


Wooden pet connects to Facebook

In this tutorial we will create a wooden pet that connects to Facebook. It will post a message of your choice on your own timeline whenever its tail is touched.

 Difficulty **Medium**

 Duration **60 minute(s)**

 Categories **Electronics**

 Cost **20 EUR (€)**

Contents

Step 1 - Setup an account and create a new feed on adafruit.io

Step 2 - Install Arduino IDE

Step 3 - Prepare the ESP32 board on your computer

Step 4 - Check that the board is correctly configured

Step 5 - Wiring the LED and the touch sensor to the ESP32

Step 6 - Place the electronic board inside the pet

Step 7 - Programming on tuniort

Step 8 - Install Adafruit_MQTT.h and Adafruit_MQTT_Client.h libraries on Arduino IDE

Step 9 - Programming on Arduino IDE

Step 10 - Create an applet on IFTTT

Step 11 -

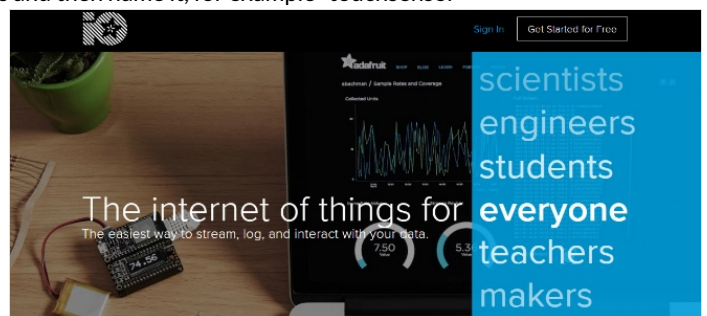
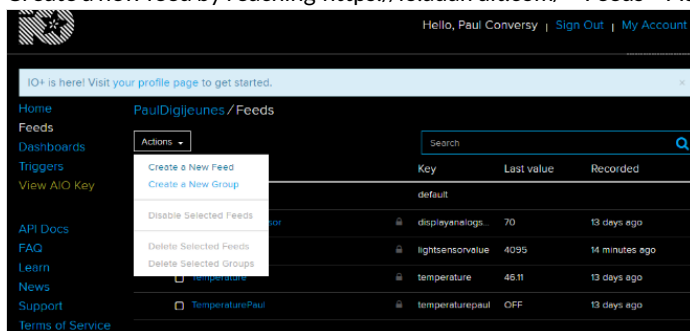
Comments

Materials

Tools

Step 1 - Setup an account and create a new feed on adafruit.io

Create a new feed by reaching <https://io.adafruit.com/> > Feeds > Actions and then name it, for example “touchsensor”



Step 2 - Install Arduino IDE

You will need to use Arduino IDE to code and upload the firmware onto your ESP32 board.

Download the software by visiting [Arduino IDE](#) > Scroll down until you see the “Download the Arduino IDE” section and choose the version based upon your operating system (e.g. If you have Windows 7, choose “Windows Installer” / if you have Windows 10, choose “Windows app”).

> On the next page choose “Just download” and run the installation files.

Step 3 - Prepare the ESP32 board on your computer

o open a Git Bash session pointing to ARDUINO_SKETCHBOOK_DIR and execute git submodule update --init --recursive

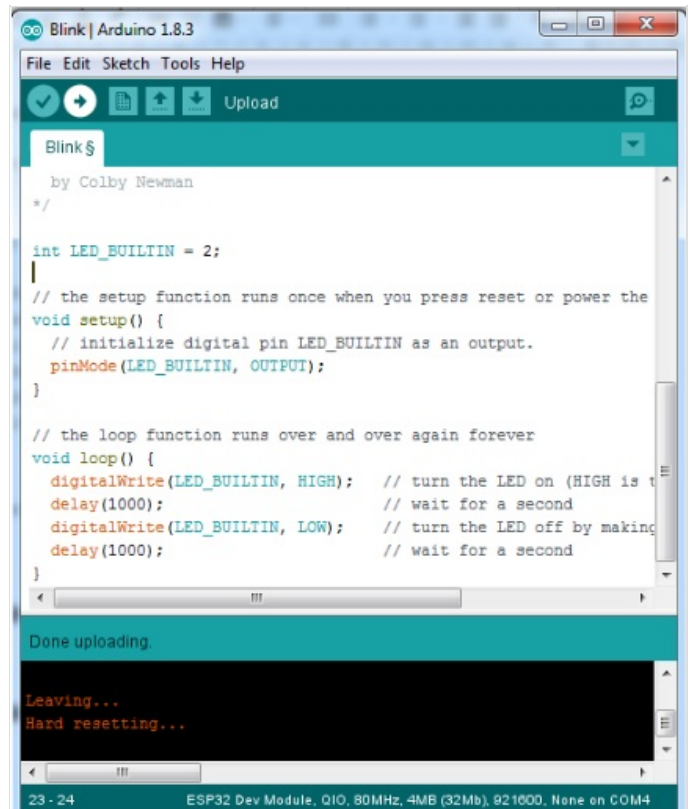
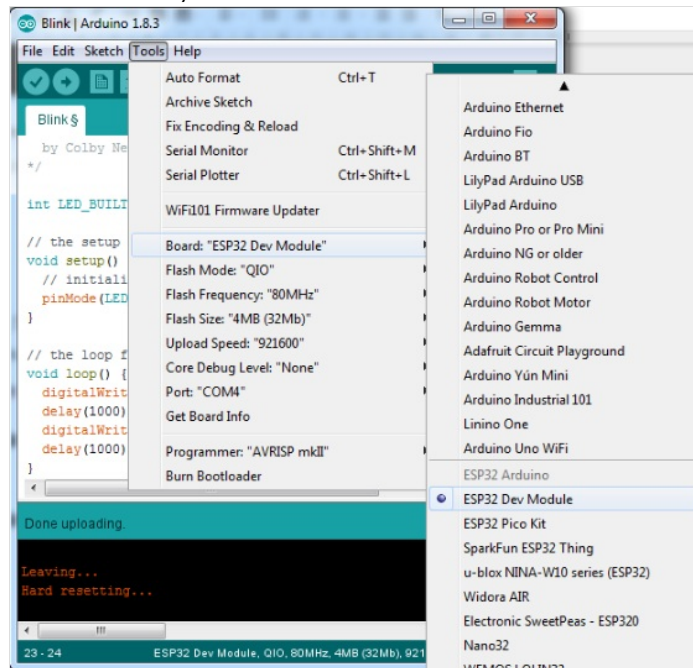
Follow the instructions provided on GitHub for your Operating System. For example, if you have Windows 7 or 10, choose “Instructions for Windows” / if you have a MacBook, choose “Instructions for Mac”
For the “Instructions for Windows” section, you can ignore the following step:

Step 4 - Check that the board is correctly configured

Launch Arduino IDE and select “ESP32 Dev Module” from the Tools menu > Board.
Fetch the Blink example from File > Examples > 01.Basics > Blink.
write `int LED_BUILTIN = 2;` at the top of the code

```
/*
ESP 32 Blink
Turns on an LED on for one second, then off for one second, repeatedly.
The ESP32 has an internal blue LED at D2 (GPIO 02)
*/
int LED_BUILTIN = 2;
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Finally, upload the code by using the right arrow (→) button at the top right corner of the window, by choosing Sketch > Upload or by pressing Ctrl+U on the keyboard.



Step 5 - Wiring the LED and the touch sensor to the ESP32

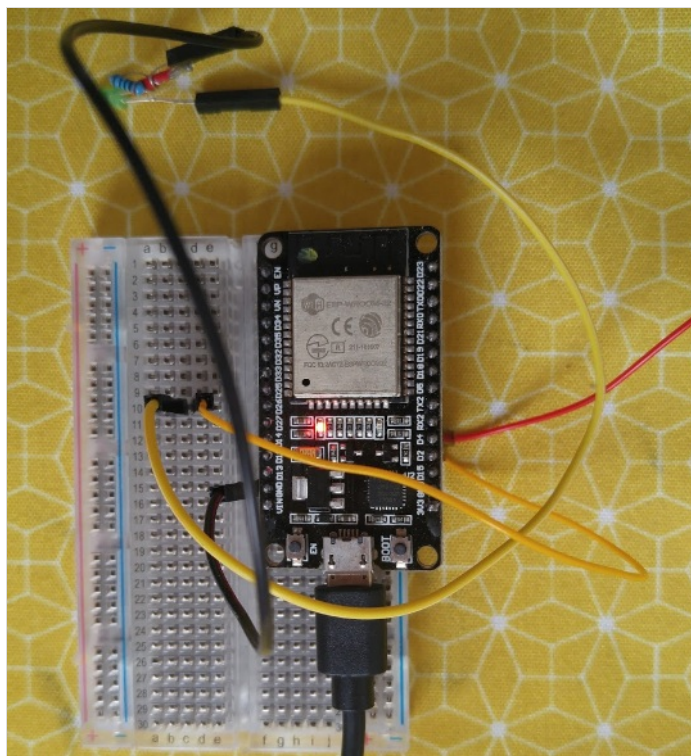
We will connect an LED to the board so that we will know when the tail actually gets touched.

Connect a male/female jumper wire to D4 on the board. D4 is by default connected to the internal touch sensor.

Connect D2 (by default connected to the board's internal LED) to the positive leg of the external LED, using a male to female jumper wire.

Connect the negative leg of the LED to GND on the board, using a male/female jumper wire.

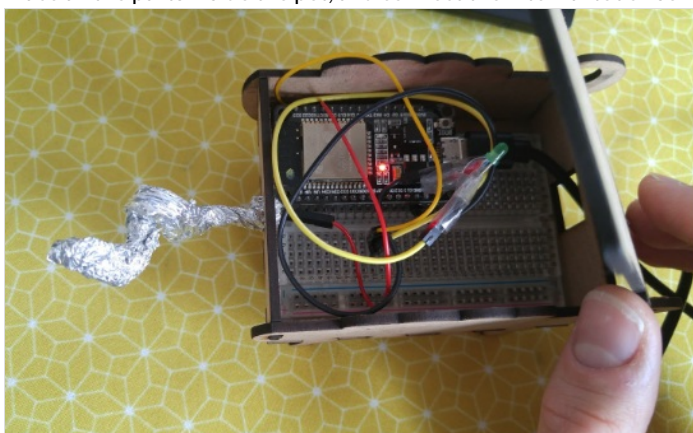
If needed, use some tape to secure the jumper wires to the LED.



Step 6 - Place the electronic board inside the pet

The wooden pet needs to be laser cut. You can download the cutting pattern for laser cutter from [here](#).

Place all the parts inside the pet, and connect the internal touch sensor to the aluminum tail.



Step 7 - Programming on tuniot

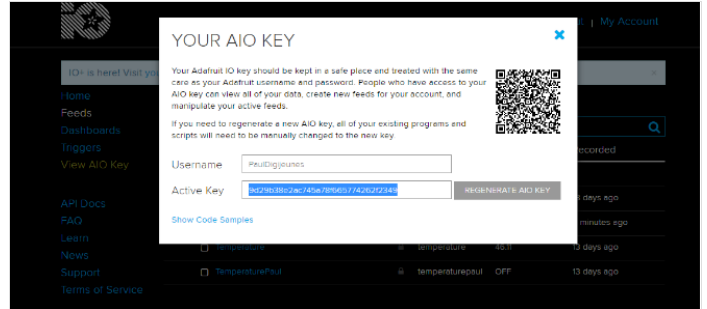
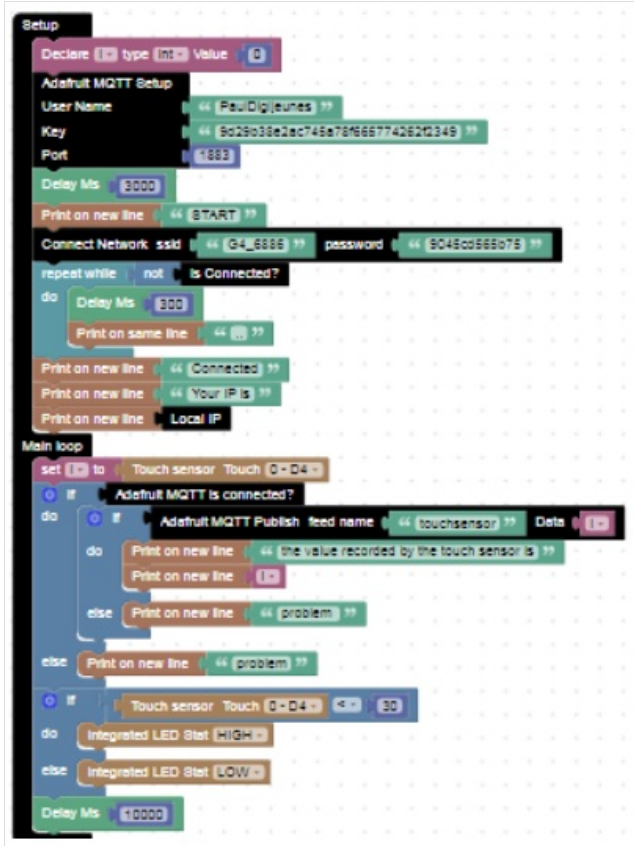
Let's create a program that registers the values recorded by the touch sensor and publishes them online.

For that we need to reach: <http://easycoding.tn/esp32/demos/code/>

Choose the appropriate blocks to create the code displayed below.

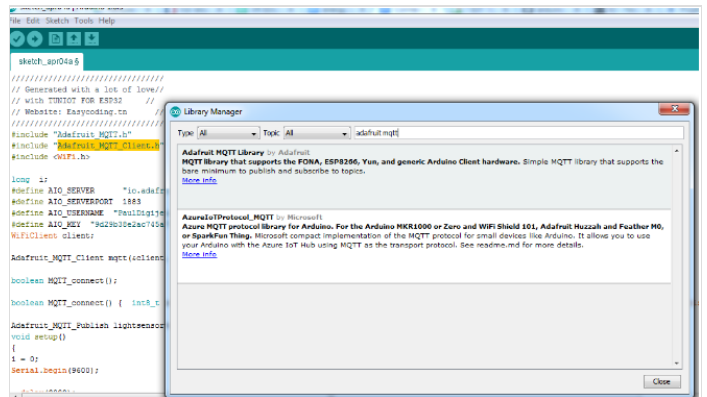
Complete the fields "ssid" and "password" with the name of your wifi-wi-fi connection and its password respectively.

The "User Name" and "Key" are available here (just click on View AIO Key).



Step 8 - Install Adafruit_MQTT.h and Adafruit_MQTT_Client.h libraries on Arduino IDE

For that we need to run Arduino IDE and go to Sketch > Include Library > Manage Libraries... > Search for "Adafruit mqtt library" and install the first result.



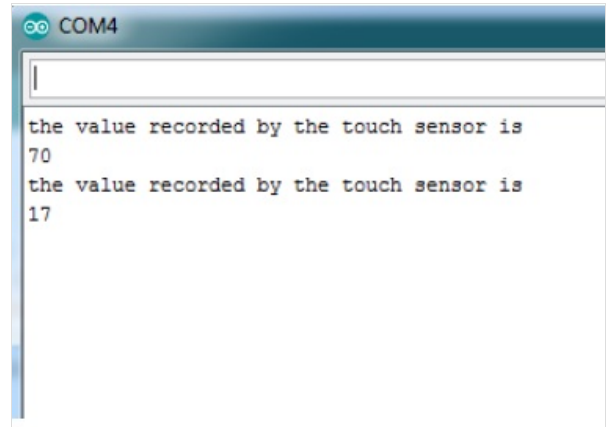
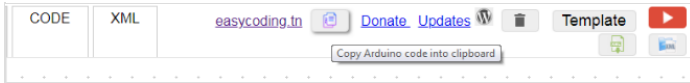
Step 9 - Programming on Arduino IDE

To upload the code on Arduino IDE, click on the “Copy Arduino code into clipboard” button.

then paste the code onto Arduino IDE, and upload it to the ESP32.

If you click on Serial Monitor (top right of the Arduino IDE screen, below the “X” button), you should see the values recorded by the touch sensor.

It means that your ESP32 is connected to the internet, and that it’s sending the value recorded by the internal touch sensor on-line, on <https://adafruit.io>.



Step 10 - Create an applet on IFTTT

For this we need to access <https://ifttt.com/> and create an account.

Then, click on My Applets.

Choose New Applet

Click on THIS

Search for Adafruit

Choose Monitor a feed on Adafruit IO

Configure it as follows

And Click on Create trigger

Click on THAT

Choose Facebook

Choose Create a status message

Type your message and click on Create action.

In this case, the message that will be posted on your Facebook timeline is “Don’t touch my tail!”.



Monitor a feed on Adafruit IO

This Trigger fires anytime it validates the data that you send to your feed. Example: If Feed Temperature > 80, fire Trigger.

Feed

The name of the feed to check.

Relationship

Relationship between two values.

Value

The value to compare against.

Create trigger

if  **then**  **that**

Create a status message

This Action will create a new plain text status message on Facebook.

Status message

Add ingredient

Create action

Step 11 -



Co-funded by the Erasmus+ Programme of the European Union