ESP8266

FOLLOWERS COUNTER ON INSTAGRAM

ESILV LÉONARD DE VINCI
ÉCOLE D'INGÉNIEURS
PARIS-LA DÉFENSE

Eden Berro – Chaïma Debiche

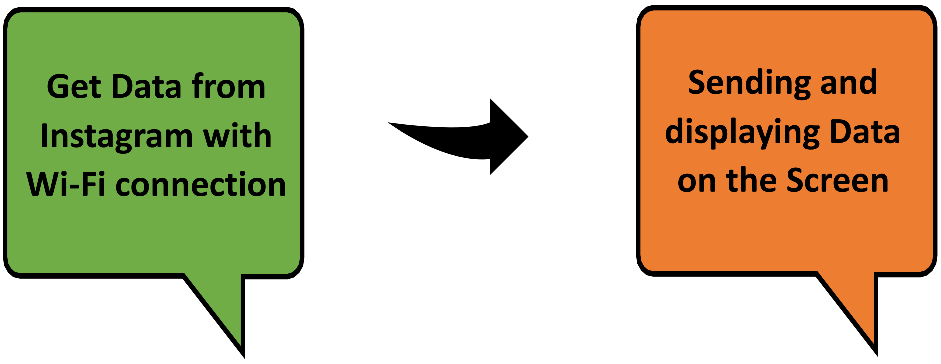
Table des matières

PRESENTATION OF THE PROJECT

The project consists in the creation of an object connected to the outside world and the direct world. Its main function will be to display the value of the number of subscribers of an Instagram account (link with the external world by Internet connection).

The Arduino card read the value in real time and it will be displayed on a screen.

The 2 mains functions that make up our object are therefore:



Get Data from Instagram with Wi-Fi connection

Sending and displaying Data on the Screen

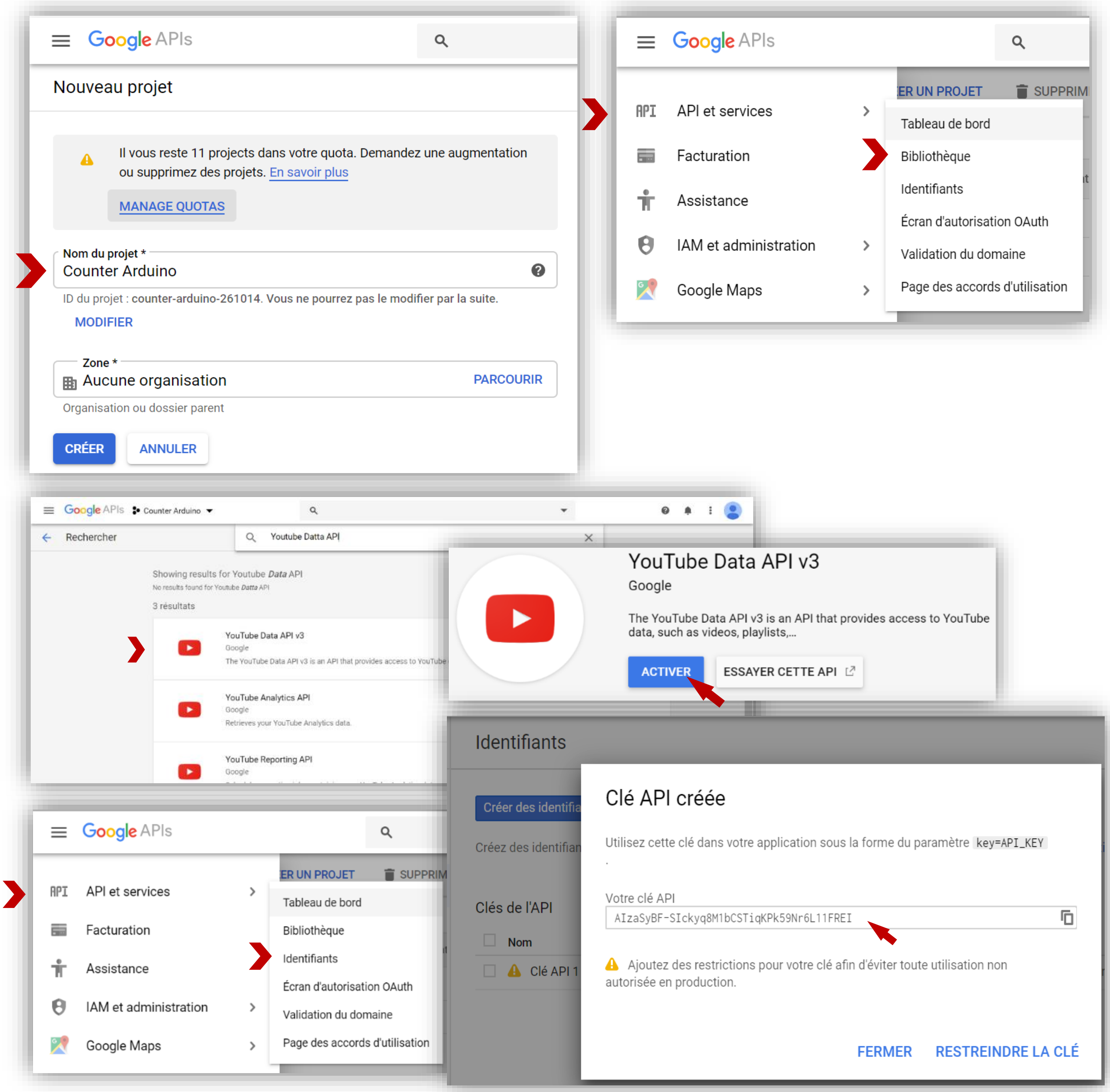
MATERIALS

We will use on this project:

- 4 modules of 8 x 8 matrices of Leds
- Wires
- A portable batterie (to power the Arduino card without being connected to the computer)

I. FIRST ATTEMPT

At the beginning, we wanted to get data from a YouTube account, like the number of followers or the number of views of a video. To get data from YouTube, we must create a Google Account. Then we must recover an API key (Application Programming Interface). To collect this, we go on the google account console: https://console.developers.google.com/cloud-resource-manager and follow these different steps to create an ID and get the key.



Nouveau projet

Il vous reste 11 projects dans votre quota. Demandez une augmentation ou supprimez des projets. En savoir plus

MANAGE QUOTAS

Nom du projet * Counter Arduino

ID du projet : counter-arduino-261014. Vous ne pourrez pas le modifier par la suite.

MODIFIER

Zone * Aucune organisation PARCOURIR

Organisation ou dossier parent

CRÉER ANNULER

Showing results for Youtube *Data* API

No results found for Youtube *Datta* API

3 résultats

YouTube Data API v3
Google
The YouTube Data API v3 is an API that provides access to YouTube

YouTube Analytics API
Google
Retrieves your YouTube Analytics data.

YouTube Reporting API
Google

YouTube Data API v3

Google

The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,...

ACTIVER ESSAYER CETTE API

Identifiants

Créer des identifia

Créez des identifian

Clés de l'API

□ Nom

□ Clé API 1

Clé API créée

Utilisez cette clé dans votre application sous la forme du paramètre `key=API_KEY` .

Votre clé API

AIzaSyBF-SIckyq8M1bCSTiqKPk59Nr6L11FREI

Ajoutez des restrictions pour votre clé afin d'éviter toute utilisation non autorisée en production.

FERMER RESTREINDRE LA CLÉ

To use the API and get data from YouTube we should wait many days to have the agreement. So, we decided to choose another network and get one that doesn't need a long protocol to get access to the data. We chose Instagram. There is a protocol to get their access, but someone create a library that is more simplify and allows us to use data from Instagram.

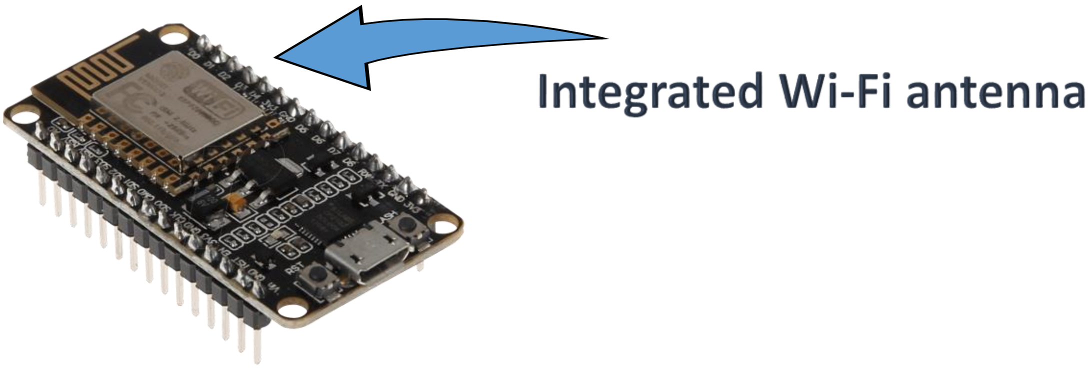
II. PRESENTATION OF THE MATERIALS

1. ESP8266 NODE MCU

The ESP8266 Node MCU is a cheap microcontroller (between 4 and 10 €) with integrated Wi-Fi connection.

You can use the module to:

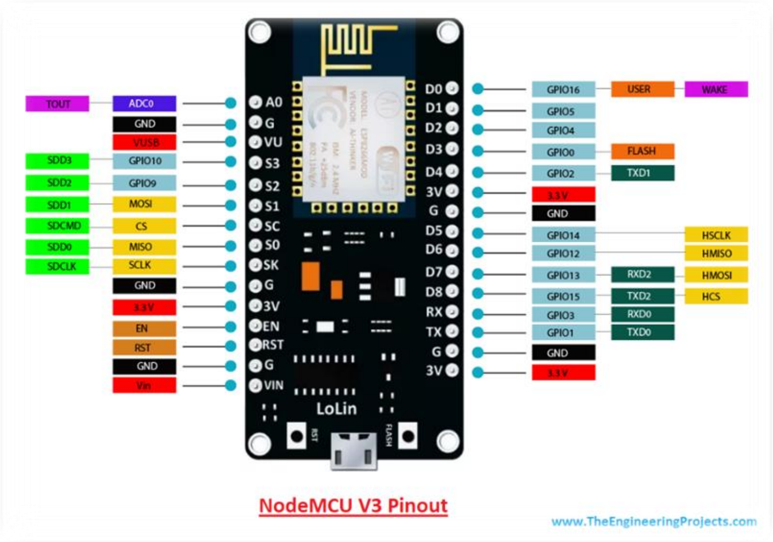
- Create a web server,
- Communicate with a web server,
- Create a mesh network,
- Create wireless connections.



We must install the library of the ESP8266 so Arduino can recognize the card in the network. And before that, we must install the driver on this link so our computer can recognize the card. First, we didn't know that there was a driver to install so we can a short time to understand how we can recognize the card and what driver we must install. This is the link of the driver:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

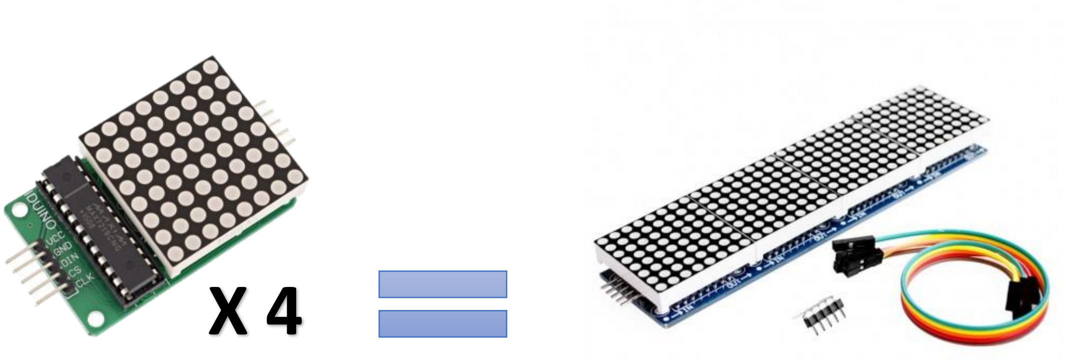
The pins of the ESP8266 are different to the Arduino. To see the similarity between these we must check the Datasheet of the ESP8266. We will use the pins DO, D1, and D2 (it means in Arduino: GPIO16, GPIO 5 and GPIO 4). This is the datasheet:



2. MAX 7219

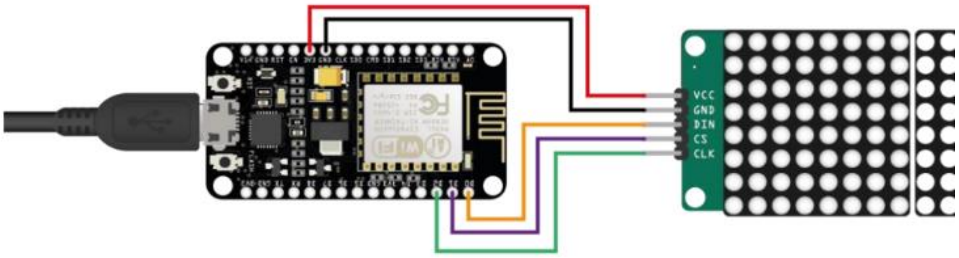
An LEDS array is a display composed of LEDS. The standard version has 8 rows and 8 columns. We call it "8x8 matrix" because it has 64 LEDS in total. In this project, we will use an 8x32 matrix!

The LEDS array connects to the ESP8266 using 5 pins as we will see after.



Cabling:

Below you will find our wiring:



We plug the Data IN signal in D7 of the ESP8266 that moves the data in the display module. The data is loaded on the rising edge of CLK generated by the master. CLK is plugged on D5 (in the schema).

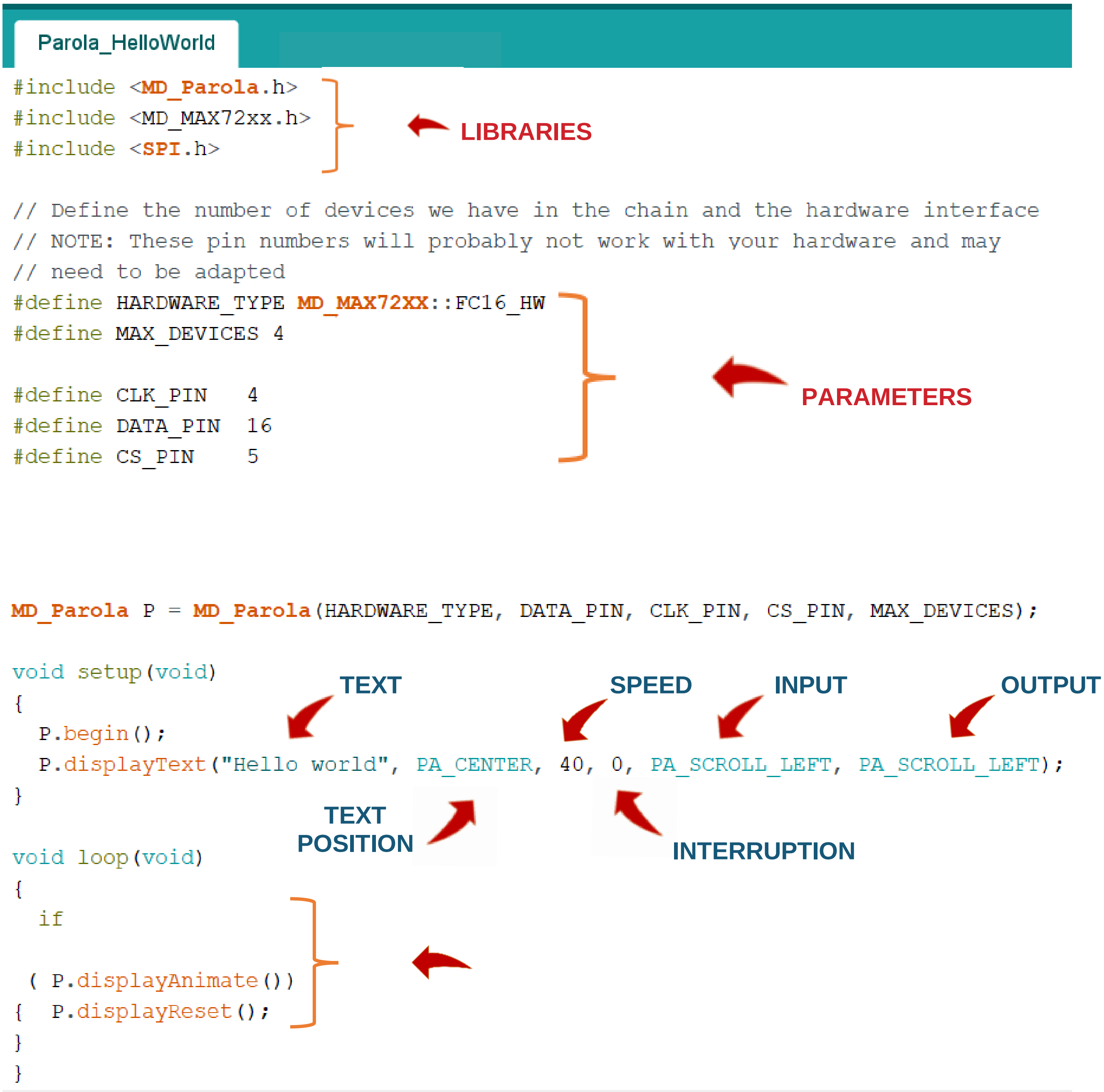
The pin CS DIN and CLK are using to send the data to the matrix. The remaining interface pins are for + 3,3V and GND.

Attention: It’s important to have enough current for the number of modules connected.

DISPLAY DATA ON THE SCREEN

HELLO WORLD

First, we must display on the led matrix, a predefined text in the code. Example "Hello World ». The goal is to understand how work the led matrix to display a normal text before displaying a text from internet.



```
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>

// Define the number of devices we have in the chain and the hardware interface
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 4

#define CLK_PIN   4
#define DATA_PIN  16
#define CS_PIN    5



MD_Parola P = MD_Parola(HARDWARE_TYPE, DATA_PIN, CLK_PIN, CS_PIN, MAX_DEVICES);

void setup(void)
{
  P.begin();
  P.displayText("Hello world", PA_CENTER, 40, 0, PA_SCROLL_LEFT, PA_SCROLL_LEFT);
}

void loop(void)
{
  if

 ( P.displayAnimate())
{  P.displayReset();
}
}
```

LIBRARIES

So, we use different libraries:

« MD_Parola.h » which is a library, which allows to display on a led matrix.
It provides functions to simplify the implementation of special text effects on the LED matrix.
This library avoids coding the necessary pixels to the display of a character.

The second libraries « MD_MAX72xx.h » allows the programmer to use the LED matrix as a pixel addressable display. And the last « SPI.h » is a library that is used to communicate equipment together at a short distance with high data transmission speed.

PARAMETERS

Then we will define the parameters of the card, like here the type of card is « FC16_HW ». We found this in the documentation of module, the total number of modules and the wired pins.

Then, we will declare a variable P which will have as characteristics all the parameters defined (look the screen of the code).

FUNCTION

In the setup function:

- First, we initialize P, in order to restore the display and avoid errors at startup.
- Second, with the instruction « P.displayText », we configure what we're going to display and how it's going to be displayed (effects).

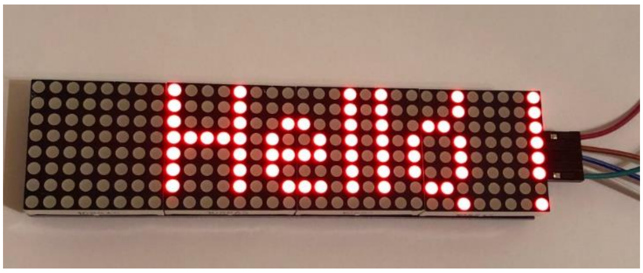
Here, we're going to display "hello world", with the following effects:

The text will scroll to the left at the input and the same for the output. We put 40 for the scrolling speed and without any interruption (0).

In the loop function:

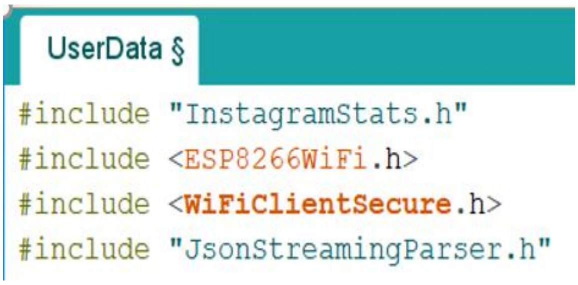
It is necessary to use the instruction « P.displayAnimate » to perform the effects on the led matrix.
Moreover, we have a condition. If we have an animation, then we reset the display.

We get this:



Second, we have created a function to retrieve data from Instagram.

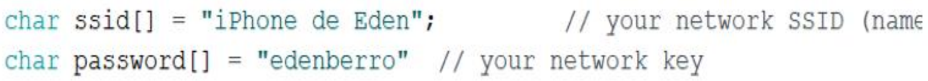
GET USER FROM INSTAGRAM



```
#include "InstagramStats.h"
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include "JsonStreamingParser.h"
```

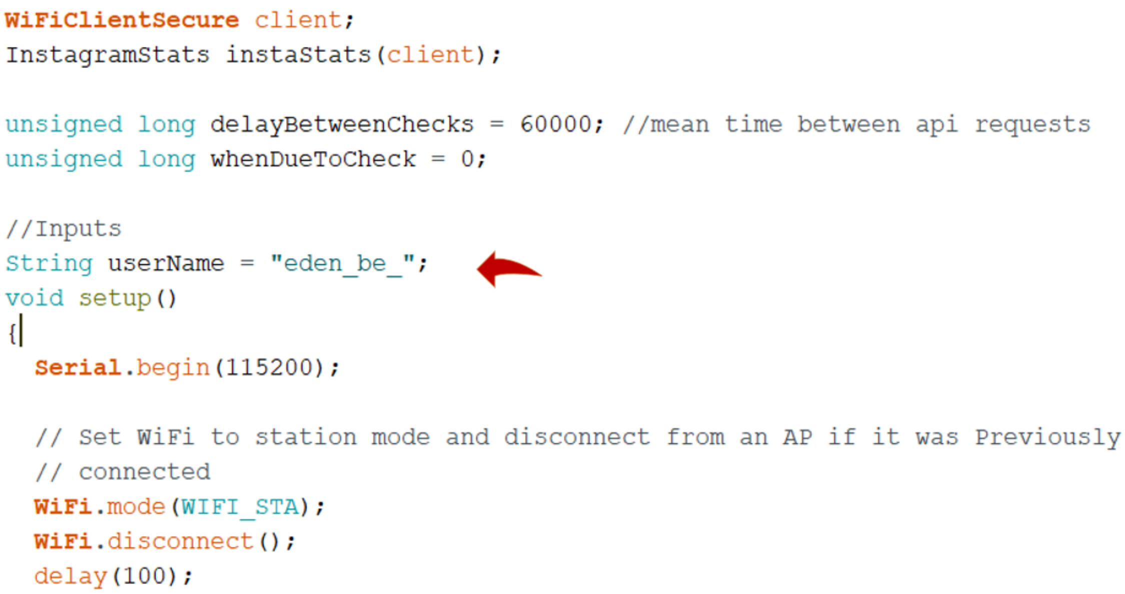
Libraries:

We use these libraries to get the data from Instagram, to connect the ESP module to the Wi-Fi and to understand and analyze the language of the Data getting from Instagram that is in Json and to translate it to the network. WificlientSecure works with the precedent libraries and we want a Wi-Fi secure.



```
char ssid[] = "iPhone de Eden";         // your network SSID (name
char password[] = "edenberro"  // your network key
```

We connect the ESP to the Phone’s internet.

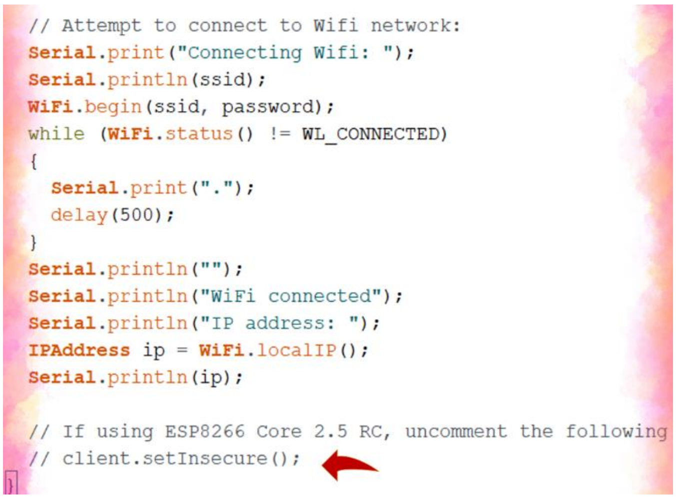


```
WiFiClientSecure client;
InstagramStats instaStats(client);

unsigned long delayBetweenChecks = 60000; //mean time between api requests
unsigned long whenDueToCheck = 0;

//Inputs
String userName = "eden_be_";
void setup()
{
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was Previously
  // connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
```

We create 2 variables named **delayBetweenChecks** and **WhenDueToCheck** that we will used at the end of the code to the delay of the asking. Unsigned because it can’t be negative.

Then, we create a variable string named with the name of the Instagram account.

In the setup we will create a function that connect the ESP to the Wi-Fi. We need more debts, so we put a big number of debts: 115200.



```
// Attempt to connect to Wifi network:
Serial.print("Connecting Wifi: ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
  Serial.print(".");
  delay(500);
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
IPAddress ip = WiFi.localIP();
Serial.println(ip);

// If using ESP8266 Core 2.5 RC, uncomment the following
// client.setInsecure();
```

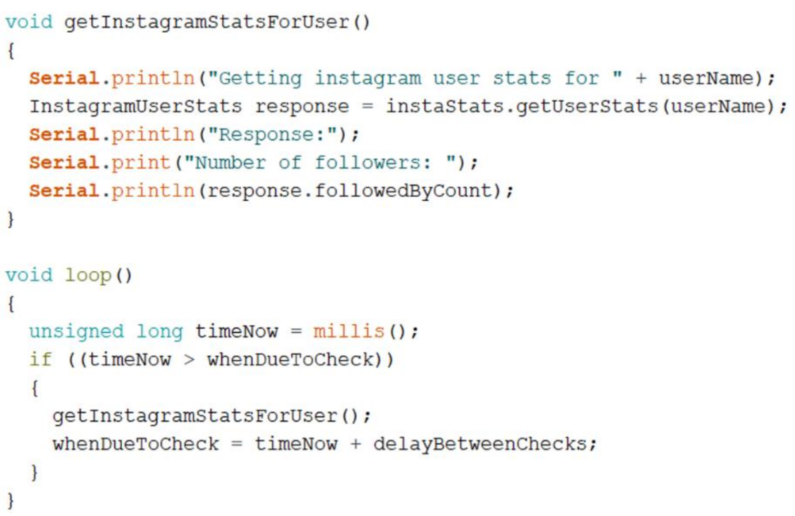
The Serial Print will told us that the card is getting connected to the Wi-Fi using the ssid and the password we give before.

We want to put some dots "." To have something like it's charging if the status is not connected until it's connected.

Then we want to display the IP addresses in the Serial.

We must deactivate the security to have a wide use, so we have to uncomment the line **//client.setInsecure();** because we are using an ESP version 3. At the beginning we were blocked in this part because we didn't uncomment this line and the code wasn't working.

This part of this code will get the user from Instagram.



```
void getInstagramStatsForUser()
{
  Serial.println("Getting instagram user stats for " + userName);
  InstagramUserStats response = instaStats.getUserStats(userName);
  Serial.println("Response:");
  Serial.print("Number of followers: ");
  Serial.println(response.followedByCount);
}

void loop()
{
  unsigned long timeNow = millis();
  if ((timeNow > whenDueToCheck))
  {
    getInstagramStatsForUser();
    whenDueToCheck = timeNow + delayBetweenChecks;
  }
}
```

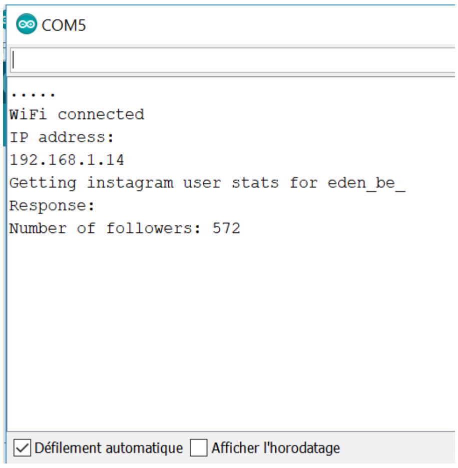
The username is the name of the variable named by the Instagram account.

Response will be the link of the account and we will get the number of followers (function part of the Instagram library). The function (followedByCount) count the number of followers of the account (response).

At the end, we create a function

Millis corresponds to the current time. This time is the variable **timeNow**. Then in the for loop, we ask the program to wait until **the timeNow** is bigger than 0 (because we initialize **WhenDueToCheck** on 0). When it's true, we want the data of Instagram, and after the 0 becomes the **timeNow** added to the **delayBetweenCheks**; we put 1 minute (60000ms). So, if the actual time is bigger than the last time + 60 seconds, we will wait. This provides economy in asking requests. It's better than asking every second the data from Instagram, it will take a lot of energy.

We get this:



COM5

```
.....
WiFi connected
IP address:
192.168.1.14
Getting instagram user stats for eden_be_
Response:
Number of followers: 572
```

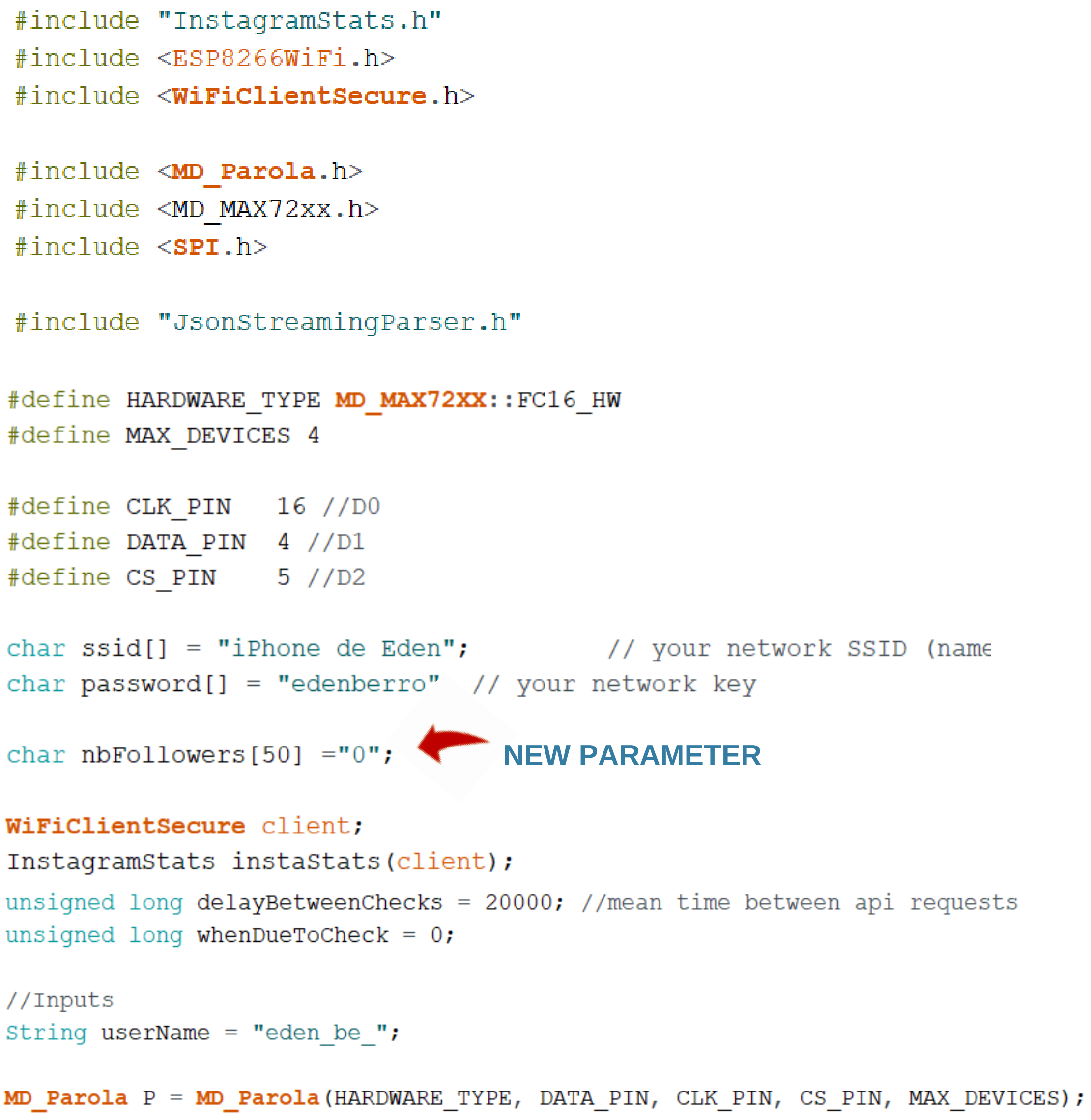
☑ Défilement automatique □ Afficher l'horodatage

DISPLAY THE NUMBER OF FOLLOWERS ON THE SCREEN

We have joined the two codes together with some changes to the display.

Of course, we will find all the libraries of each code.

Following the libraries, we will find all the parameters of the codes and in addition, an array that can contain 50 characters that is initialized to 0. Other parameters such as the display can be modified according to the cabling, or the network can change in relation to the available network.

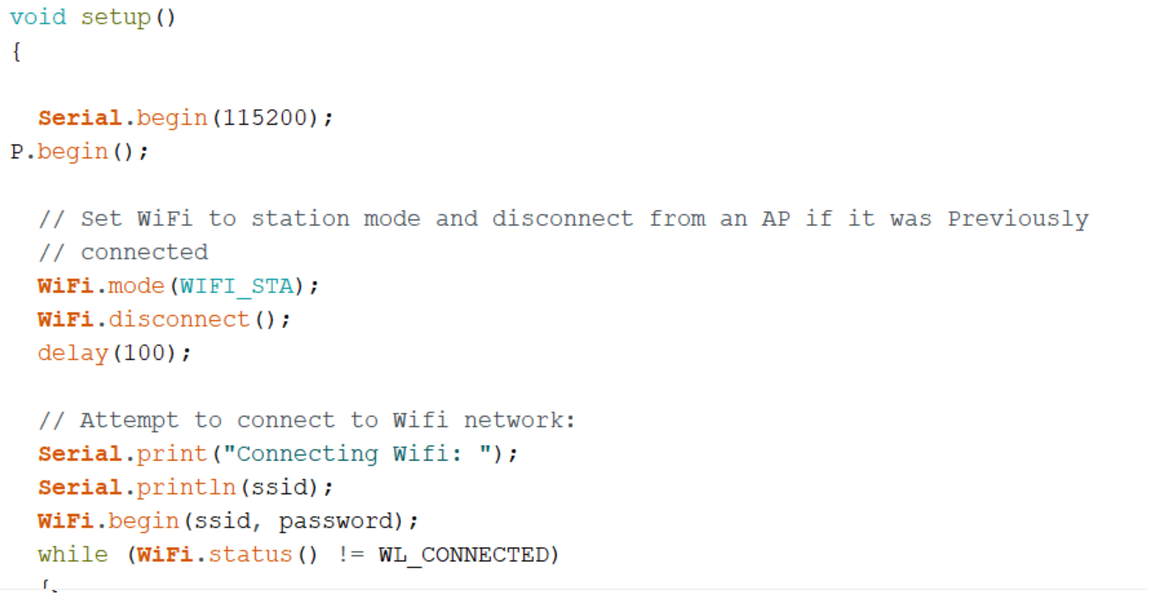


```
#include "InstagramStats.h"
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>

#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>

#include "JsonStreamingParser.h"

#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
#define MAX_DEVICES 4

#define CLK_PIN   16 //D0
#define DATA_PIN  4 //D1
#define CS_PIN    5 //D2

char ssid[] = "iPhone de Eden";          // your network SSID (name
char password[] = "edenberro"  // your network key

char nbFollowers[50] ="0";

WiFiClientSecure client;
InstagramStats instaStats(client);
unsigned long delayBetweenChecks = 20000; //mean time between api requests
unsigned long whenDueToCheck = 0;

//Inputs
String userName = "eden_be_";

MD_Parola P = MD_Parola(HARDWARE_TYPE, DATA_PIN, CLK_PIN, CS_PIN, MAX_DEVICES);
```

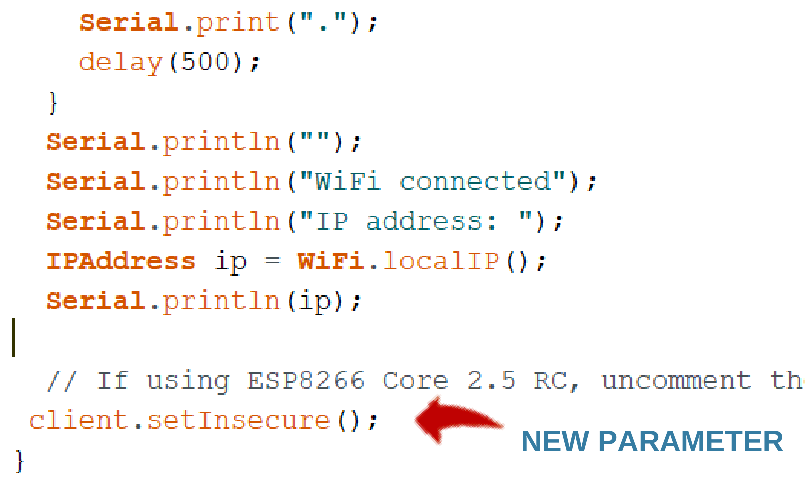
NEW PARAMETER

The Setup function is simply used to connect to the network selected in the parameters. The process flow of the function is identical to the initial indicator.

However, the last instruction of this function allows you to remove the network security and thus disable the controls.

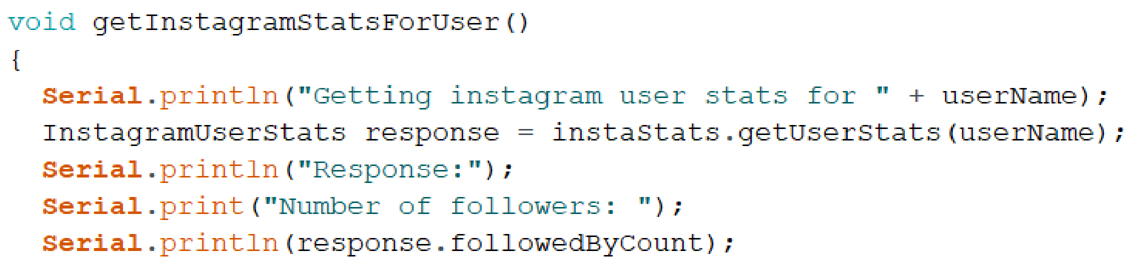


```
void setup()
{

  Serial.begin(115200);
P.begin();

  // Set WiFi to station mode and disconnect from an AP if it was Previously
  // connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  // Attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
```

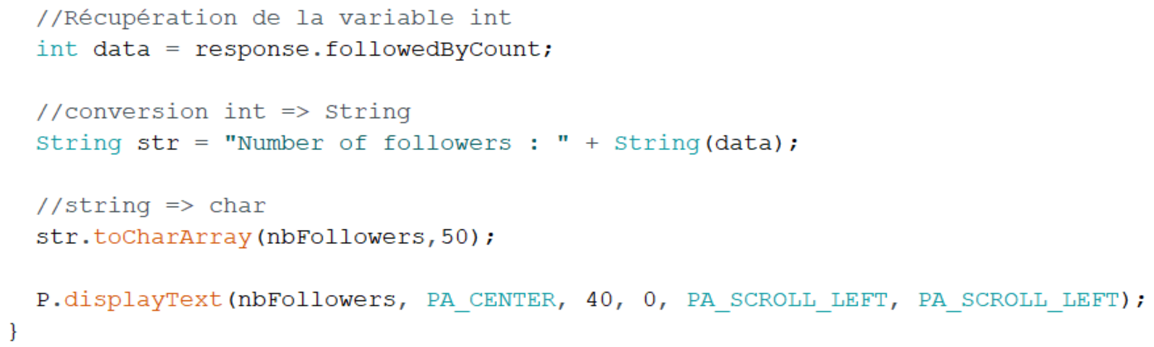


```
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  IPAddress ip = WiFi.localIP();
  Serial.println(ip);

  // If using ESP8266 Core 2.5 RC, uncomment the following
 client.setInsecure();
}
```

NEW PARAMETER



```
void getInstagramStatsForUser()
{
  Serial.println("Getting instagram user stats for " + userName);
  InstagramUserStats response = instaStats.getUserStats(userName);
  Serial.println("Response:");
  Serial.print("Number of followers: ");
  Serial.println(response.followedByCount);
```



```
  //Récupération de la variable int
  int data = response.followedByCount;

  //conversion int => String
  String str = "Number of followers : " + String(data);

  //string => char
  str.toCharArray(nbFollowers,50);

  P.displayText(nbFollowers, PA_CENTER, 40, 0, PA_SCROLL_LEFT, PA_SCROLL_LEFT);
}
```

```
void loop()
{
  if( P.displayAnimate())
  {
    P.displayReset();
    }
  unsigned long timeNow = millis();
  if ((timeNow > whenDueToCheck))
  {
    getInstagramStatsForUser();
    whenDueToCheck = timeNow + delayBetweenChecks;
  }
}
```

When recovering data on Instagram, the steps are identical. But to display it on the led matrix it is first necessary store, the data in an "int" (the recovered data is an integer). And then convert it to a string character and finally copy this string into the array declared in the top of this code (parameters part) in order to display it on the matrix module.

For the loop function, there is no change. It contains all the instructions of both codes combined in the same function.

CONCLUSION

This project allowed us to handle with different components. learn how to connect to a Wi-Fi network. Discover new functions such as retrieving values from the internet. We encountered a lot of difficulties on this project which was originally a YouTube subscriber display and not Instagram. But to get an API key for YouTube is quite long because they do a check then they accept or not the API request. For Instagram thanks to the library we can freely recover data from the site. We had difficulties to find the good libraries and to succeed displaying data on the Matrix.

The main function was respected, so we thought about what could be improved.

For example:

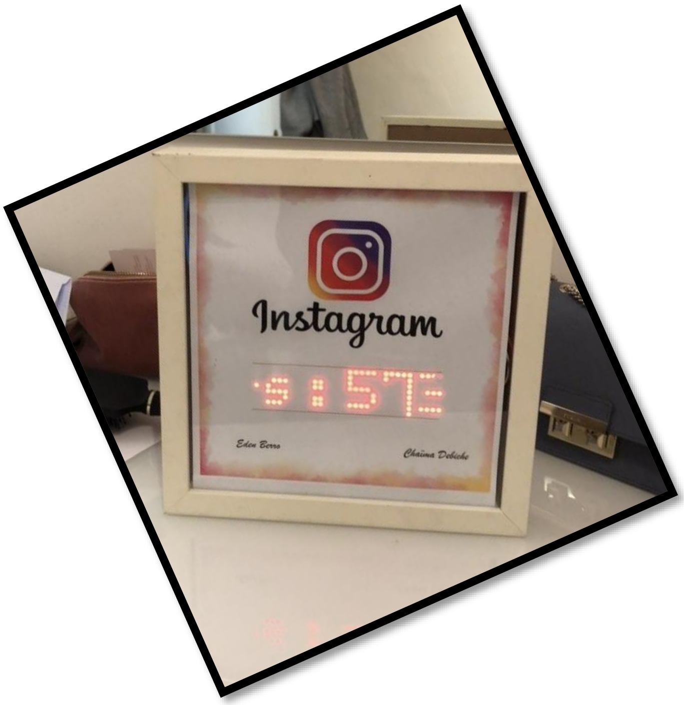
Add instructions when connecting to the Wi-Fi network

- To identify if the password is incorrect.
- To announce that there is no network found.

We put the price that deserved our project after the Part “Sources” at the end of the report.

FINAL RESULT

In class, we made a presentation of the project. And to see if the project worked well without being connected to the PC, we asked the class to follow the account presented in class. And it works! The number of followers increase after 60sec when people were following the account. We create a frame to have a beautiful prototype and to complete the goal of this project. It can be useful for influent person on social media or for compagnies who pay attention to their mark. Also, it can increase the number of follower and it gives more popularity to the company because people will try to follow the company account to see the difference on the board.



SOURCES

Frame:

https://www.youtube.com/watch?v=3ytCxp5J7QM

How to connect to YouTube's API:
https://developers.google.com/youtube/v3/gettingstarted?fbclid=IwAR0xeYR52TessL9NDXOcTJeJ0juf1TMsbbJqD4pG18EbQtwSqy_034pJvk8

Liquid Crystal Library:

https://www.arduino.cc/en/Reference/LiquidCrystal

In the first TD of Arduino we see how to connect the Arduino by WIFI so we will use it:
https://blogs.msdn.microsoft.com/jdupuy/2015/01/22/zone61-guide-pour-configurer-votre-cartearduino-yn/

Powering an Arduino without USB:

https://zestedesavoir.com/tutoriels/364/alimenter-une-arduinosans-usb/ +
https://www.locoduino.org/spip.php?article16

Arduino Instagram Stats Library:

https://www.youtube.com/watch?v=ssceP406eYk

Materials:

ESP8266 :
https://www.amazon.fr/gp/product/B06Y1ZPNMS/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1

Matrix7266:

https://www.amazon.fr/gp/product/B07NMDZ89S/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1

PRICE

The material itself cost 15€ and to cost the work and the time past on this project it will cost about 50€.