


Snap circuits tutorial

Snap circuits are a fun support to introduce kids to circuitry and electronic prototyping. They can also be used to tackle topics related to energy saving.

 Difficulté Facile

 Durée 1 heure(s)

 Catégories Maison

 Coût 30 EUR (€)

Sommaire

Introduction

Étape 1 - 3D printing the snap part

Étape 2 - Putting it all together

Étape 3 - Circuitry activity

List of parts

Étape 4 - Electronic prototyping and coding activity

List of parts

Étape 5 - Home automation and energy saving

Étape 6 - Create a new project in the Blynk app

Étape 7 - Choose Your Hardware

Étape 8 - Auth Token

Étape 9 - Program the ESP32 board

Étape 10 - Set up the Blynk app

Notes et références

List of parts

Commentaires

Introduction

In this tutorial, you will learn how to create your own snap circuits embedding electronic components of different kinds, and how to implement educational activities revolving around this support: activities on circuitry, electronic prototyping and programming, energy saving and home automation.

Matériaux

Outils

Étape 1 - 3D printing the snap part

First, you will need to 3D print the snap part. The stl file ready for 3D printing is available [here](#).

You can also tweak this design on [tinkercad](#) and make it suitable to hold 3-legged components (ex. temperature sensor, light sensor).

A timelapse of the 3D printing process is available [here](#).

Étape 2 - Putting it all together

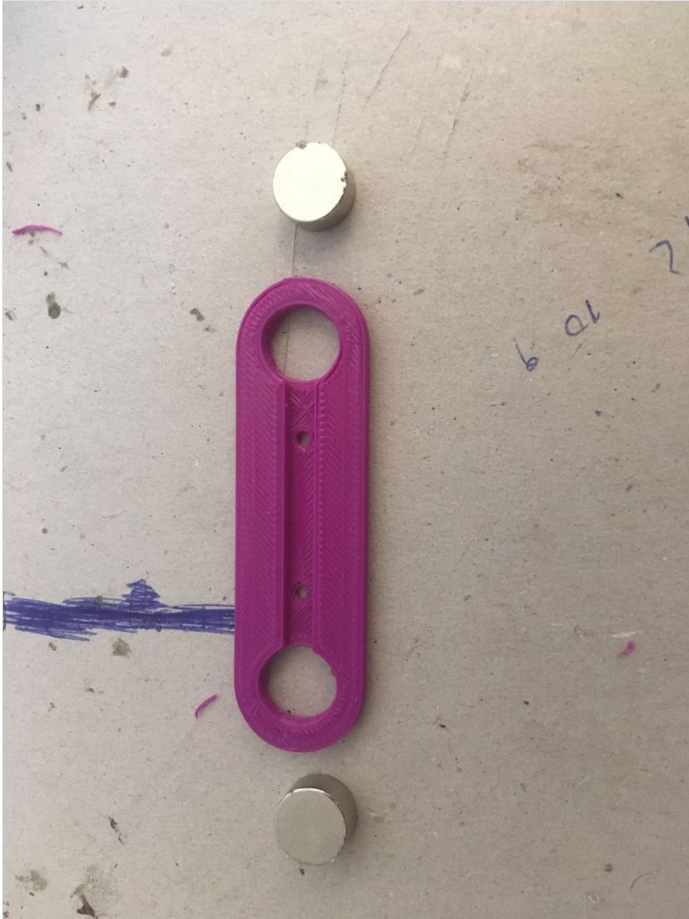
A timelapse of the assembling process is available [here](#).

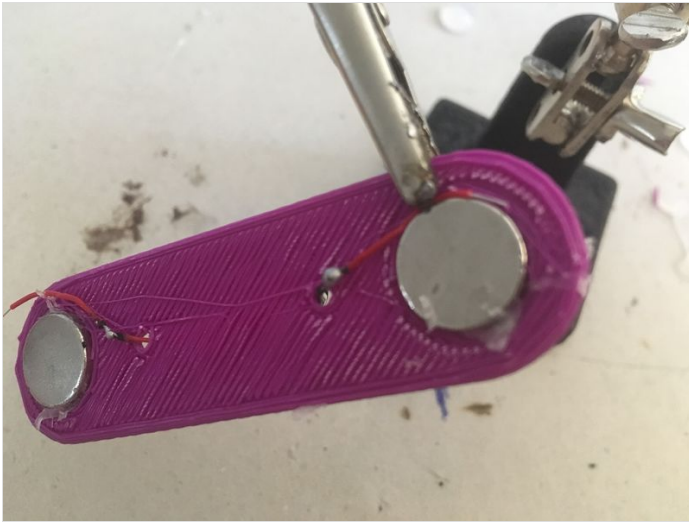
Put some hot glue around the perimeter of each hole.

Place the 2 magnets on the snap part, one per each hole. Make sure each magnet is securely fixed to the snap via the hot glue.

Next put your component into place and solder each leg onto one magnet.

You can thus create snap components containing LEDs, buzzers and vibrating motors. That's just 3 examples, however, in principle any electronic component can be mounted on a snap support.





Étape 3 - Circuitry activity

List of parts

- several snap supports
- 1 power supply (3V should be enough)
- crocodile cables

The circuitry activity is an introduction to electricity and circuits, enabled by the snap supports.

You can use different snap parts to create simple circuits, arranged in series and parallel.

To create a simple series circuits, place two snap parts (ex. LED snap and mini dc motor snap) as pictures below. Then power the circuit by plugging a battery pack (3V is enough to power a few electronic components). one end of the circuit goes to the + of the battery pack, the other end goes to the -. Beware of the polarity of the LED (anode and cathode must be connected to positive and negative of the battery pack respectively), otherwise the LED snap won't light up.

To create a simple parallel circuit, place two snap components as in the picture below.

Then power the circuit by plugging the ends of one snap part to a battery pack. Again, beware of the polarity of the LED snap. Connect the anode to the + of the battery pack, the cathode to the -.

Étape 4 - Electronic prototyping and coding activity

List of parts

- 1x Arduino board (arduino Uno or nano or any other version are Ok) + USB power cable
- crocodile cables
- jumper wires (male-male)
- snap components

In this activity, you'll be using the snap components to build an electronic system and subsequently program it via a visual coding software. [Download mBlock 3](#), not the latest version.

You can plug the snap components to the arduino board, as if it were regular electronic components. Remember that some components (ex. LED) have polarity, so make sure to connect the anode with a digital pin and the cathode to GND.

Try for example to have a snap Led blink.

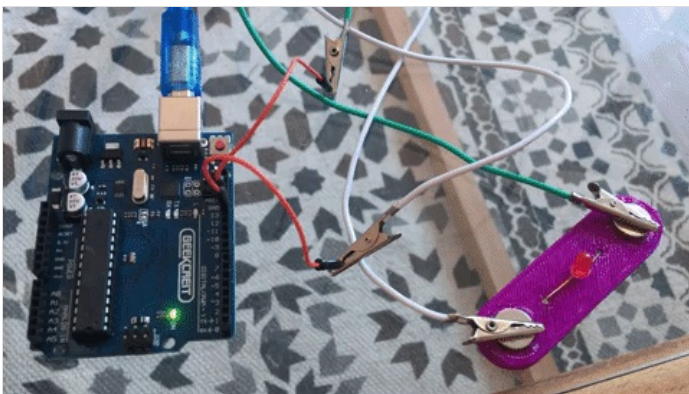
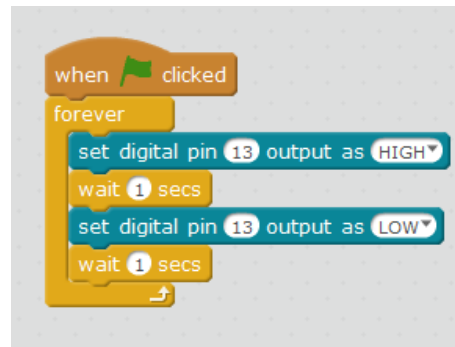
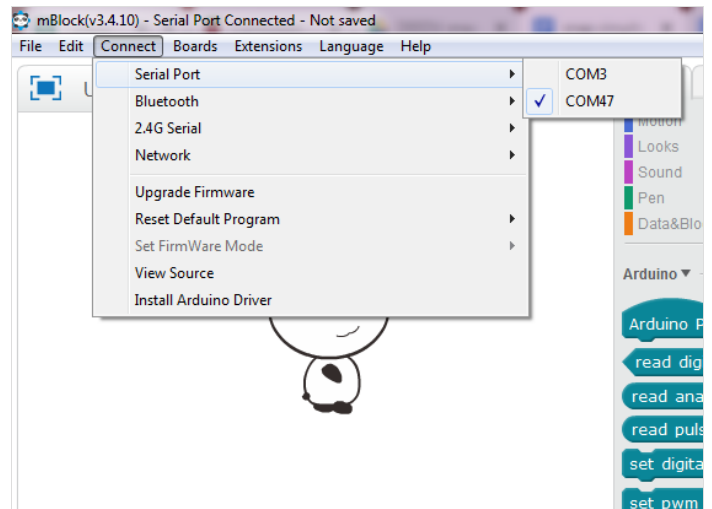
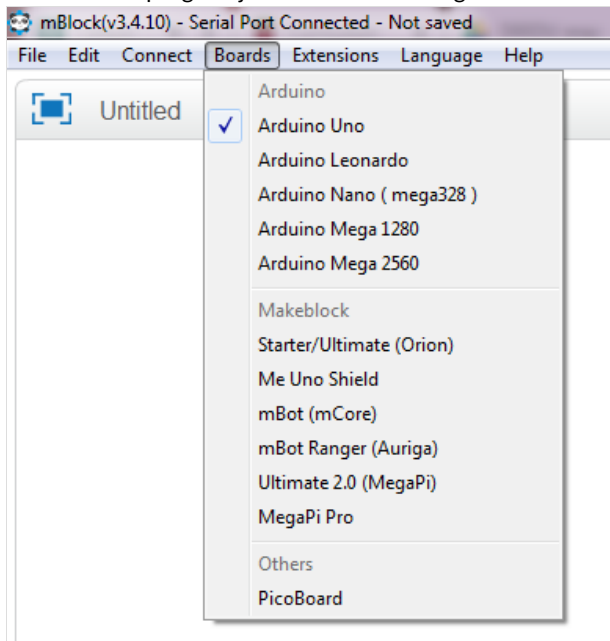
First of all, wire the snap component to the arduino board, and plug the board to your pc.

Next, launch mBlock, select the board that you re using under "Boards", and connect to it by clicking on "Connect" and selecting the correct port (in the example below COM47).

Move the available programming blocks around to obtain this:

We've connected the Led snap to pin 13, if you choose a different pin don't forget to select the correct pin in the code too.

To launch the program just hit the Green Flag.



Étape 5 - Home automation and energy saving

Snap components can be used to run activities on connect objects. It is possible for example to arrange miniature electrical appliances in a miniature house, and control them remotely. Being able to remotely control one s appliances gives the user the obvious advantage of being able to choosing when they're running and when they aren't, thus contributing to saving energy and making the miniature house as energy efficient as possible.

We've designed a number of 3D printable miniature electronic appliances that can be placed on top of a snap component. You can for example imagine to place the miniature oven on top of a Led or a miniature 3D printer on top of a mini vibrating motor snap, thus emulating real-life operations of those appliances.

Find all appliances available for 3D printing by clicking on the links below:

[snap circuit tv](#)

[snap circuit stove](#)

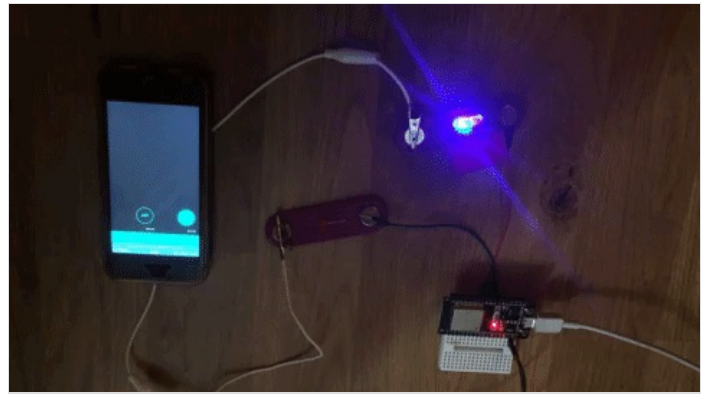
[snap circuit 3D printer](#)

[snap circuit mixer](#)

[snap circuit washing machine](#)

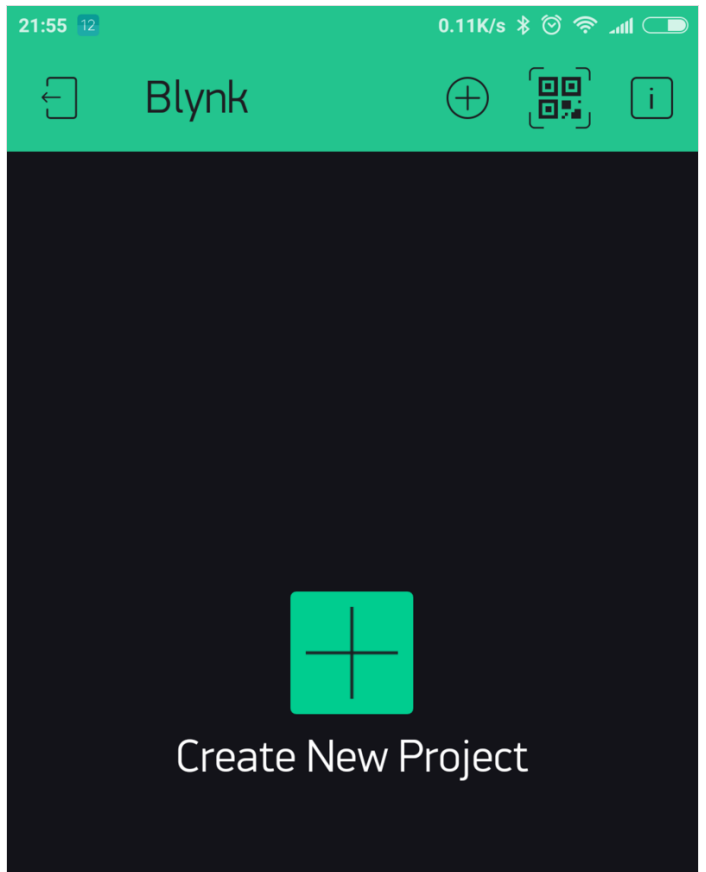
This activity will require the Blynk application. So, first download

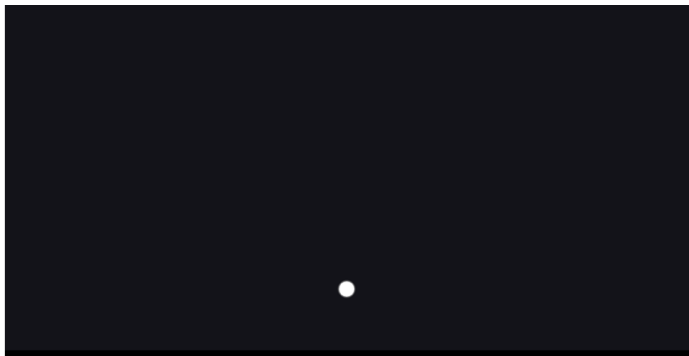
[Blynk](#) on your smartphone.



Étape 6 - Create a new project in the Blynk app

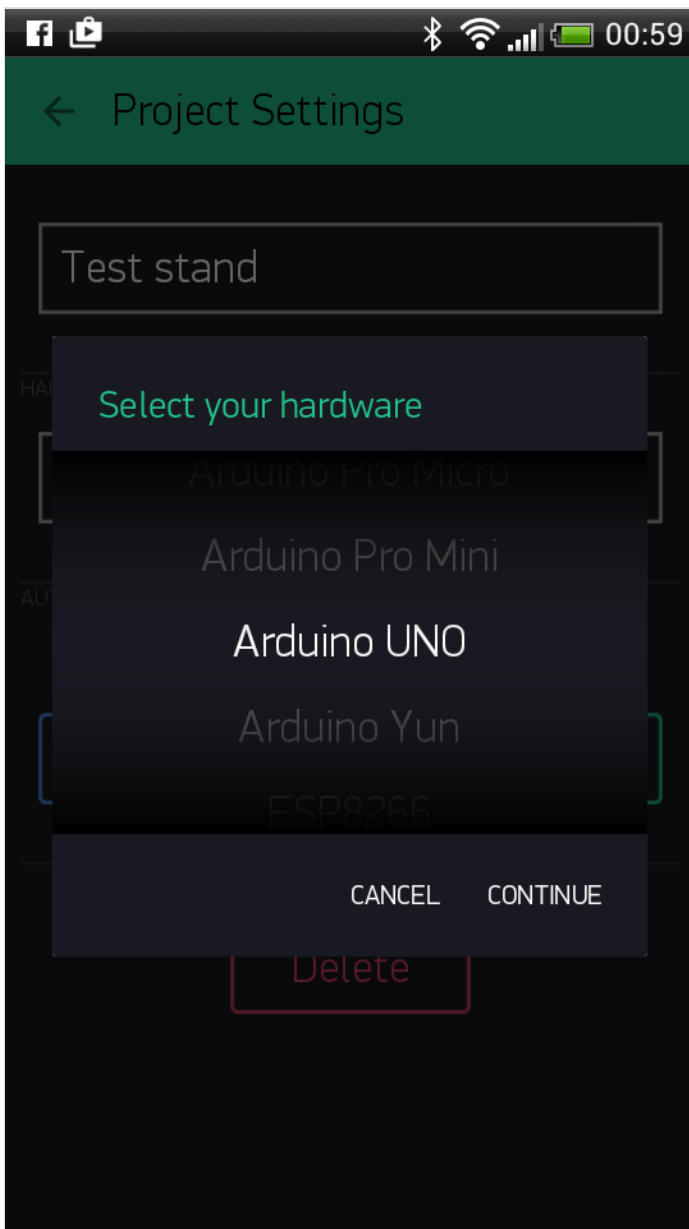
After you've successfully logged into your account, start by creating a new project.





Étape 7 - Choose Your Hardware

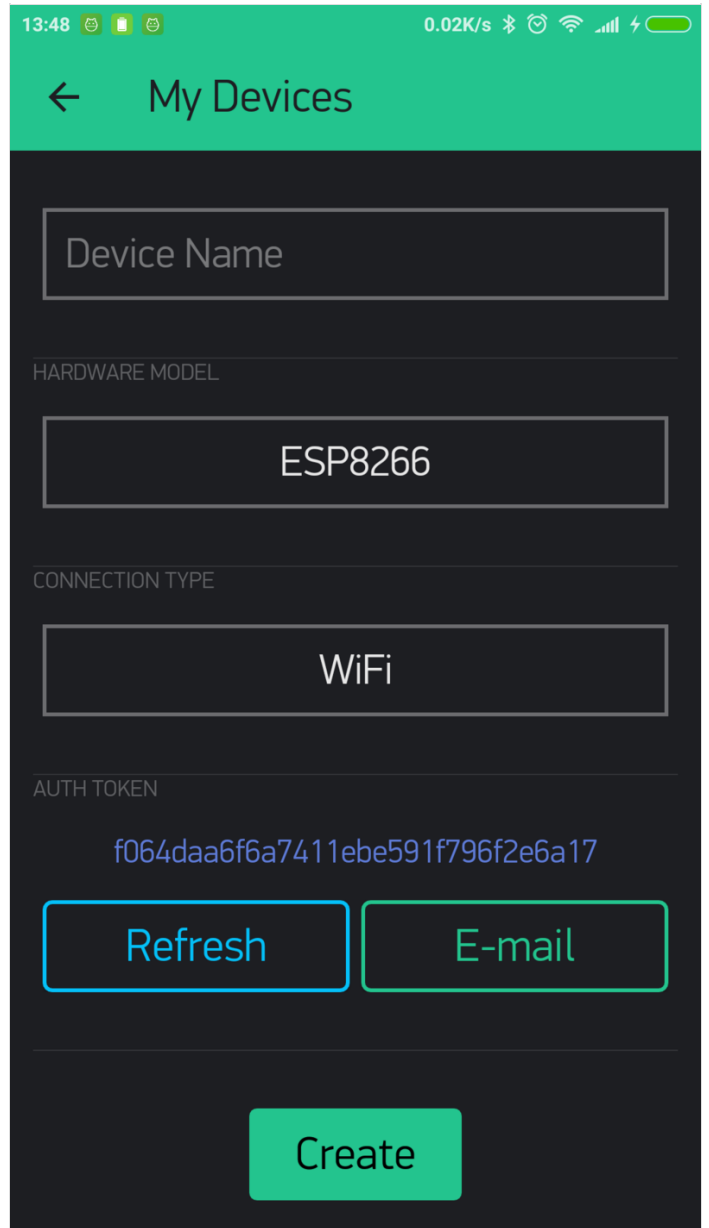
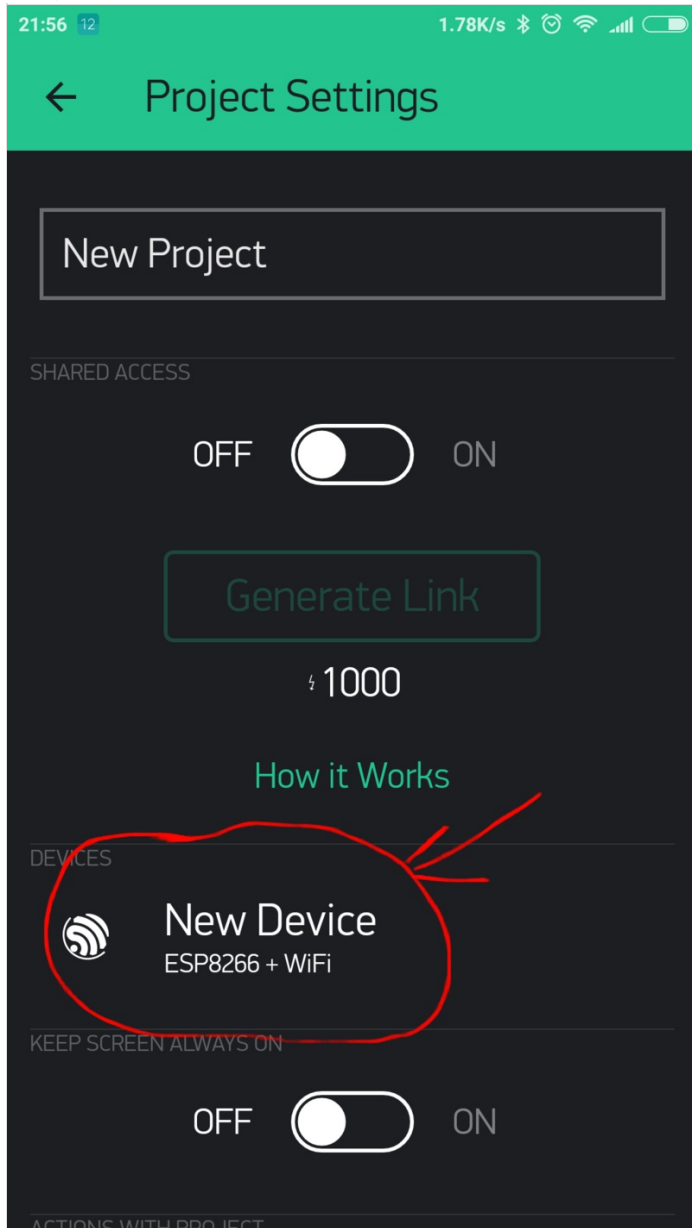
Select the hardware model you will use. If you are following this tutorial you will probably be using an ESP32 board.



Étape 8 - Auth Token

Auth Token is a unique identifier which is needed to connect your hardware to your smartphone. Every new project you create will have its own Auth Token. You'll get Auth Token automatically on your email after project creation. You can also copy it manually. Click on devices section and selected required device:

And you'll see token :



Étape 9 - Program the ESP32 board

Head to [this](#) website, select your hardware, the connection mode (ex. wi-fi) and choose the Blynk Blink example.

Copy the code and paste it on Arduino IDE (prior to that, make sure you select the correct board and the correct port - under "Tools"-).

Replace "YourAuthToken" with the token available on the app, replace "YourNetworkName" and "YourPassword" with your wi-fi credentials.

Finally, upload the code onto the board.



The screenshot shows the Blynk website interface. On the left, there are dropdown menus for 'Board' (ESP32), 'Connection' (ESP32 WIFI), and 'Example' (Blynk Blink). On the right, there is a code editor with the following code:

```
/* This is an example of how to connect an ESP32 board to the Blynk cloud.
Example.
***** */
// Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

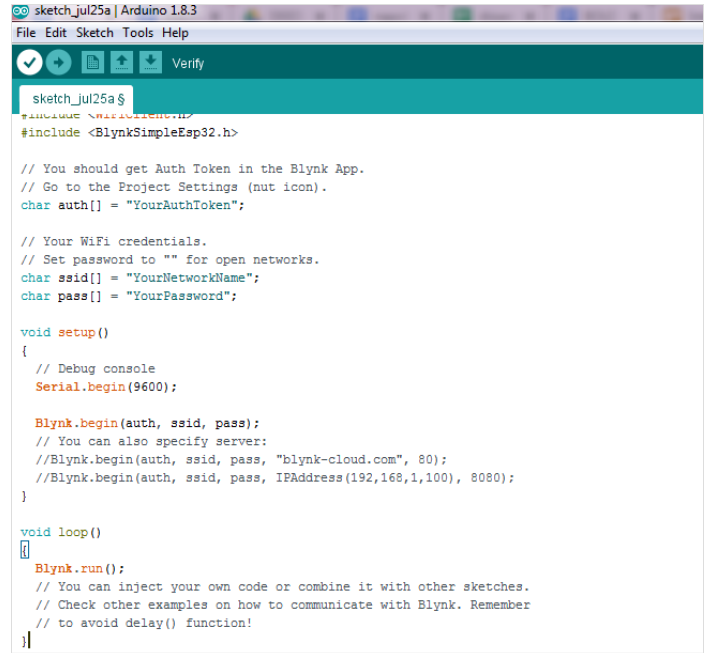
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

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

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```



The screenshot shows the Arduino IDE interface with the code for sketch_jul25a. The code is as follows:

```
sketch_jul25a $
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.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()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

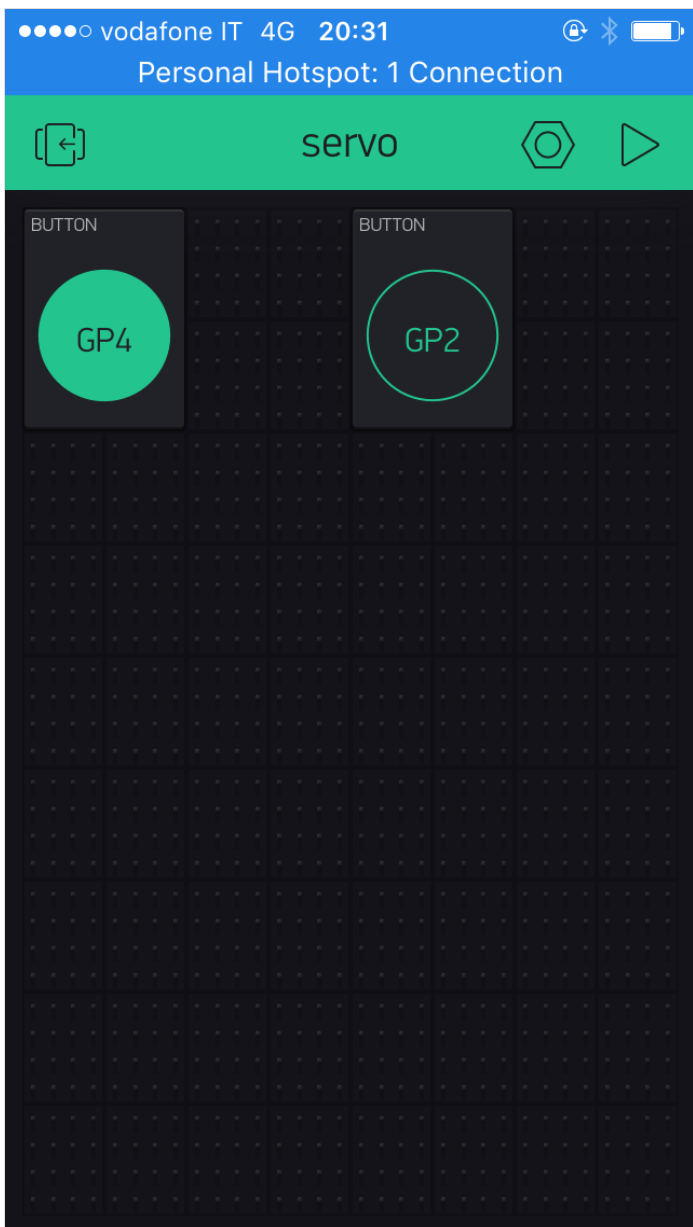
void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!
}
```

Étape 10 - Set up the Blynk app

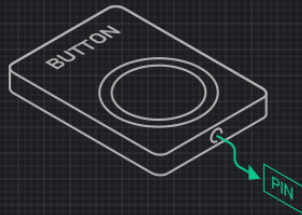
In your Blynk project, choose button widgets, as many buttons as you have snaps to control remotely. In our example we'll add two buttons widgets since we have two snap parts to control (both are LEDs).

Next select the first button and, under output, choose the port to which one of your snap is connected to the ESP32 board (ex. GP4). Make sure to have 0 and 1 next to GP4, just like in the picture below. You can also choose whether the button will function in mush or switch mode. Do the same for the second button, only this time connect to the relevant ESP32 pin (ex. GP2).

Finally, launch the app by clicking on the Play symbol. If everything goes OK, you'll be notified that your project is online, and you shall be able to remotely control your snaps.

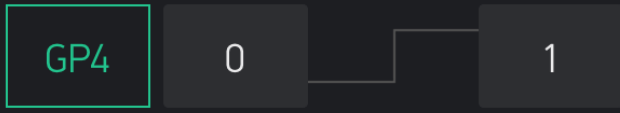


i Button Settings **OK**



Button

OUTPUT

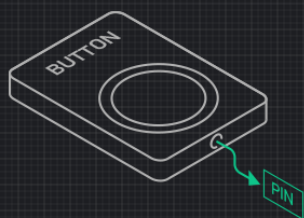


MODE

PUSH SWITCH

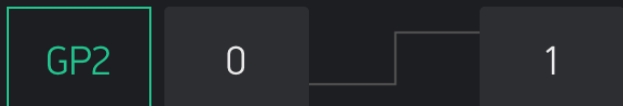
ON/OFF LABELS

i Button Settings **OK**



Button

OUTPUT



MODE

PUSH SWITCH

ON/OFF LABELS

Notes et références

List of parts

1x 3D printed snap support

1x electronic component (ex. Led, buzzer, mini vibrating motor)

2x 12x6mm magnets