



Connected house publishes sensor data on-line

In this tutorial we will create a connected house using the ESP32 board, that publishes the values recorded by a photoresistor and an internal temperature sensor online, specifically on <https://adafruit.io>.

 Difficulty **Medium**

 Duration **120 minute(s)**

 Categories **Electronics, Energy**

 Cost **25 EUR (€)**

Contents

Introduction

Step 1 - Set up an account on adafruit.io

Step 2 - Create a new Feed on adafruit.io

Step 3 - Prepare the ESP32 board on your computer

Step 4 - Check that the board is correctly configured

Step 5 - Wiring the photoresistor and the temperature sensor to the ESP32

Step 6 - Programming on tuiot

Step 7 - Install Adafruit_MQTT.h and Adafruit_MQTT_Client.h libraries on Arduino IDE

Step 8 - Programming on Arduino IDE

Comments

Introduction

In this tutorial we will create a connected house using the ESP32 board, that publishes the values recorded by a photoresistor and an internal temperature sensor online, specifically on <https://adafruit.io>.

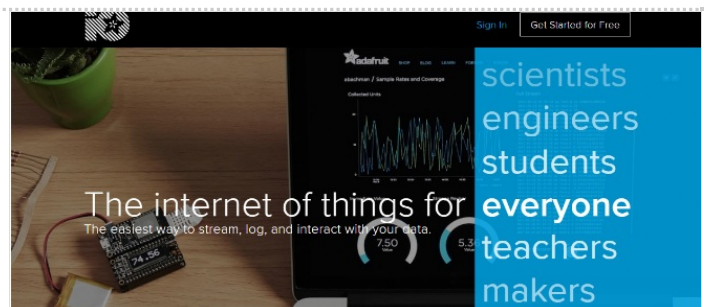
Materials

- ESP32 board
- jumper wires
- power cable
- breadboard
- light sensor

Tools

- computer with Arduino IDE and internet connection

Step 1 - Set up an account on adafruit.io



Step 2 - Create a new Feed on adafruit.io

Fichier:Connected house publishes sensor data on-line 2.PNG

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

Step 3 - Prepare the ESP32 board on your computer

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".

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.

Fichier:Connected house publishes sensor data on-line 4.PNG

Fichier:Connected house publishes sensor data on-line 3.PNG

Step 5 - Wiring the photoresistor and the temperature sensor to the ESP32

Fichier:Connected house publishes sensor data on-line 5.PNG

Fichier:Connected house publishes sensor data on-line 6.PNG

Fichier:Connected house publishes sensor data on-line 8.PNG

Fichier:Connected house publishes sensor data on-line 7.PNG

Step 6 - Programming on tuniot

Let's create a program that registers the values recorded by the photoresistor and the internal temperature 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: (see image)

1. "Variables" section - Declare 'i' type 'int' Value + "Math" section - the actual value "0"
2. "IOT" section > "Web services" subsection - Adafruit MQTT Setup (User Name and Key are retrievable from <https://io.adafruit.com/> > View your AIO Key)
3. "Various" section - Delay Ms 3000
4. "Serial" section - Print on new line 'Start'
5. "IOT" section > "IOT Station" subsection - Connect Network ssid... password... (the ssid is the name of your local wifi network connection and its password)
6. "Loops" section - repeat while.. do + "Logic" section - not + "IOT" section > "IOT Station" subsection - Is Connected
7. "IOT" section > "IOT Station" subsection - Local IP

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

Fichier:Connected house publishes sensor data on-line 10.PNG

Fichier:Connected house publishes sensor data on-line 9.PNG

Step 7 - Install Adafruit_MQTT.h and Adafruit_MQTT_Client.h libraries on Arduino IDE

Fichier:Connected house publishes sensor data on-line 11.PNG

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 8 - 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 light sensor and by the temperature sensor:

It means that your ESP32 is connected to the internet, and that it's sending the value recorded by the photoresistor and by the internal temperature sensor online, on <https://adafruit.io>.

Fichier:Connected house publishes sensor data on-line 12.PNG

Fichier:Connected house publishes sensor data on-line 13.PNG
