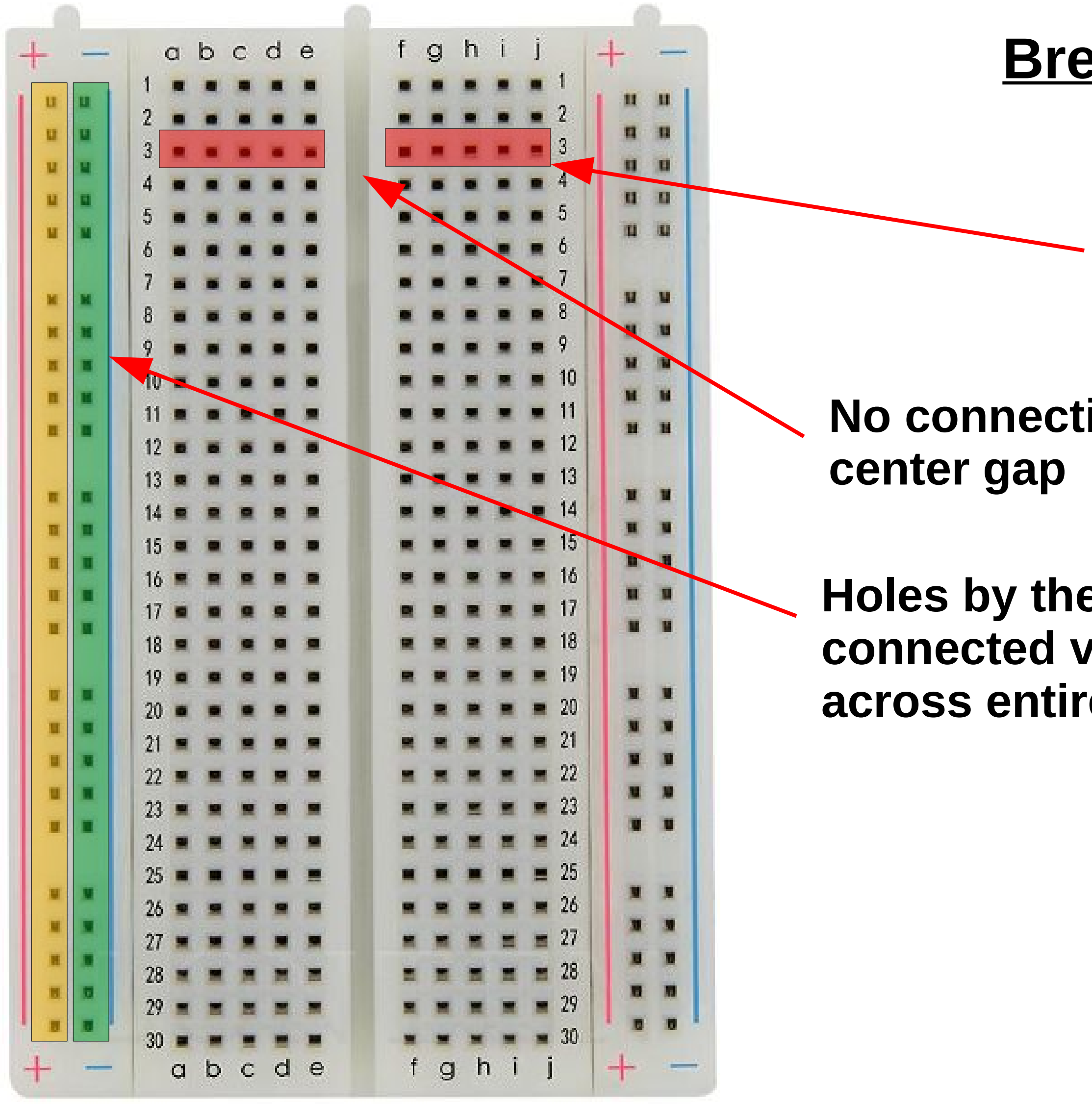
Tap on "PIN" and select "Digital D4"

D4 is connected to the built-in LED. You can set it to…

- 0 (LOW) : Turns ON LED
- 1 (HIGH) : Turns OFF LED

By default, Blynk will send a 0 when the button is released, and a 1 when the button is pressed. You can modify this by changing the 0 and 1 next to the PIN button.

Exercise 1: Basic Blynk



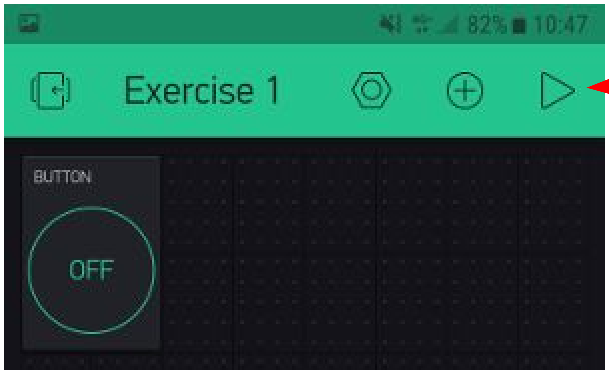
Breadboard Basics

Holes in the same row are connected

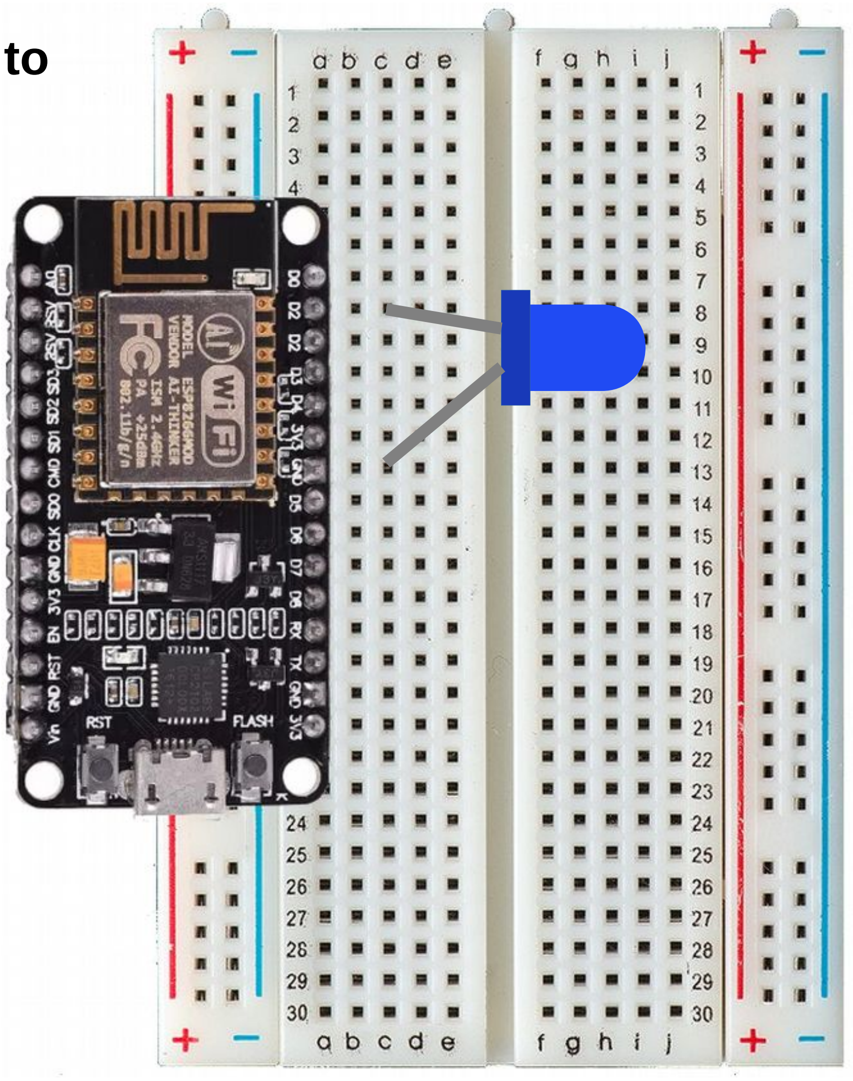
No connection across center gap

Holes by the side are connected vertically across entire board

Exercise 1: Basic Blynk

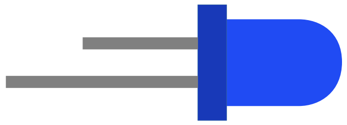


Tap the play button to start your app.



- Try using some other pins. You'll have to connect an external LED to see them work.

LED
The longer leg should be connected to positive. Shorter leg to negative / ground.



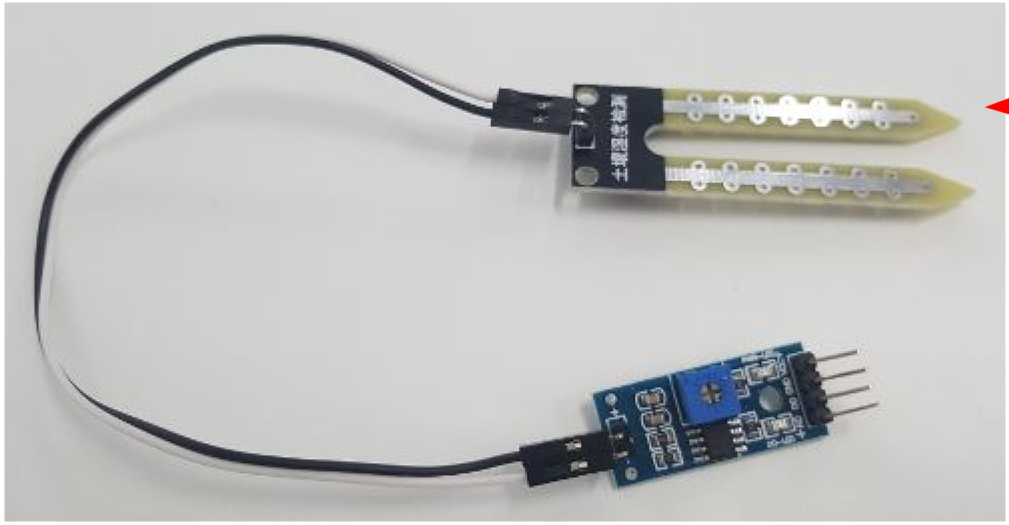
Slides available at: http://aposteriori.com.sg/other_resources

Exercise 2 : Analog Data

- Read analog data from the ESP8266
- Control power output

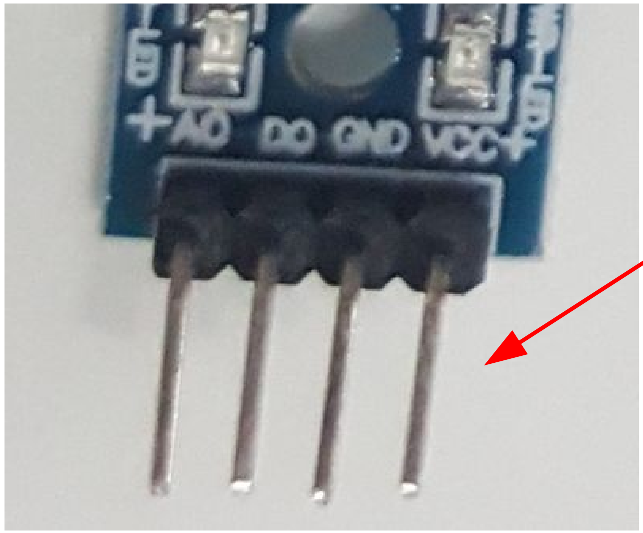
A POSTERIORI
Play · Experience · Learn

Exercise 2 : Analog Data



Connect the sensor probe to the sensor board (...orientation doesn't matter)

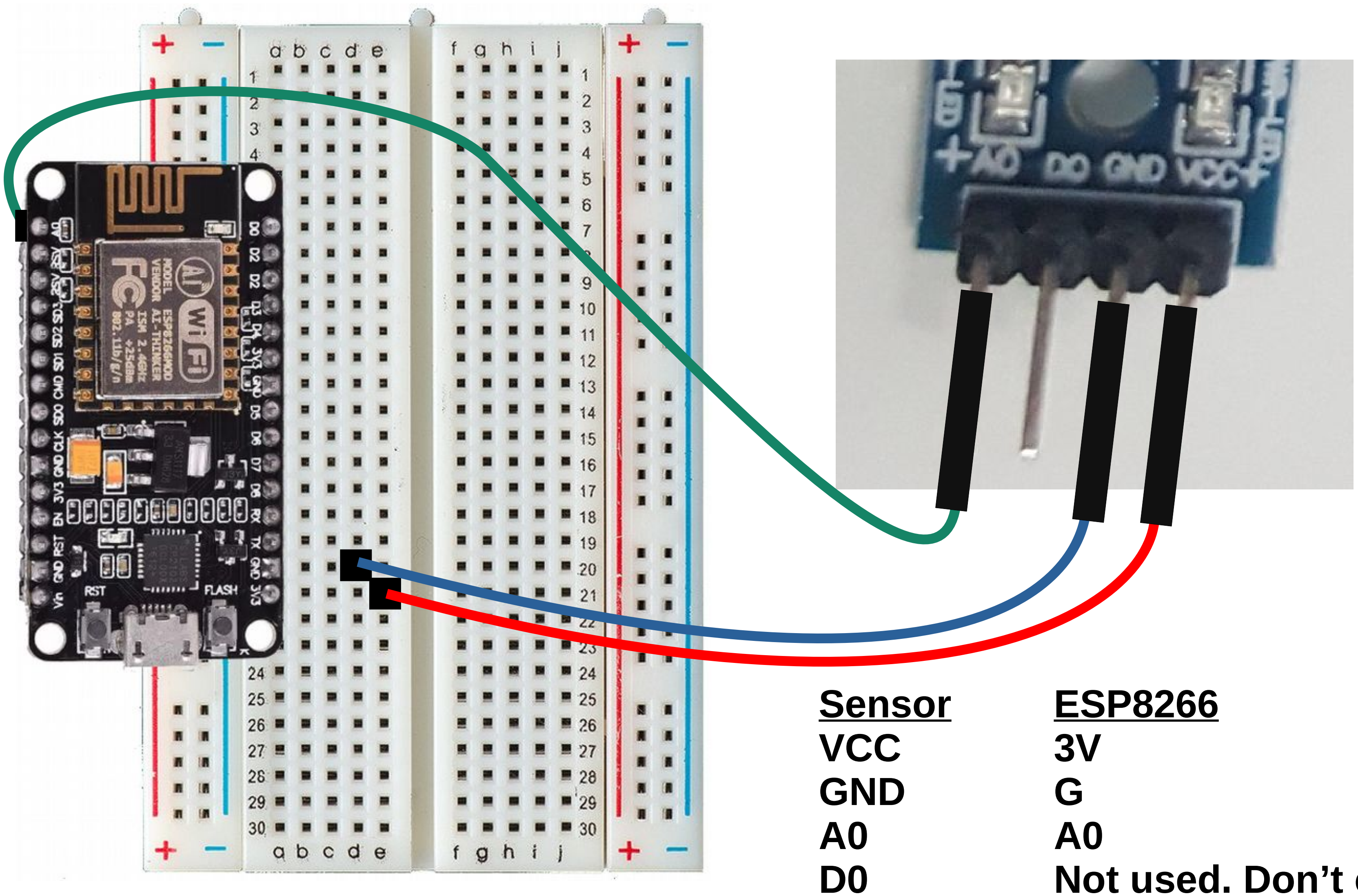
Hygrometer
This is a device for measuring water.



Connect the sensor board to the ESP8266 according to the following table:

Sensor	**ESP8266**
VCC	**3V**
GND	**G**
A0	**A0**
D0	**Not used. Don't connect**

Exercise 2: Analog Data

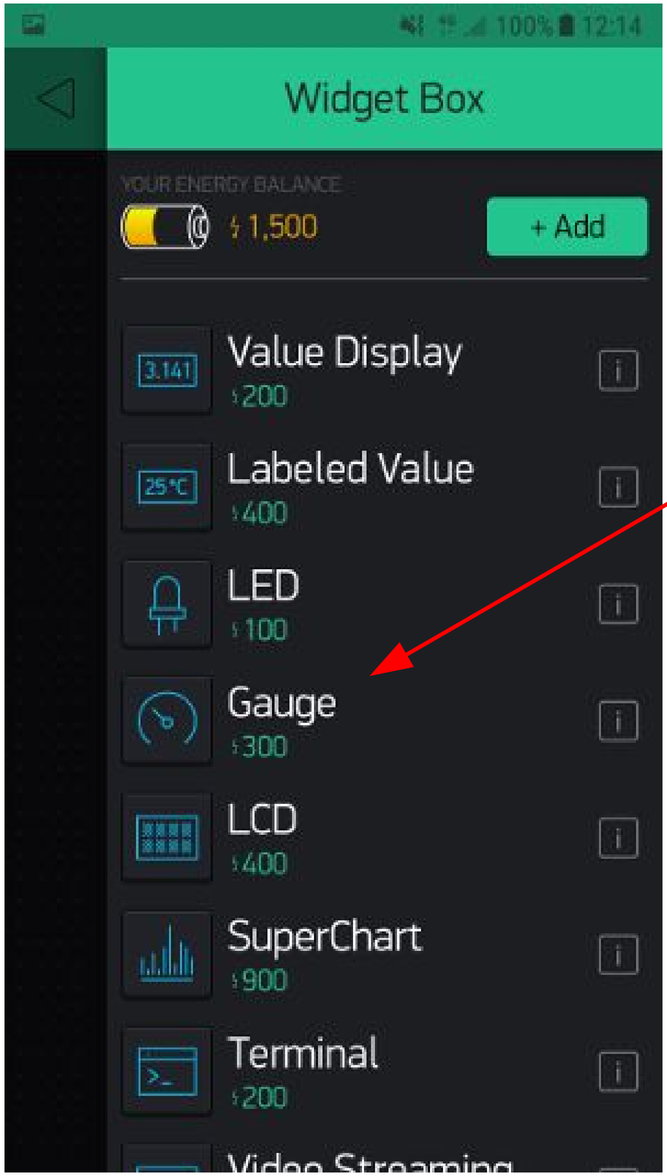


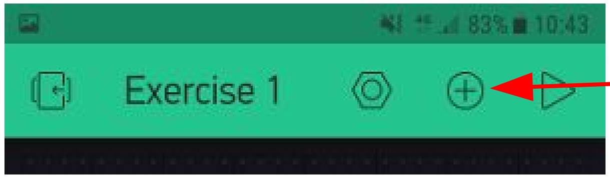
Sensor	ESP8266
VCC	3V
GND	G
A0	A0
D0	Not used. Don’t connect

A POSTERIORI
Play · Experience · Learn

Slides available at:
http://aposteriori.com.sg/other_resources

Exercise 2 : Analog Data

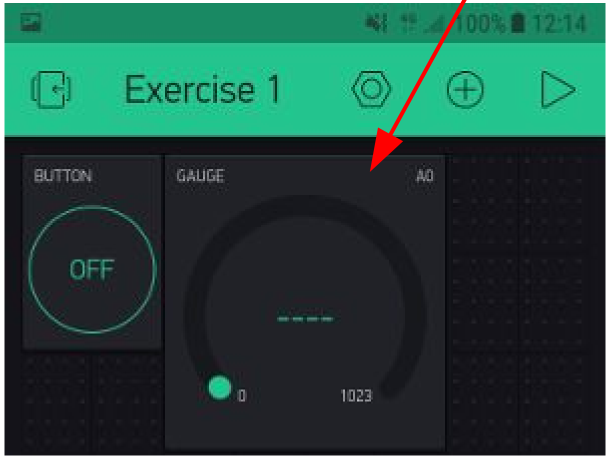




Tap the "+" to open a widget box

Tap this…

...to add a Gauge to your screen

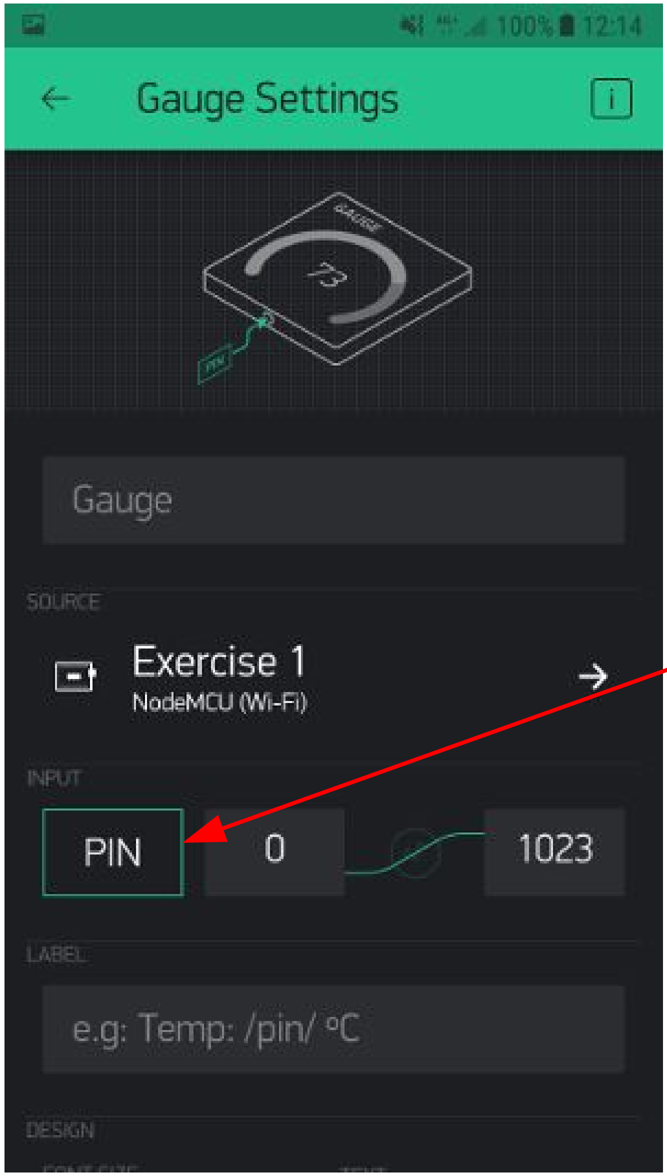


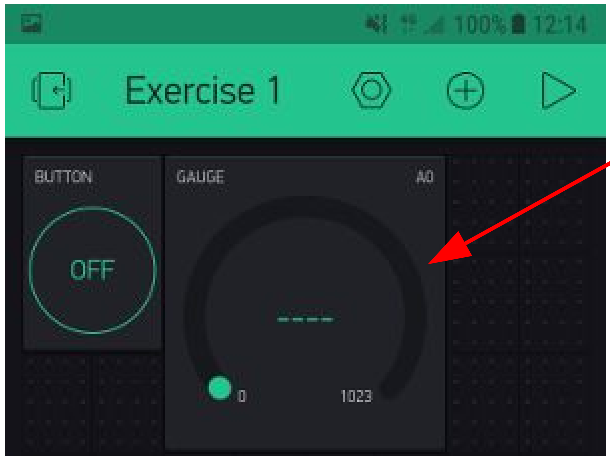
You can also try…

- Level H
- Level V
- Value Display
- Labeled Value

...they provide different ways of displaying the analog value.

Exercise 2 : Analog Data





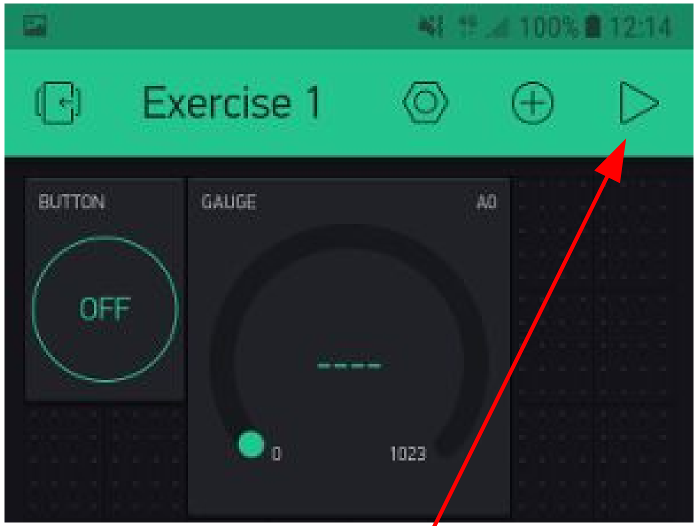
Tap the gauge to open its settings page

Tap on “PIN” and select “Analog A0”

By default, the Blynk app will **PULL** data from the microcontroller once every 1s. You can change the frequency at the bottom of this settings page.

The microcontroller can also **PUSH** data to the Blynk app.

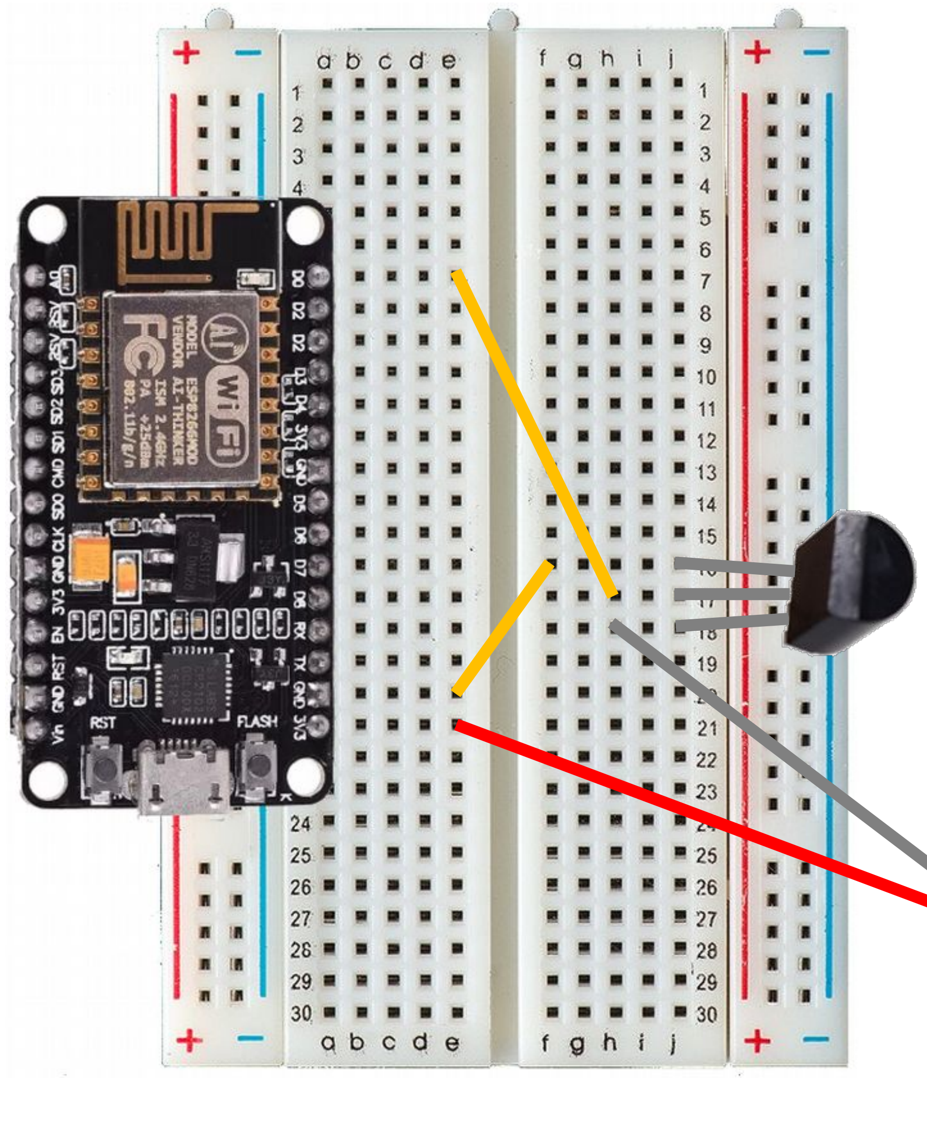
Exercise 2 : Analog Data



Tap the play button to start your app.

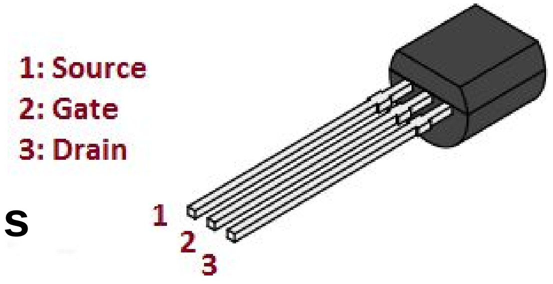
- Try dipping the hygrometer probe into water and watch the gauge value change
- You can use the hygrometer to…
  - Measure water level in your pet's water bowl
  - Measure soil moisture for your plants
  - Measure rain intensity

Exercise 2 : Analog Data



Connecting the pump

The pump motor requires more current than what the ESP8266 output pins can provide, so we'll use a transistor as a switch to turn the motor on and off.



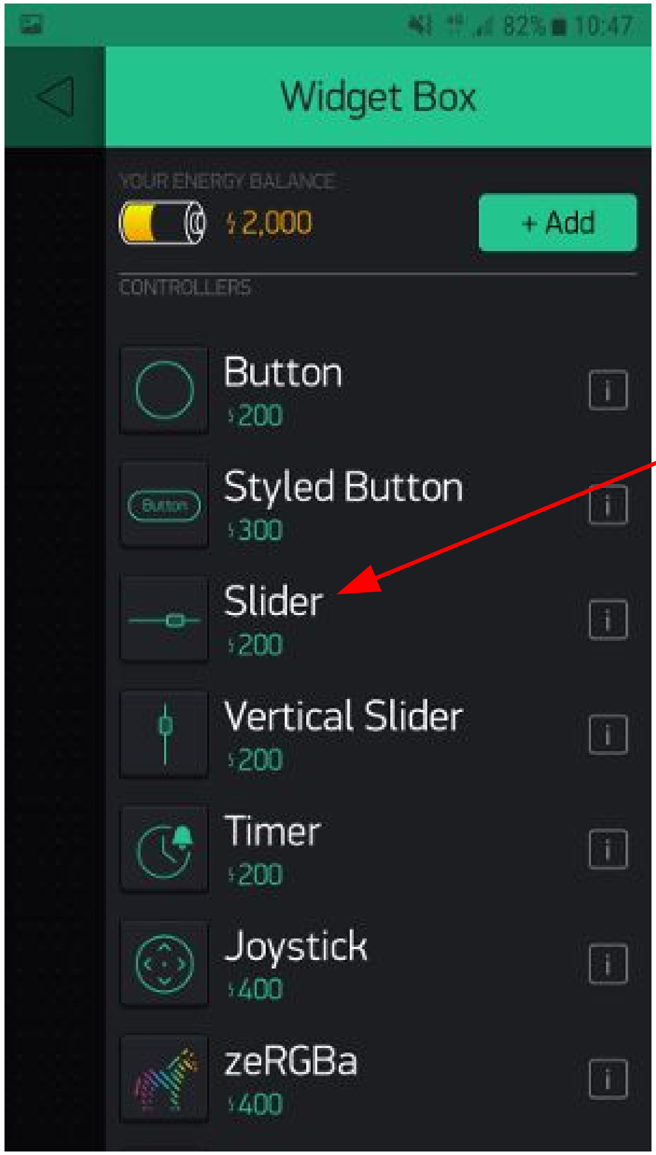
Transistor connections

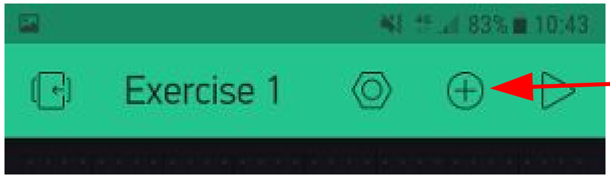
Source	→	G
Gate	→	D0
Drain	→	Motor (White wire)

Pump connections

Red Wire	→	3V
White Wire	→	Transistor (Drain)

Exercise 2 : Analog Data

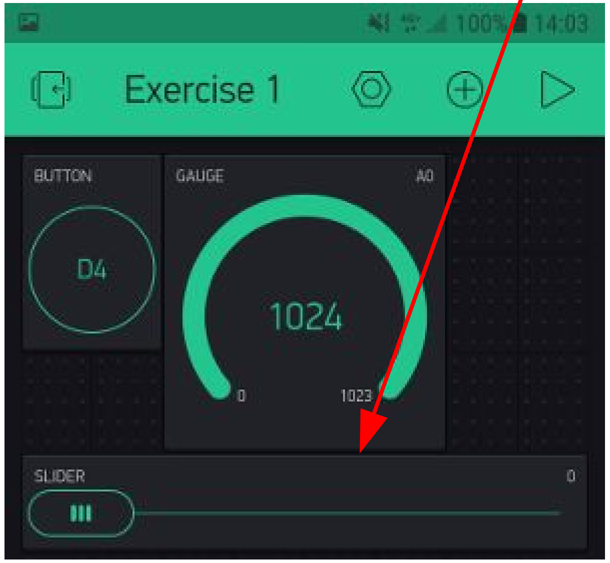




Tap the “+” to open a widget box

Tap this…

...to add a slider to your screen



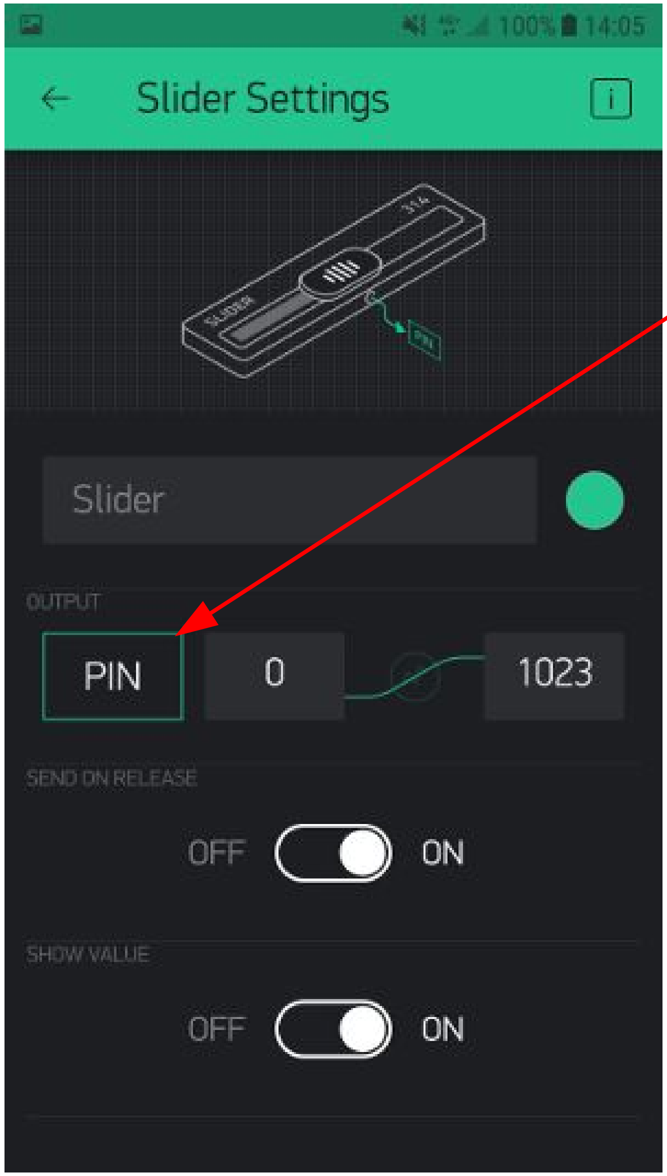
Your slider is probably smaller than what you see here.

Press and hold the slider to resize it.

A POSTERIORI
Play · Experience · Learn

Slides available at: http://aposteriori.com.sg/other_resources

Exercise 2 : Analog Data


Slider Settings

Slider

OUTPUT

PIN	0	1023

SEND ON RELEASE: OFF ON

SHOW VALUE: OFF ON

Open the slider's setting page, then tap on "PIN" and select "Digital D0"

You can also try using D4 to control the LED brightness.

Run your app and try adjusting the motor speed / LED brightness.

Note:
If your power is set too low, the motor may not turn at all.

Exercise 3 : Timer and Push

- Setting timers
- Pushing data from microcontroller to Blynk

A POSTERIORI
Play · Experience · Learn

Exercise 3 : Timer and Push

Why?

- Timer
  - Allow microcontroller to do things on it’s own without user control (eg. water plants once a day)
- Push
  - Pulling data only works when app is running
  - Having the microcontroller push data to the server allows us to record readings even when the app is off

Exercise 3 : Timer and Push

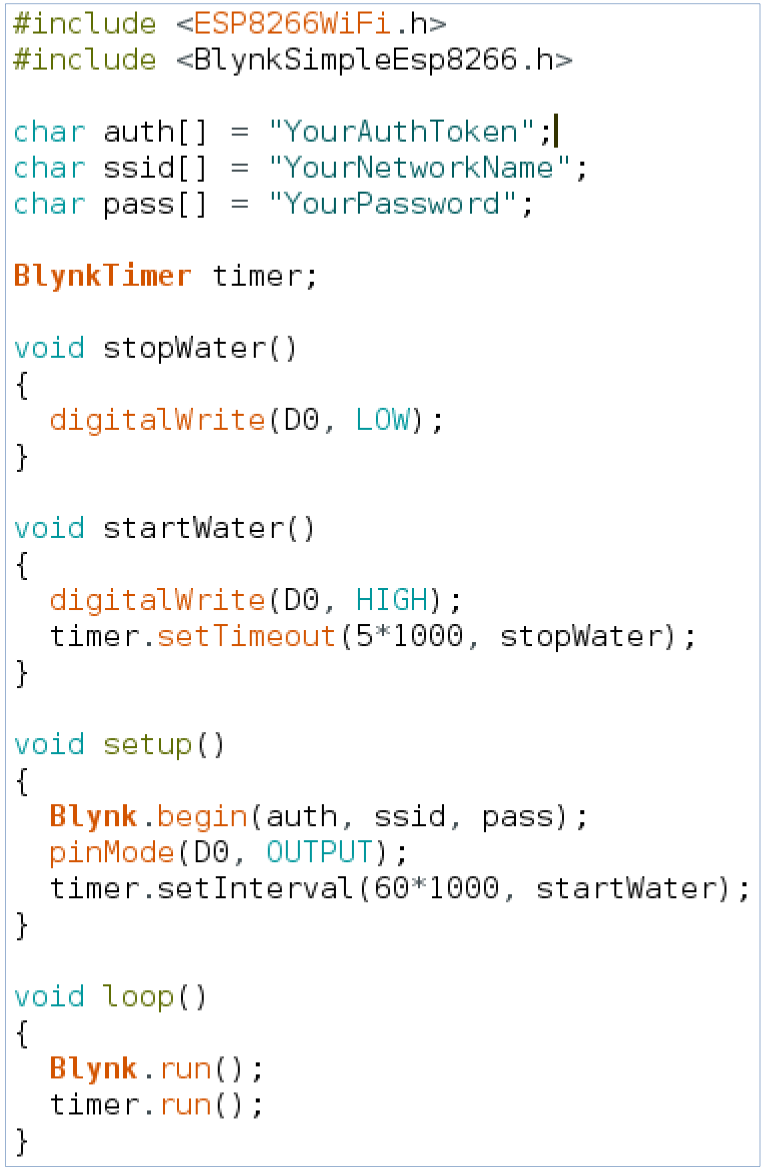
These are the same as before with comments removed to save screen space. **(Remember to set your own auth token, ssid, and password)**

This creates a timer object. We'll be using it later to set our timer.

"stopWater" and "startWater" are functions. They **will not run** until we call them. We'll look at them in more details later.

Download "Exercise 3 part 1" from...

https://www.aposteriori.com.sg/other-resources/



```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "YourAuthToken";
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

BlynkTimer timer;

void stopWater()
{
  digitalWrite(D0, LOW);
}

void startWater()
{
  digitalWrite(D0, HIGH);
  timer.setTimeout(5*1000, stopWater);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
}

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

Exercise 3 : Timer and Push

pinMode
Sets pin D0 as an output. Most microcontroller pins can be used as either an input or output.

timer.setInterval
Set an interval timer for 60*1000 milliseconds (ie. 60 secs). Interval means that it will run the specified function (startWater) once every 60 secs and repeat it forever.

timer.run
This allows the timer to run.

```
BlynkTimer timer;

void stopWater()
{
  digitalWrite(D0, LOW);
}

void startWater()
{
  digitalWrite(D0, HIGH);
  timer.setTimeout(5*1000, stopWater);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
}

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

Exercise 3 : Timer and Push

startWater
Our interval timer will run this function once every 60 secs.

digitalWrite
Set D0 to high (3.3V). This will turn the pump on.

timer.setTimeout
Set an timeout timer for 5*1000 milliseconds (ie. 5 secs). Timeout means that the specified function (stopWater) will run only once without repeating.

```
BlynkTimer timer;

void stopWater()
{
  digitalWrite(D0, LOW);
}

void startWater()
{
  digitalWrite(D0, HIGH);
  timer.setTimeout(5*1000, stopWater);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
}

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

Exercise 3 : Timer and Push

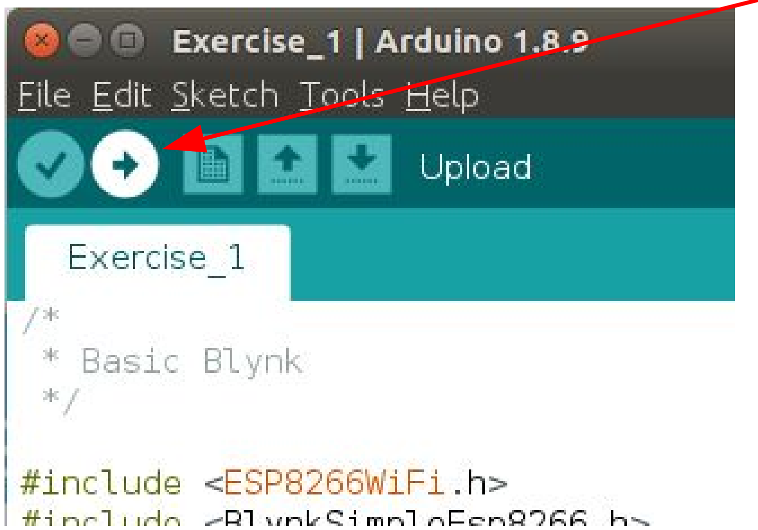
stopWater
Our timeout timer will run this function after 5 secs.

digitalWrite
Set D0 to low (0V). This will turn the pump off.

```
BlynkTimer timer;

void stopWater()
{
  digitalWrite(D0, LOW);
}

void startWater()
{
  digitalWrite(D0, HIGH);
  timer.setTimeout(5*1000, stopWater);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
}

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

Exercise 3 : Timer and Push

Connect
Connect your ESP8266 to your computer via a USB cable



```
/*
 * Basic Blynk
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

Upload
Click the upload button to upload your program into your microcontroller

Test Program
Test out your program, then try modifying it to make the LED on pin D4 blink using the timer.

A POSTERIORI
Play · Experience · Learn

Slides available at:
http://aposteriori.com.sg/other_resources

Exercise 3 : Timer and Push

Download “Exercise 3 part 2” from...

https://www.aposteriori.com.sg/other-resources/

Remember to set your own auth token, ssid, and password

recordWater()
This is a function. We’ll run it periodically using an interval timer.

timer.setInterval
Set an interval timer for 2000 milliseconds (ie. 2 secs). This will cause the timer to run the “recordWater” function every 2 seconds.

Warning: Do not set it any lower than 100 milliseconds. Excessive writes to the Blynk server will cause your account to be banned.

```
BlynkTimer timer;

void stopWater()
{
  digitalWrite(D0, LOW);
}

void startWater()
{
  digitalWrite(D0, HIGH);
  timer.setTimeout(5*1000, stopWater);
}

void recordWater()
{
  int water = analogRead(A0);
  Blynk.virtualWrite(V1, water);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
  timer.setInterval(2000, recordWater);
}

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

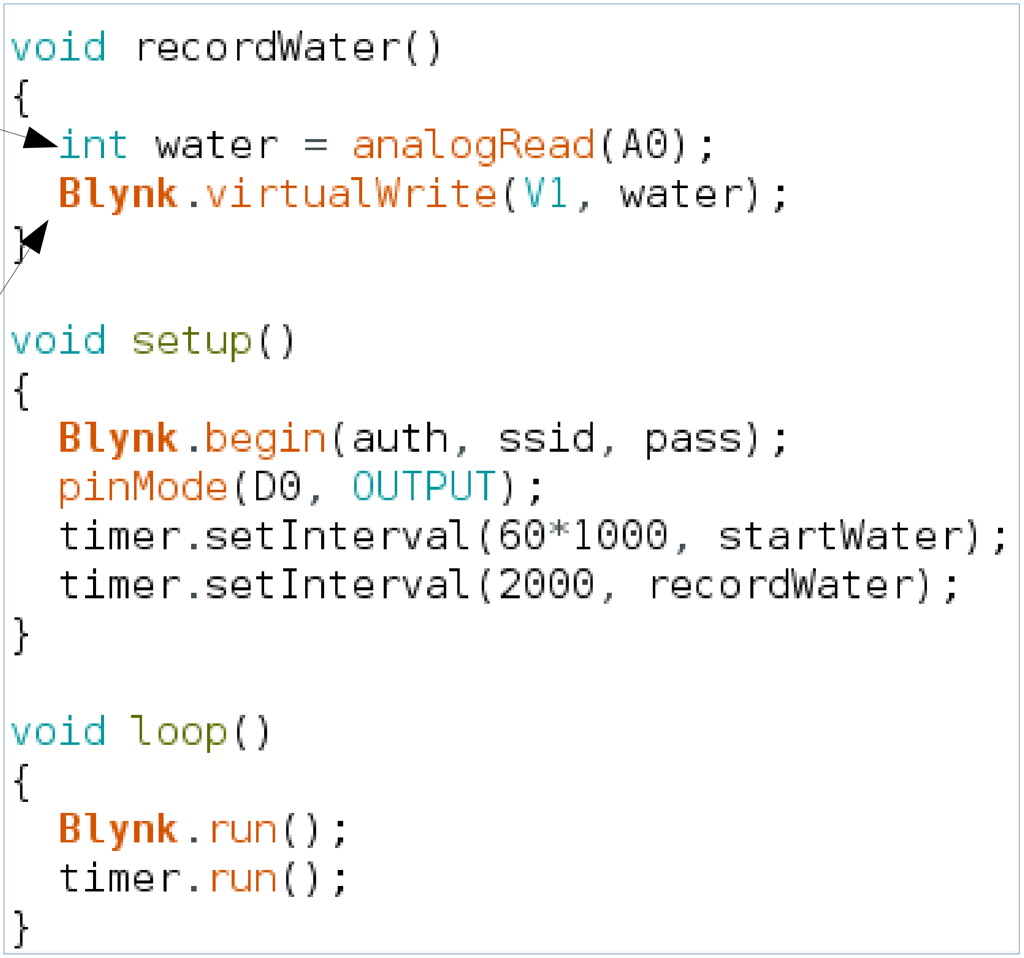
Exercise 3 : Timer and Push

analogRead()
Reads the value from pin A0 and place the value inside a interger (int) variable named “water”.

The value can range from 0 (min) to 1023 (max).

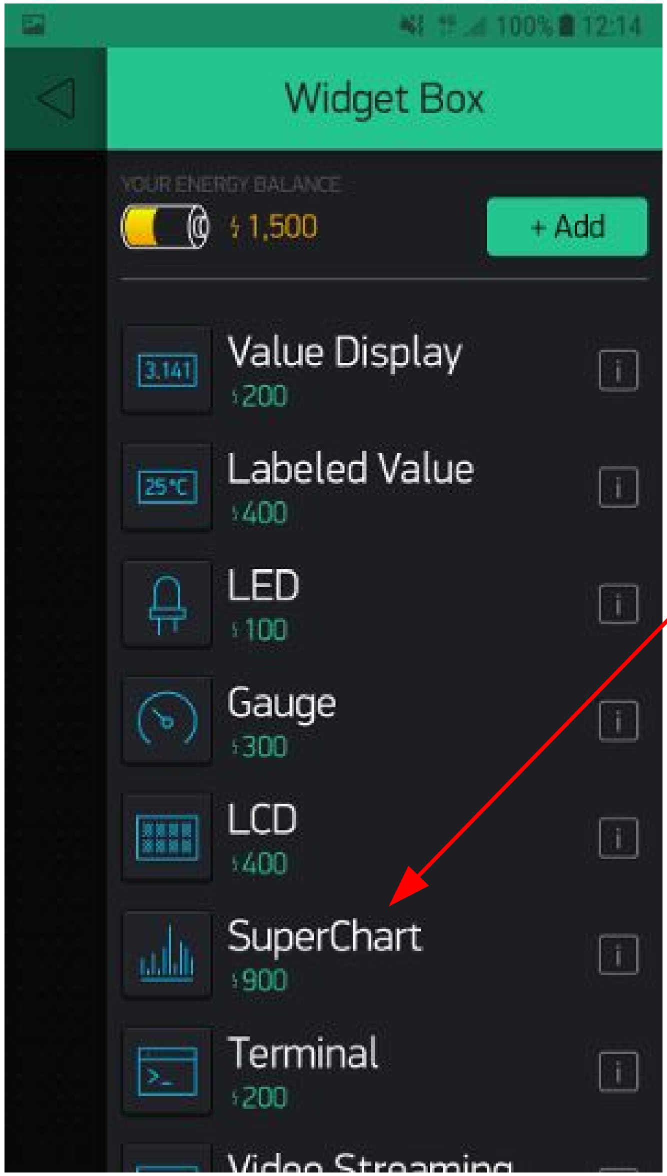
Blynk.virtualWrite
Writes the value of “water” to a virtual pin (V1). You can then read this value from your Blynk app.

The server will store these values even if your phone is off.



```
void recordWater()
{
  int water = analogRead(A0);
  Blynk.virtualWrite(V1, water);
}

void setup()
{
  Blynk.begin(auth, ssid, pass);
  pinMode(D0, OUTPUT);
  timer.setInterval(60*1000, startWater);
  timer.setInterval(2000, recordWater);
}

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

Exercise 3 : Timer and Push

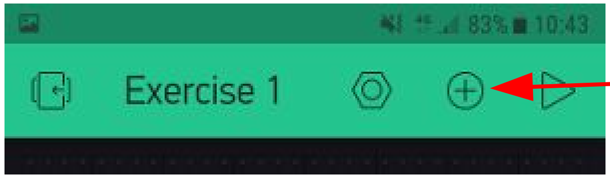

Widget Box

YOUR ENERGY BALANCE

1,500

+ Add

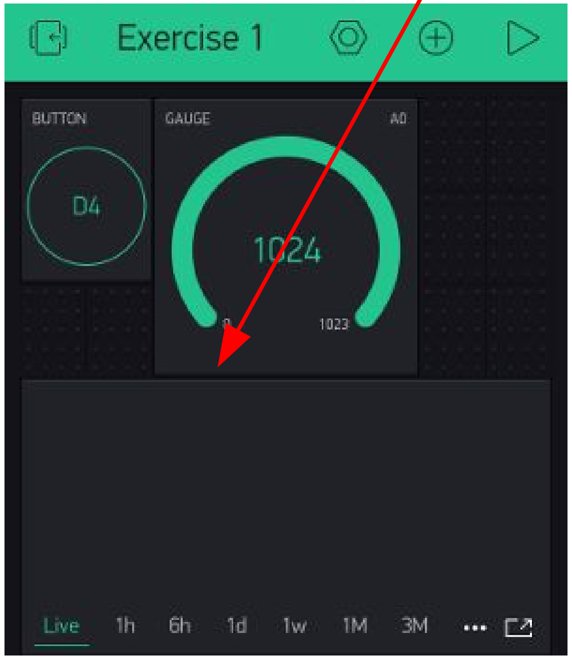
Value Display	200
Labeled Value	400
LED	100
Gauge	300
LCD	400
SuperChart	900
Terminal	200
Video Streaming	



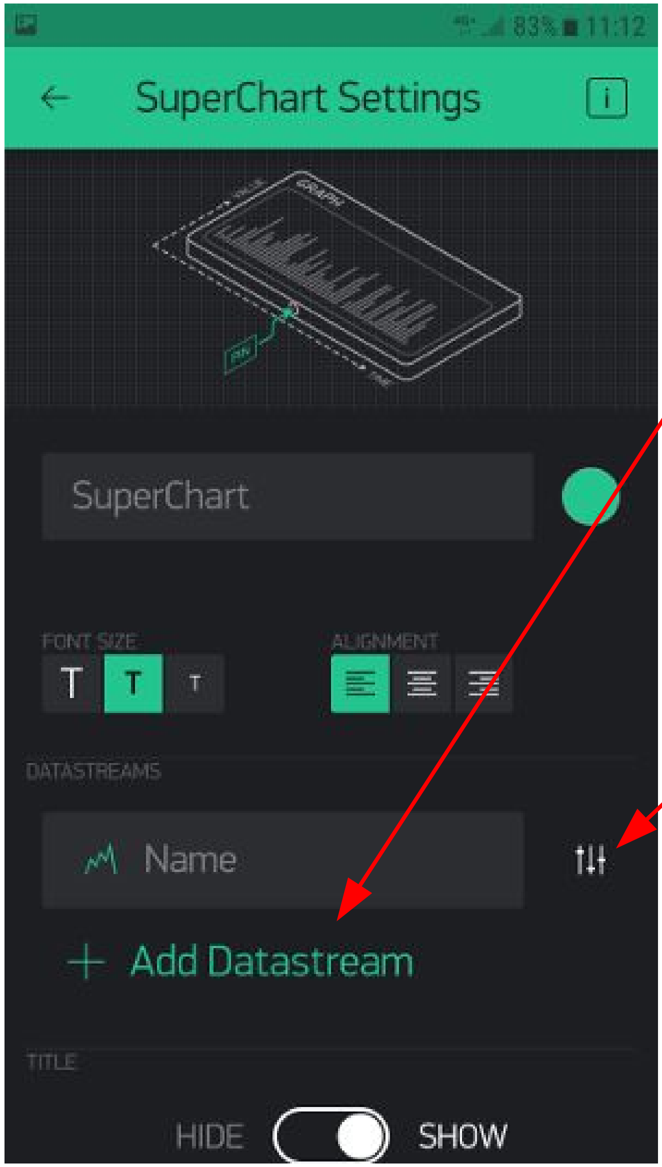
Tap the "+" to open a widget box

Tap this…

…to add a Chart to your screen



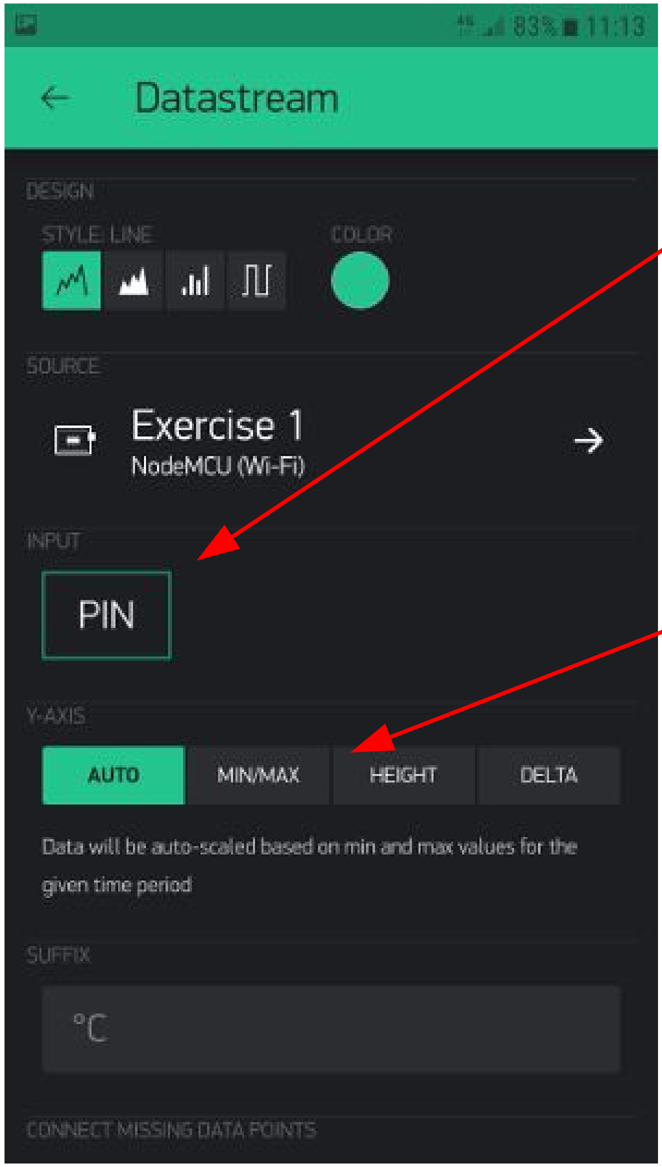
Exercise 3 : Timer and Push



Add a new Datastream
You can add more than one stream if you have multiple data sources (eg. water sensor and light sensor).

After adding the datastream, tap here to edit the datastream.

Exercise 3 : Timer and Push



Set PIN to Virtual V1
You can also use Analog A0, but if you do so, there will be no data recordings when your app is not running.

Optionally, set the Y-AXIS to MIN/MAX and set the range to…

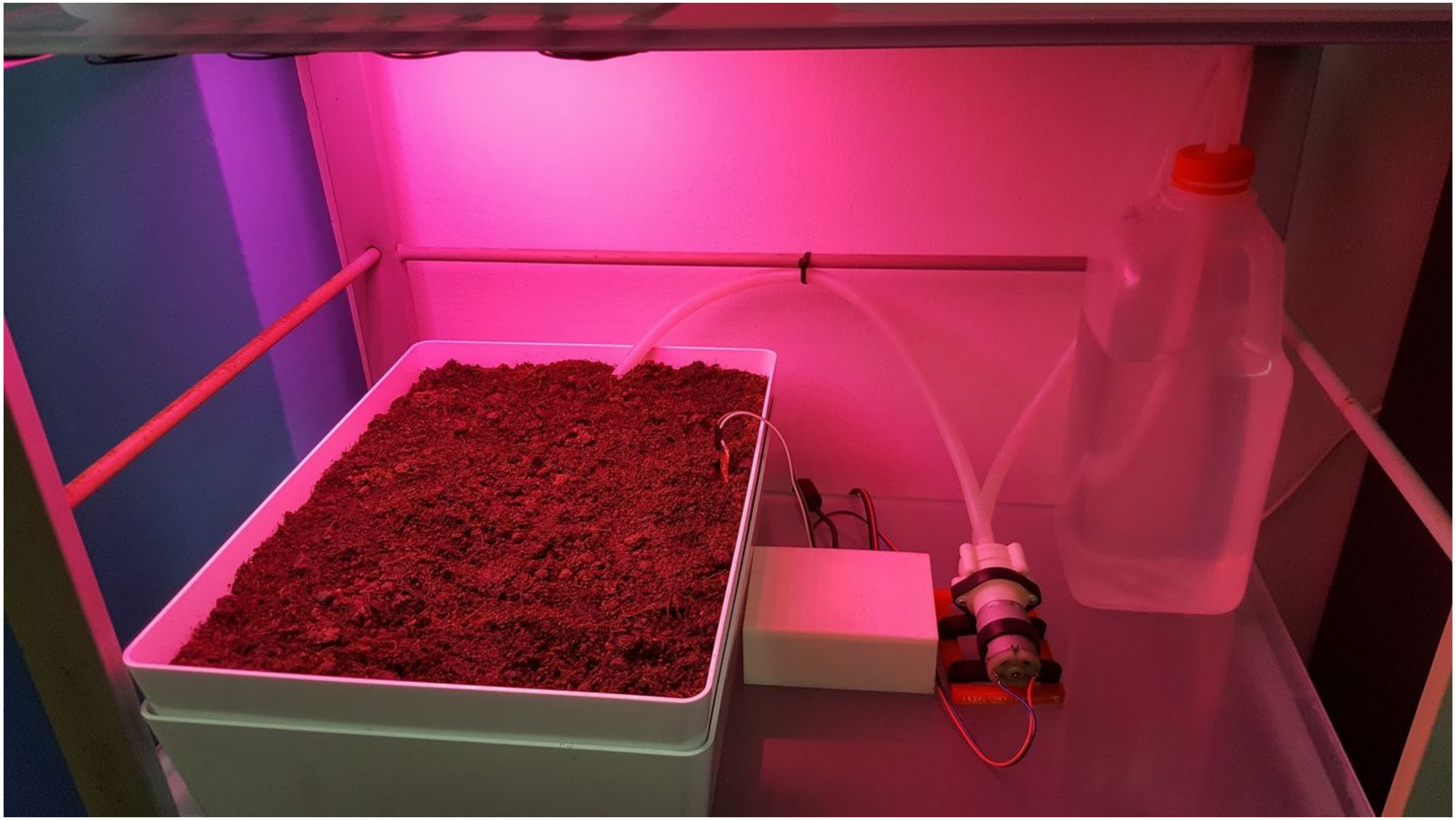
- **MIN : 0**
- **MAX : 1023.**

Challenges and Improvements

- Modify the watering function to only run if the water level is low
- Mount the pump and the ESP8266 properly
- If used for watering plants, add in controllable LED lights
- Add in warning notifications if the water level is too low

Challenges and Improvements

Plant Watering System
(complete with controllable LED lights)



A POSTERIORI
Play · Experience · Learn

Slides available at:
http://aposteriori.com.sg/other_resources

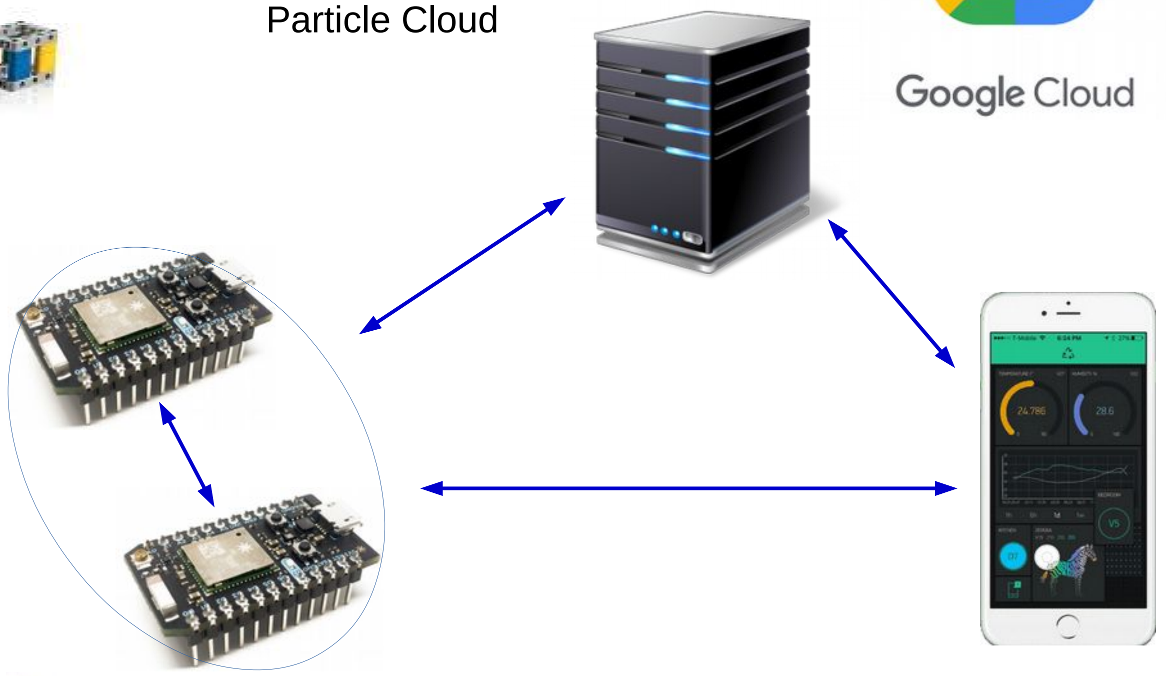
Learning More

Lego EV3 robots

Particle Electron
(3G connection)

Raspberry Pi
(Higher processing power)

Particle Cloud



Webpages /
Progress Web Apps

Learning More

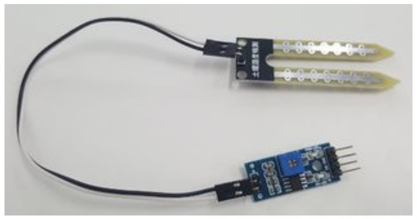
LED Strips

Servo Motors

Stepper Motors

RFID Readers

Ultrasonic
Distance Sensor



Temperature and
Humidity

Camera
(for image recognition
or QR Codes)

Wheeled motors
(for robots)

Gyro and
acceleration

Laser Distance
Sensor

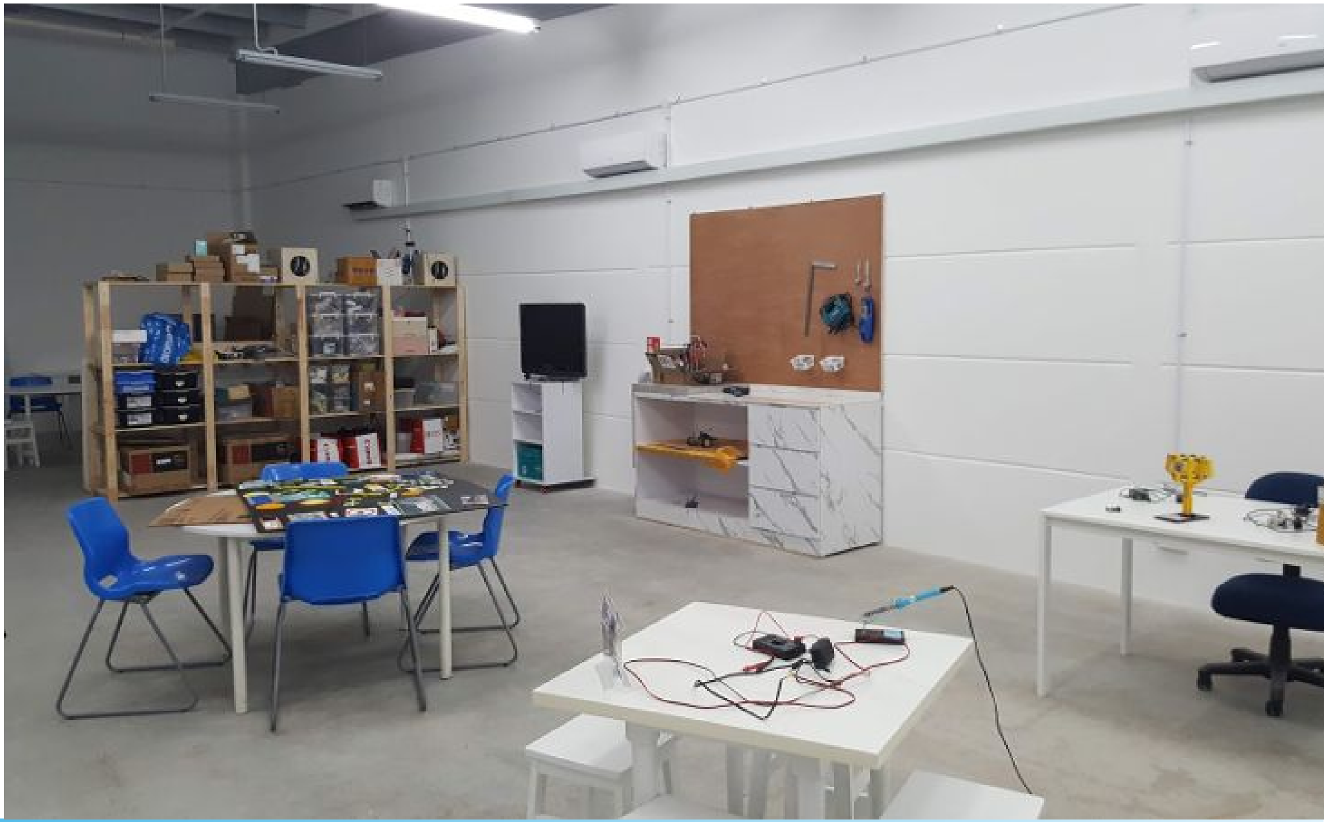
Our Company

A POSTERIORI
Play · Experience · Learn

We do...

- Electronics
- Robotics
- Drones
- Coding
- 3D Design
- Science
- Making and Tinkering

All workshop participants gets 20% off (limited time only).



Slides available at:
http://aposteriori.com.sg/other_resources

Copyright