🏆 A4.1 Learning activity

Control circuit to activate and deactivate a DC motor, using NodeMCU ESP32 via Bluetooth

Instructions

- Realize an assembled control system by means of **Bluetooth**, capable of controlling a DC motor, using a NodeMCU **ESP32**, one and one **IC L293D**.
- Any activity or challenge must be carried out using the **MarkDown style with .md** extension and the VSCode development environment, and must be prepared as a **single page** document, that is, if the document has images, links or Any external document must be accessed from tags and links, and must be named with the nomenclature **A4.1_TituloAactivity_NombreAlumno.pdf.**
- It is a requirement that the .md contains a tag of the link to the repository of your document in GITHUB, for example **Link to my GitHub** and at the end of the challenge it should be uploaded to github.
- From the **.md** file, export a **.pdf** file that must be uploaded to classroom within its corresponding section, serving as evidence of its delivery, since being the **official** platform, the rating of your activity.
- Considering that the .PDF file, which was obtained from the .MD file, both must be identical.
- Your repository in addition to having a **readme**.md file in your root directory, with information such as student data, work team, subject, career, advisor data, and even logo or images, it must have a section of contents or index, which really are links or **links to your .md documents** , _ avoid using text_ to indicate internal or external links.
- A structure is proposed as indicated below, however any other that supports you can be used to organize your repository.

```
- readme.md
  - blog
    - C4.1_TituloActividad.md
    - C4.2_TituloActividad.md
    - C4.3_TituloActividad.md
    - C4.4_TituloActividad.md
  - img
  - docs
    - A4.1_TituloActividad.md
    - A4.2_TituloActividad.md
    - A4.3_TituloActividad.md
```

Sources of support to develop the activity

- ☑ Random Nerd Tutorial DHT Humidity and Temperature
- ☑ DC motor with IC L293 and ESP32

Development

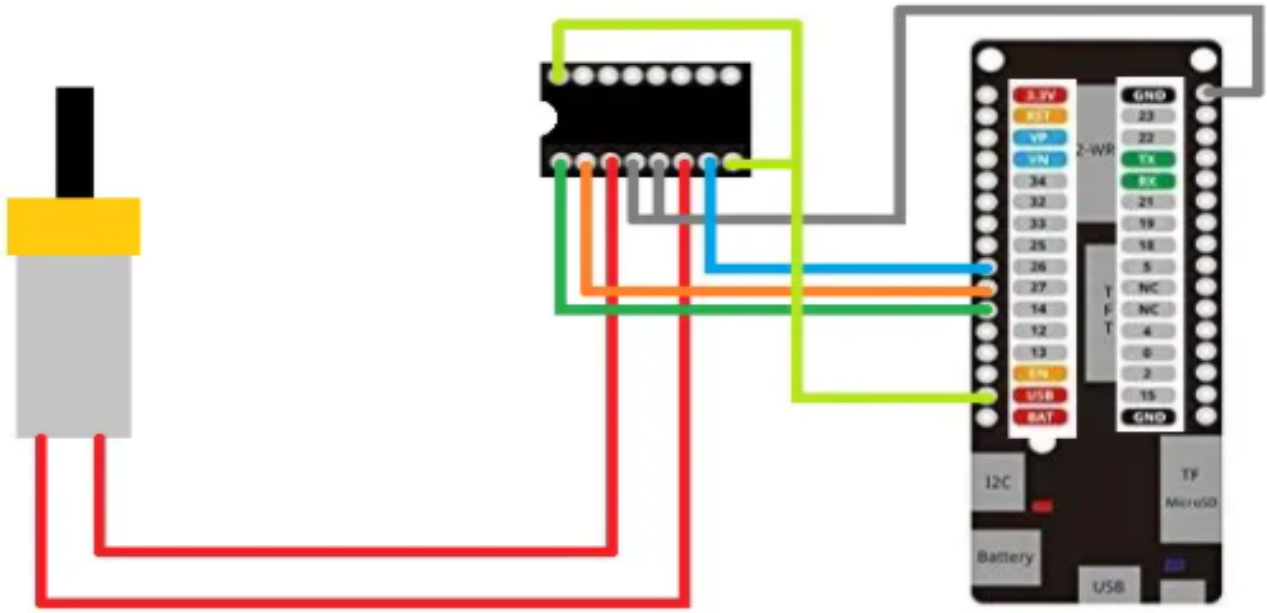
1.Use the following list of materials to prepare the activity

Quantity	Description
1	IC L293D
1	5V voltage source
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M
1	Gear motor

2. Based on the images shown in **Figure 1** , assemble an electronic circuit, in order to obtain a system capable of complying with the following instructions:

- Through the application "Serial Bluetooth terminal" that can be downloaded from the google play Store or even any other that you consider, you must control the start and shutdown of a DC motor, that is, there will be two requests, which one of they will represent the **"START" and the other option "STOP"**
- The motor must be able to rotate clockwise for 5 seconds, at the end of that time it must brake for 1 second and reverse its rotation for another 5 seconds, that is, the activity must have the following sequence: The **stop** can be executed at any time, and the engine will be executing 5s forward, 1s stop, 5s reverse, 1s stop, 5s forward, 1s stop, 5s reverse, ...

Figure 1 ESP32 IC L293 DC Motor Circuit

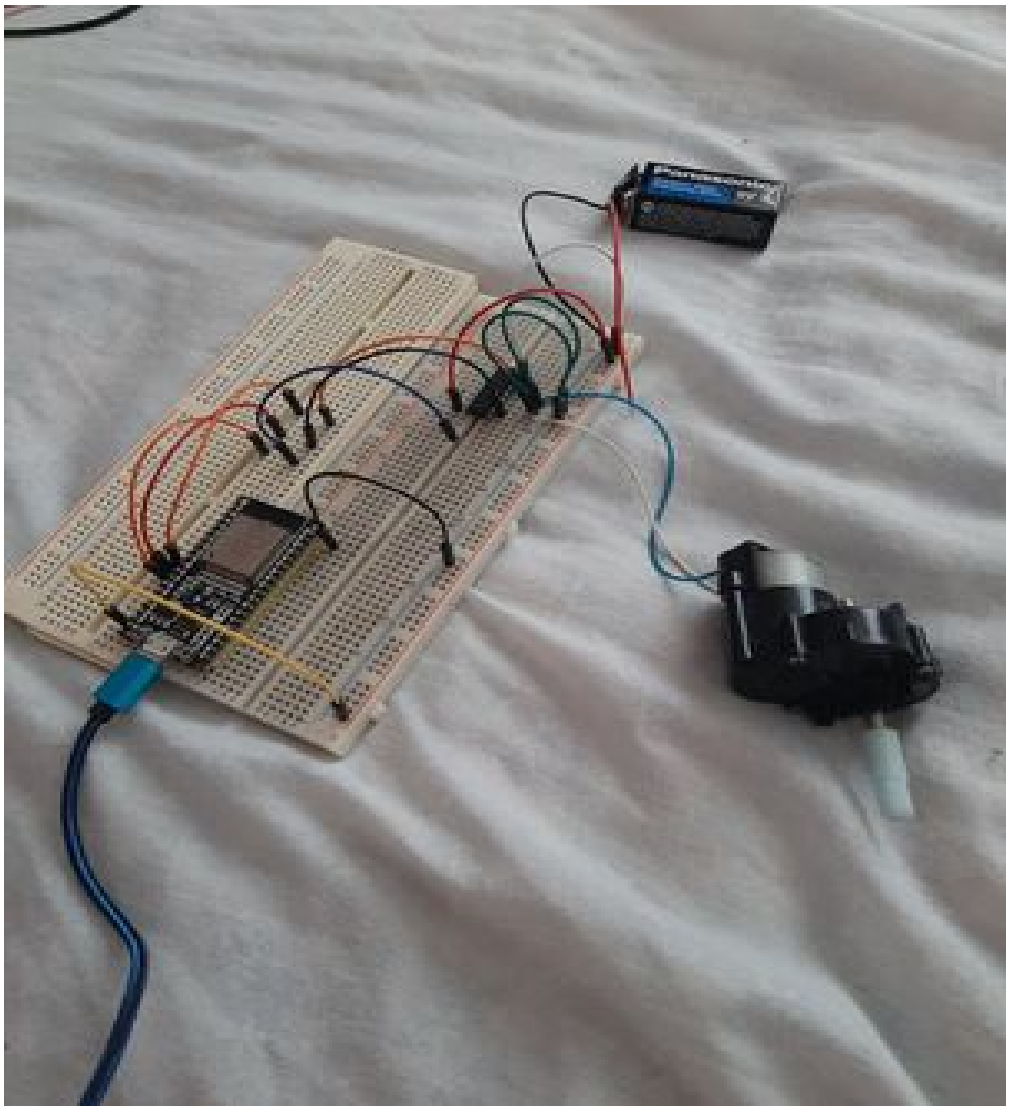


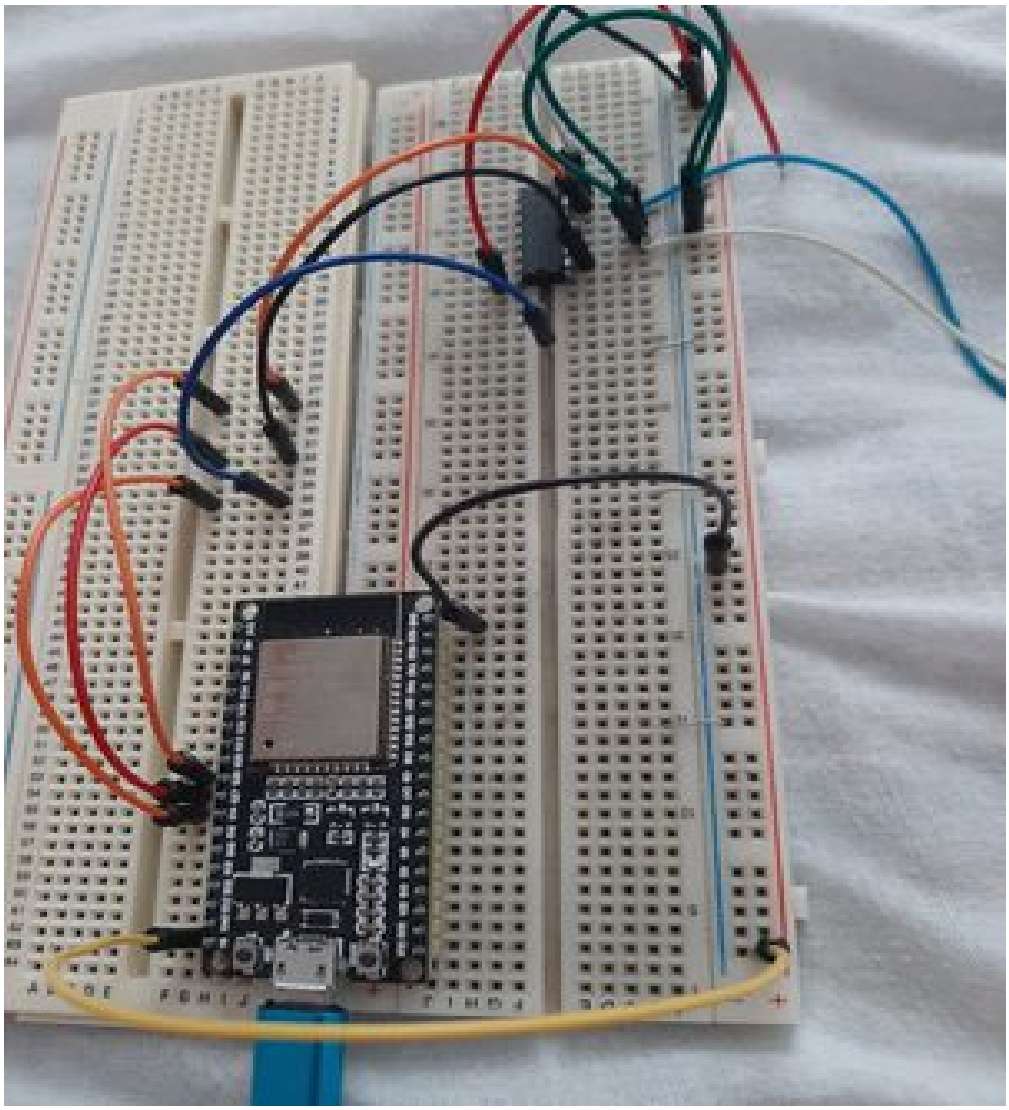
3. Place the picture of the assembled circuit here
4. Place the program created within the Arduino environment in this place

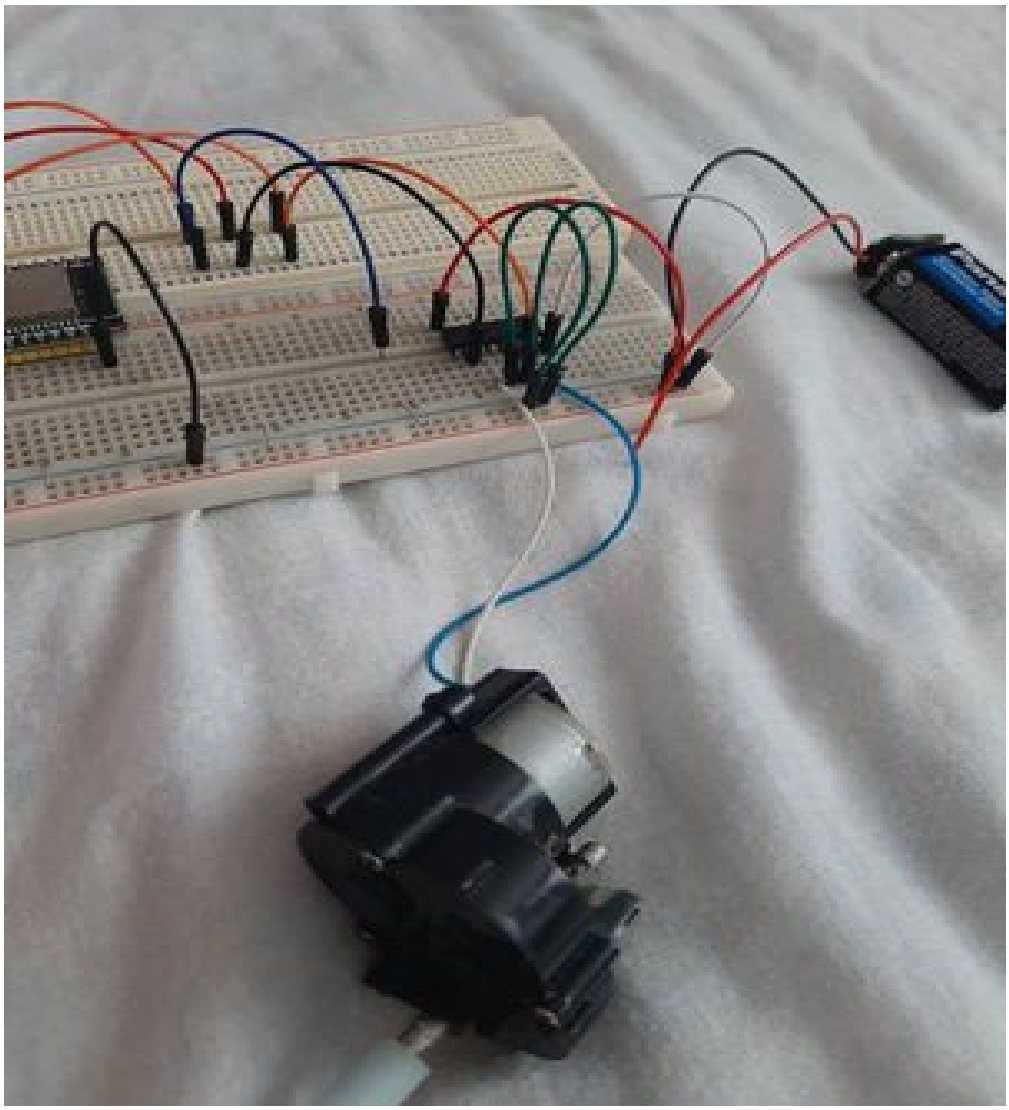
```
int motor1Pin1 = 27; //in3
int motor1Pin2 = 26; //in4
int enable1Pin = 14; //in2

bool I;
bool P;
bool Giro = false;
char Ent;


#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
  // put your setup code here, to run once:
   pinMode(motor1Pin1, OUTPUT);
   pinMode(motor1Pin2, OUTPUT);
   pinMode(enable1Pin, OUTPUT);


   Serial.begin(9600);

   // testing
   Serial.print("Testing DC Motor...");
   SerialBT.begin("ESP32test"); //Bluetooth device name
   Serial.println("The device started, now you can pair it with bluetooth!");


}

void loop() {

     Sigue:
     if (SerialBT.available()) {
      Ent = SerialBT.read();
      Serial.println(Ent);
     }

      if(Ent=='I')
      {
        I=true;
        Giro=false;
      }

      if(Ent=='P')
      {
        I=false;
      }
```

```
      if(I==true)
      {
        if(Giro)
        {
           digitalWrite(motor1Pin1, HIGH);
           digitalWrite(motor1Pin2, LOW);
           Giro = false;
        }
        else
        {
          digitalWrite(motor1Pin1, LOW);
          digitalWrite(motor1Pin2, HIGH);

          Giro = true;
        }
        for(int i =0; i<5000;i++)
        {
           delay(1);
           if (SerialBT.available())
           {
              Ent = SerialBT.read();
              Serial.println(Ent);

              if(Ent=='P')
              {
                I=false;
                digitalWrite(motor1Pin1, LOW);
                digitalWrite(motor1Pin2, LOW);
                goto Sigue;
              }

           }
         }

      }

       digitalWrite(motor1Pin1, LOW);
       digitalWrite(motor1Pin2, LOW);
       delay(1000);
}
```

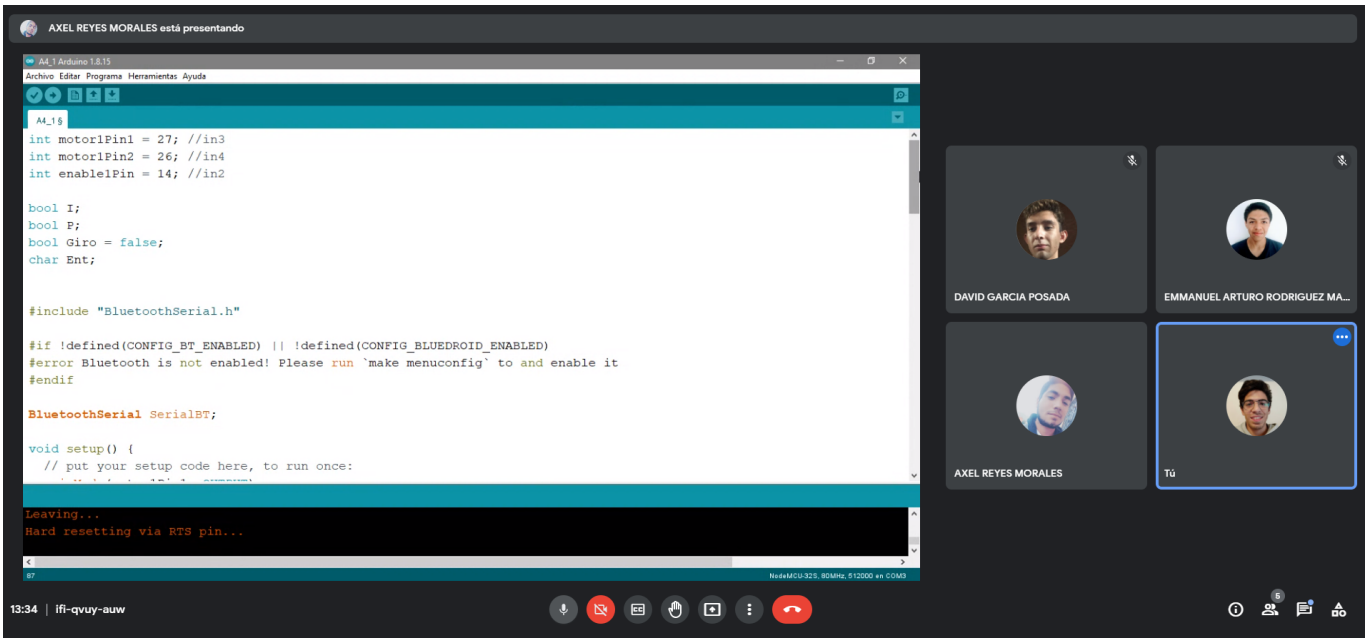
5. Place here evidence that you consider important during the development of the activity.

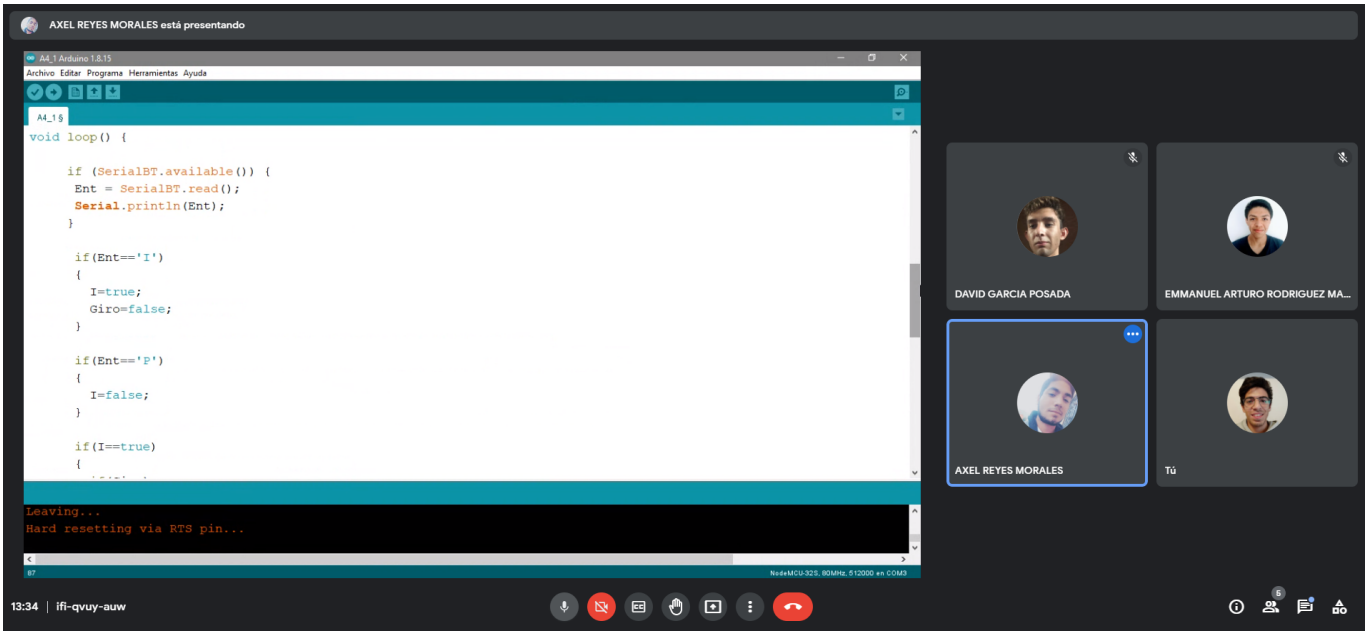


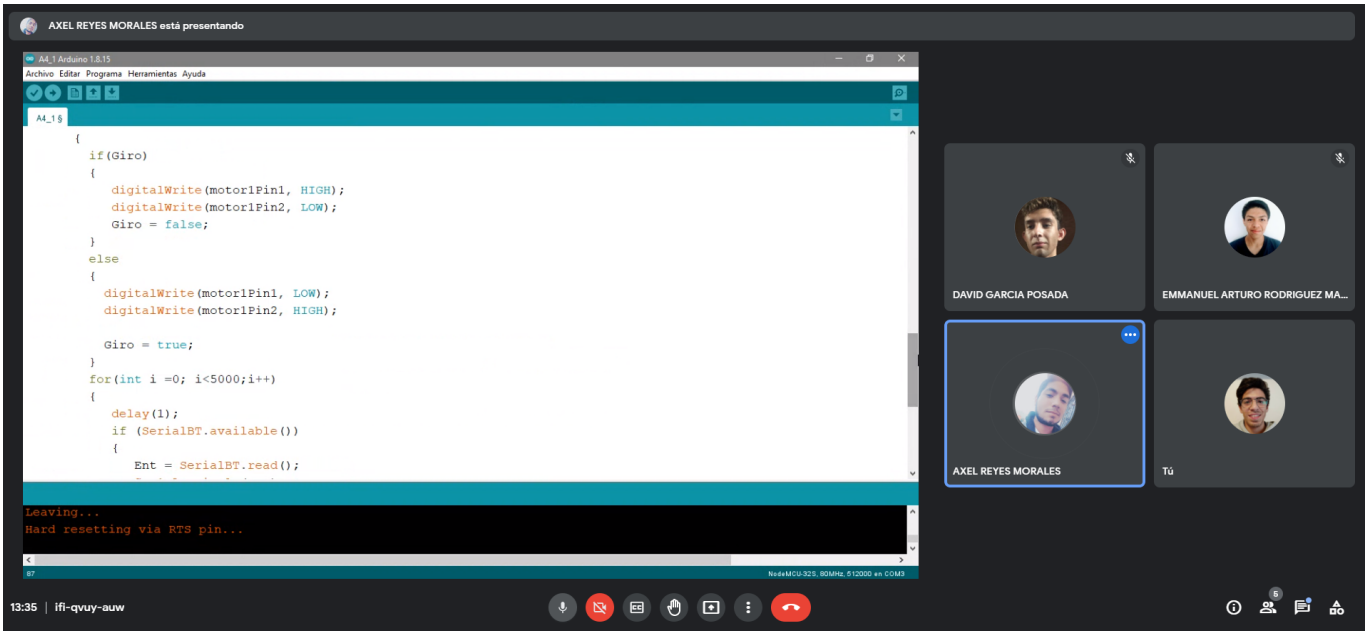


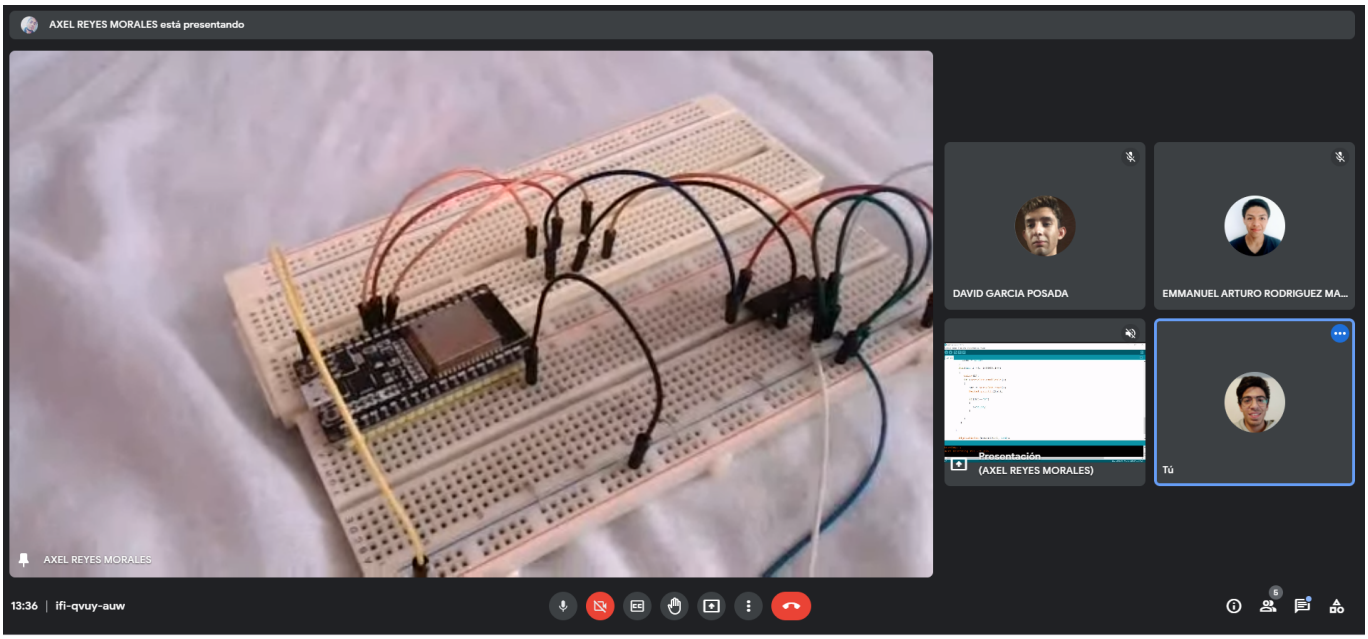


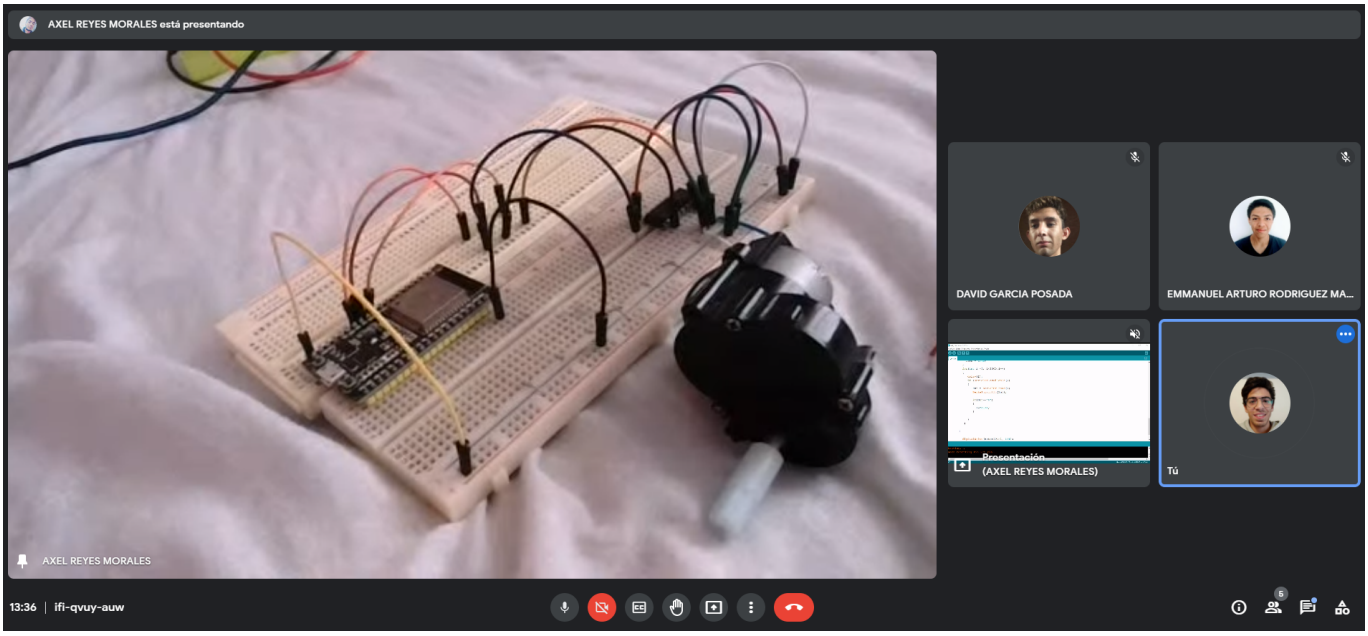
6. Insert images of **evidence** such as meetings of the team members held for the development of the activity


AXEL REYES MORALES está presentando
int motor1Pin1 = 27; //in3
int motor1Pin2 = 26; //in4
int enable1Pin = 14; //in2
bool I;
bool P;
bool Giro = false;
char Ent;
#include "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif
BluetoothSerial SerialBT;
void setup() {
// put your setup code here, to run once:
Leaving...
Hard resetting via RTS pin...
DAVID GARCIA POSADA
EMMANUEL ARTURO RODRIGUEZ MA...
AXEL REYES MORALES
Tú
13:34 | ifi-qvuy-auw


AXEL REYES MORALES está presentando
void loop() {
if (SerialBT.available()) {
Ent = SerialBT.read();
Serial.println(Ent);
}
if(Ent=='I')
{
I=true;
Giro=false;
}
if(Ent=='P')
{
I=false;
}
if(I==true)
{
Leaving...
Hard resetting via RTS pin...
DAVID GARCIA POSADA
EMMANUEL ARTURO RODRIGUEZ MA...
AXEL REYES MORALES
Tú
13:34 | ifi-qvuy-auw


AXEL REYES MORALES está presentando
{
if(Giro)
{
digitalWrite(motor1Pin1, HIGH);
digitalWrite(motor1Pin2, LOW);
Giro = false;
}
else
{
digitalWrite(motor1Pin1, LOW);
digitalWrite(motor1Pin2, HIGH);
Giro = true;
}
for(int i =0; i<5000;i++)
{
delay(1);
if (SerialBT.available())
{
Ent = SerialBT.read();
Leaving...
Hard resetting via RTS pin...
DAVID GARCIA POSADA
EMMANUEL ARTURO RODRIGUEZ MA...
AXEL REYES MORALES
Tú
13:35 | ifi-qvuy-auw





1. Conclusions

Axel: To conclude I can say that the use of bluetooth with the Esp32 microcontroller was somewhat complicated to understand at first, but the operation was understood for the reading of data to be able to use it within the program and to be able to perform the sequence of the engine, making use of the L293D, to make it turn clockwise and also counterclockwise. It was also possible to make it possible to stop the engine at any time during the execution of the program, with this you can carry out some other practices and make variations of it to know a little more about the operation of these tools.

David: During this activity we were able to observe the process of connection via bluetooth with the NodeMCU, with this we were able to control a gear mottor and change the time it was active as well as the direction, also although with some dificulty we were able to make it stop when we send a comand via bluetooth. to make this work we encounter some dificulties like the inability to stop the program at any time and some dificulties to make the program know it had to stop the mottor but with some reserch and time we were able to find a way to make posible what the activity ask for

Emmanuel: In conclusion, a circuit was made with the ESP32 microcontroller with bluetooth functionality and the L293D controller so that a motor would perform the action of turning according to the hands of the clock

for 5 seconds and stop for 1 second, and then turn to the side. contrary for another 5 seconds and then stop again and change the course, in addition to that it had to stop the execution at any time regardless of how many seconds it took to stop, this was the complicated thing but it was possible to do it with a variable that would determine when it should be the engine running and when it should stop.

Oscar Rojas: With the esp32 microcontroller we can take advantage of the bluetooth functionalities to do in this case and with the use of the L293D controller to perform a start and stop action of a motor, in this case the motor turn to one direction for 5 seconds and stops for 1 second, and when this delay reach its end the motor start turning to the other side, this is a practice to understand how to control a motor and make it turn to both directions just with the implementation of a L293D microcontroller.

Repositories

Axel Go to my repository

David Go to my repository

Emmanuel Go to my repository

Oscar Rojas Got to my repository