



A4.1 Actividad de aprendizaje

Circuito de control para activar y desactivar un motor DC, utilizando NodeMCU ESP32 por medio de Bluetooth

Instrucciones

- Realizar un sistema ensamblado de control por medio de **Bluetooth**, capaz de control a un motor DC, utilizando un NodeMCU **ESP32**, un y un **IC L293D**.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A4.1_NombreApellido_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo **.PDF**, el cual fue obtenido desde archivo **.MD**, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
- 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
```

Fuentes de apoyo para desarrollar la actividad

- [Random Nerd Tutorial DHT Humedad y temperatura](#)
- [Motor DC con IC L293 y ESP32](#)

Desarrollo

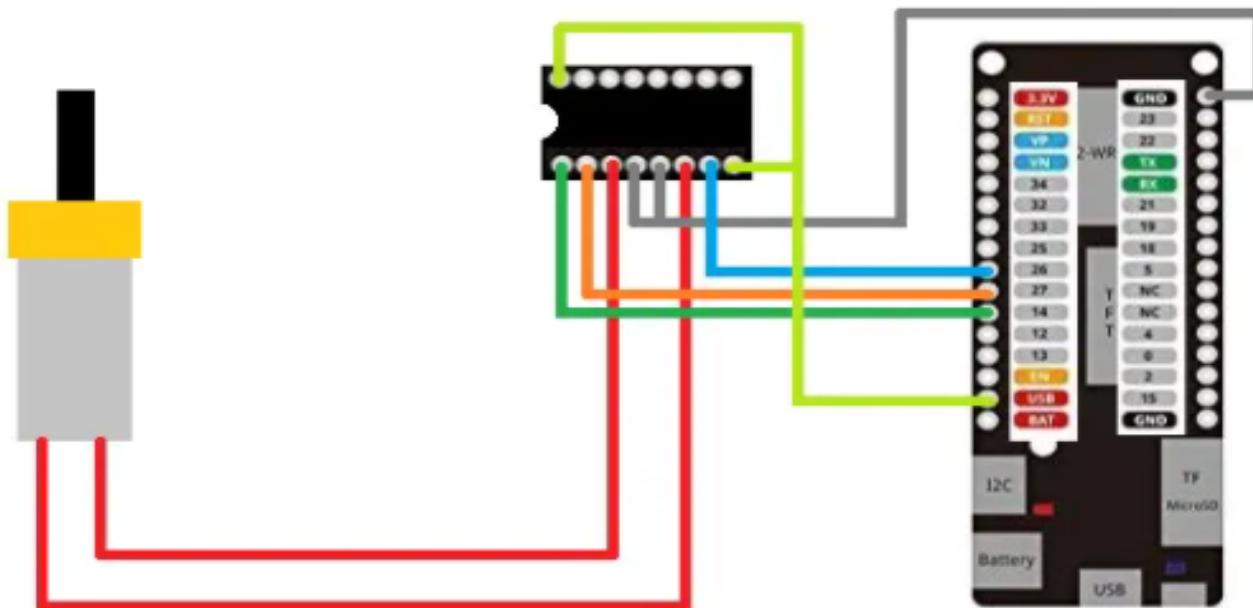
1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

Cantidad	Descripción
1	IC L293D
1	Fuente de voltaje de 5V
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M

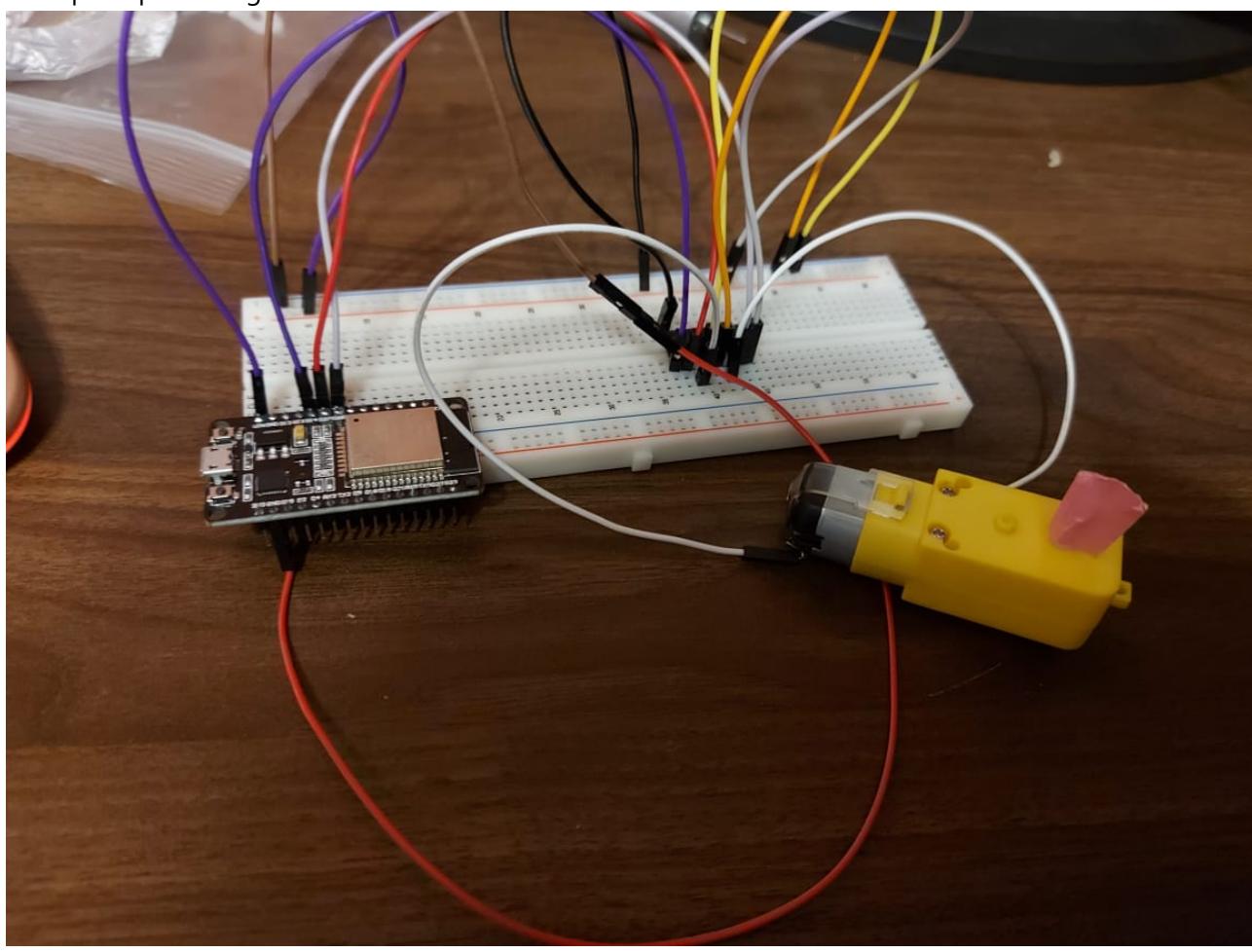
2. Basado en las imágenes que se muestran en las **Figura 1**, ensamblar un circuito electrónico, con la finalidad de obtener un sistema capaz de cumplir con las instrucciones siguientes:

- Por medio de la aplicación "Serial Bluetooth terminal" que puede ser descargada del play Store de google o incluso cualquier otra que considere, se deberá controlar el arranque y apagado de un motor DC, es decir se contara con dos peticiones, la cual una de ellas representara el "START" y la otra opción "STOP"
 - El motor debe ser capaz de girar a favor de las manecillas del reloj durante 5 segundos, al cumplirse ese tiempo debe frenar 1 segundo e invertirá su giro durante otros 5 segundos.

Figura 1 Circuito ESP32 IC L293 Motor DC



3. Coloque aqui la imagen del circuito ensamblado



4. Coloque en este lugar el programa creado dentro del entorno de Arduino

```
#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;
int received;// valor se guardara en esta variable
char receivedChar;// el valor recibido CHAR se guarda en esta variable

const char turnON ='o'; //variable para encender
const char turnOFF ='f';//Variable para apagar

const int motor1 = 27; //pines del motor
const int motor2 = 26;
const int enable1Pin = 14;//pin para habilitar el L239D
bool motor = false; //bandera para utilizar la secuencia
int Tiempo=0; // para medir el tiempo de activacion despues de que enciende el
motor

const int freq = 30000;
const int pwmChannel = 0;
```

```
const int resolution = 8;
int dutyCycle = 200;

void setup() {

    SerialBT.begin("ESP32"); //nombre del dispositivo BT
    Serial.println("El dispositivo se emparejo!");
    pinMode(enable1Pin, OUTPUT); // ponemos los pines del motor y del L239D como
salida
    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
    //configuracion del pwm del motor
    ledcSetup(pwmChannel, freq, resolution);
    ledcAttachPin(enable1Pin, pwmChannel);
    Serial.begin(115200);//iniciar consola
}

void loop() {
    receivedChar =(char)SerialBT.read();// el valor de la variable es igual al
valor recibido en la consola BT

    if (Serial.available()) { //si la consola esta disponible el BT puede escribir y
la consola lo puede leer
        SerialBT.write(Serial.read());
    }

    if (SerialBT.available()) {
        SerialBT.print("Recibido:");// Escribir en BT app
        SerialBT.println(receivedChar);// Escribir en BT app
        Serial.print ("Recibido:");//imprimir en el monitor serial
        Serial.println(receivedChar);//imprimir en el monitor serial

        if(receivedChar == turnON)iniciar(); // si el caracter que se envia es igual a
la variable "turnON" iniciar la secuencia del motor
        else if (receivedChar == turnOFF) Apagar(); // si el caracter que se envia es
igual a la variablw "turnOFF" detener la secuencia del motor
    }
    if (motor) Secuencia();
}

//Iniciar la secuencia
void iniciar(){
    Tiempo = millis();
    motor=true;
}

//detener la secuencia
void Apagar(){
    Detener();
    motor=False;
}

//secuencia del motor
```

```
void Secuencia(){
    int tiempoTranscurrido= millis() - Tiempo; //El tiempo transcurrido es igual a
    la del tiempo de activacion del motor
    if(tiempoTranscurrido < 5000) Encendido(); // Encender el motor durante 5
    segundos
    else if(tiempoTranscurrido < 6000) Detener(); // Apagar el motor despues de los
    5 segundos
    else if(tiempoTranscurrido < 11000) Reversa(); // Encender el motor en reversa
    otros 5 segundos
    else if(tiempoTranscurrido < 12000) Detener(); // Detener el motor
    else iniciar(); //Reiniciar la secuencia
}

void Encendido(){
dutyCycle = 255;
ledcWrite(pwmChannel, dutyCycle);
Serial.println("Motor hacia delante");
digitalWrite(motor1, LOW);
digitalWrite(motor2, HIGH);
}

// para mover motor hacia atras
void Reversa(){
Serial.println("El motor se mueve hacia atras");
digitalWrite(motor1, HIGH);
digitalWrite(motor2, LOW);
}

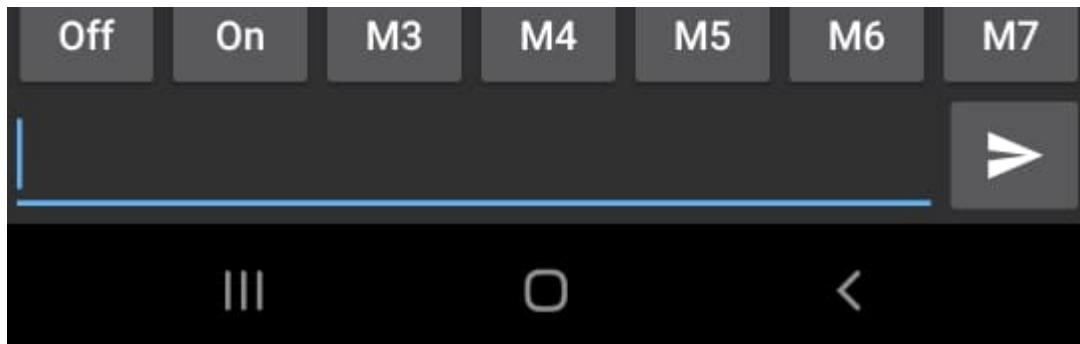
void Detener(){
Serial.println("el motor se detuvo");
digitalWrite(motor1, LOW);
digitalWrite(motor2, LOW);
}
```

5. Coloque aquí evidencias que considere importantes durante el desarrollo de la actividad.

- [Video Demostrativo](#)

The screenshot shows a mobile application interface for a terminal or serial port. At the top, there is a blue header bar with the time "8:28" and several icons for signal strength, battery, and connectivity. Below the header, the word "Terminal" is displayed in white text on a dark background. To the right of "Terminal" are three small white icons: a left arrow, a right arrow, a trash can, and a vertical ellipsis. The main area of the screen is a dark gray scrollable text view. The text contains numerous lines of log output from a device, primarily in green and white colors. The log includes messages such as "Connecting to ESP32 ...", "Connected", and many instances of "Recibido:" followed by various characters like '^M' and '^Z'. There are also occasional lines starting with "o" and "f". The text is continuous and scrolls vertically.

```
20:24:37.035 Connecting to ESP32 ...
20:24:38.717 Connected
20:24:40.638 o
20:24:40.680 Recibido:♦
20:24:40.695 Recibido:o
20:24:40.695 Recibido:^M
20:24:55.188 f
20:24:55.747 Recibido:f
20:24:55.747 Recibido:^M
20:26:41.379 o
20:26:41.457 Recibido:♦
20:26:41.457 Recibido:o
20:26:41.457 Recibido:^M
20:27:05.818 f
20:27:05.889 Recibido:f
20:27:05.921 Recibido:^M
20:27:08.441 o
20:27:08.493 Recibido:o
20:27:08.518 Recibido:^M
20:27:09.777 f
20:27:09.836 Recibido:f
20:27:09.863 Recibido:^M
20:27:11.048 o
20:27:11.147 Recibido:♦
20:27:11.176 Recibido:o
20:27:11.176 Recibido:^M
20:27:12.347 f
20:27:12.400 Recibido:f
20:27:12.428 Recibido:^M
20:27:13.593 o
20:27:13.652 Recibido:♦
20:27:13.682 Recibido:o
20:27:13.682 Recibido:^M
20:27:15.246 f
20:27:15.310 Recibido:f
20:27:15.343 Recibido:^M
```



COM8

[Enviar](#)

Autoscroll

Nueva línea

115200 baud

[Clear output](#)

COM8

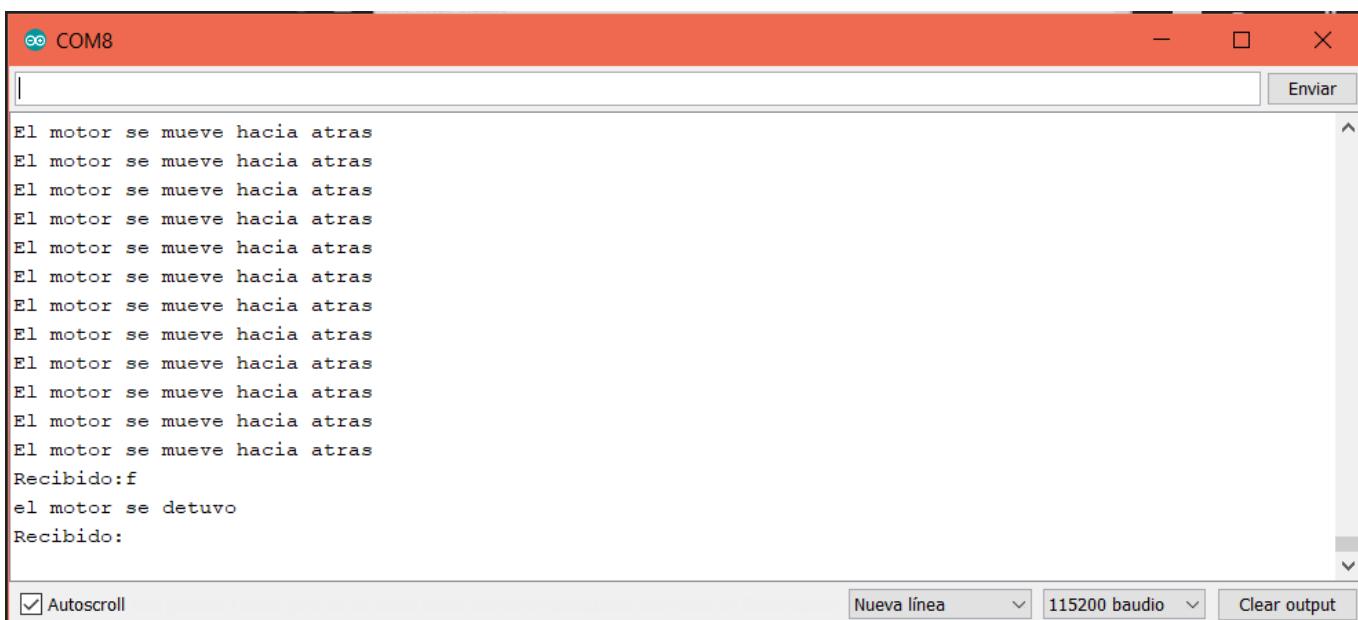
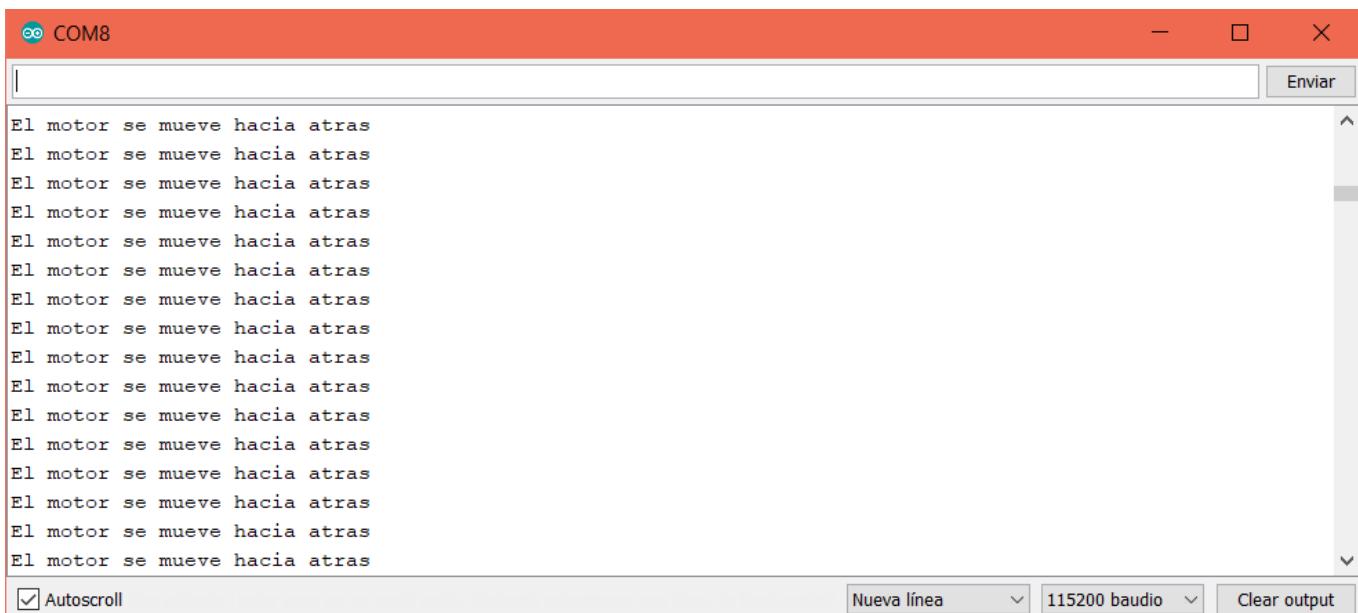
Enviar

Autoscroll

Nueva línea

115200 baud

[Clear output](#)



Slack

Screenshot of a Slack channel named 'zerox' showing a conversation between users. The messages include:

- VANESSA MARLENNE RODRIGUEZ BAEZ: Chicos Ya me salio
- ALEJANDRO DIAZ NAVA: genial
- VANESSA MARLENNE RODRIGUEZ BAEZ: entonces mañana, reunion
- VANESSA MARLENNE RODRIGUEZ BAEZ: Son una de las evidencias
- ALEJANDRO DIAZ NAVA: Ahorita les mando el codigo

The message from Alejandro Diaz Nava includes three attached files, each showing a terminal window with the text "sola atraz". Below the attachments is a photograph of several colorful wires connected to a breadboard or similar electronic component.

Screenshot of a Slack channel named 'zerox' showing a message from Alejandro Diaz Nava containing a C++ code snippet for an ESP32 microcontroller. The code is intended to control two DC motors via an L239D driver module. The code includes comments explaining the setup and configuration of pins and variables.

```

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

BluetoothSerial SerialBT;
int received;// valor se guardara en esta variable
char receivedChar;// el valor recibido CHAR se guarda en esta variable

const char turnON = '0'; //variable para encender
const char turnOFF = '1';//Variable para apagar

const int motor1 = 27; //pines del motor
const int motor2 = 26;
const int enable1Pin = 14;//pin para habilitar el L239D
bool motor = false; //bandera para utilizar la secuencia
int tiempo=0; // para medir el tiempo de activacion despues de que enciende el motor

const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

void setup()
{
    SerialBT.begin("ESP32"); //nombre del dispositivo BT
    Serial.println("El dispositivo se emparejó!");
    pinMode(enable1Pin, OUTPUT); // ponemos los pines del motor y del L239D como salida
    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
    //configuracion del num del motor
}

```

Messenger | Facebook x Slack | zerox | ITN_SistemasProg x (9) WhatsApp x Contenido del canal - YouTube x + No sincronizando

https://app.slack.com/client/T018PEE2NLD/G01AV2KBNQL Buscar en ITN_SistemasProgramables

ITN_SistemasProgram... zerox

@ Menciones y reacciones

⋮ Más

▼ Canales

general

varios

zerox Añadir canales

⋮ Mensajes directos

Slackbot

VANESSA MARLENNE RO... Añadir compañeros de equi...

zerox Añadir un tema

```
else if(tiempoTranscurrido < 6000) Detener(); // Apagar el motor despues de Hoy v 'ndos
else if(tiempoTranscurrido < 11000) Reversa();// Encender el motor en reversa otros 5 segundos
else if(tiempoTranscurrido < 12000) Detener(); // Detener el motor
else iniciar(); //Reiniciar la secuencia
}

void Encendido(){
dutyCycle = 255;
ledcWrite(pwmChannel, dutyCycle);
Serial.println("Motor hacia delante");
digitalWrite(motor1, LOW);
digitalWrite(motor2, HIGH);
}

// para mover motor hacia atras
void Reversa(){
Serial.println("El motor se mueve hacia atras");
digitalWrite(motor1, HIGH);
digitalWrite(motor2, LOW);
}

void Detener(){
Serial.println("el motor se detuvo");
digitalWrite(motor1, LOW);
digitalWrite(motor2, LOW);
}

Si mañana Reunión para explicarles el código y que miren como funciona el programa
```

Enviar mensaje a zerox

B I ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋

Aa @ ☺ 0 >

Reunión

Google | Carlos A. envió un mensaje | Meet - izv-uite-qky | +

AliExpress Facebook Booking.com Google ¡Bienvenido a Facebook! YouTube - Broadcast... Juegos PC Descargas... Cinemex Materias GitHub - vicjar/topi... »

vanessa rodriguez está presentando

A4.1.2 Arduino 1.8.5 Archivo Editar Programa Herramientas Ayuda

```
#include "BluetoothSerial.h"

#ifndef CONFIG_BT_ENABLED || !defined(CONFIG_BLUEROID_ENABLED)
#error Bluetooth is not enabled! Please run make menuconfig to enable it
#endif

BluetoothSerial SerialBT;
int receivedChar; // valor se guarda en esta variable
char receivedChar; // el valor recibido CHAR se guarda en esta variable

const char turnON = "o"; //variable para encender
const char turnOFF = "f"; //variable para apagar

const int motor1 = 27; //pines del motor
const int motor2 = 26;
const int enablePin = 14; //pin para habilitar el L239D
bool motor = false; //bandera para utilizar la secuencia
int Tiempo; // para medir el tiempo de activación después de que enciende el motor

//PINS
const int freq = 30000;
const int pmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

void setup() {
    SerialBT.begin("ESP32"); //nombre del dispositivo BT
    Serial.println("El dispositivo se enciende");
    pinsMode(enablePin, OUTPUT); // ponemos los pines del motor y del L239D como salida
    pinsMode(motor1, OUTPUT);
    pinsMode(motor2, OUTPUT);
    //configuración del pin del motor
    ledcSetup(pmChannel, freq, resolution);
    ledcAttachPin(enablePin, pmChannel);
    ledcAttachPin(motor1, pmChannel);
    ledcAttachPin(motor2, pmChannel);
}

void loop() {
    receiveChar = (char)SerialBT.read(); // el valor de la consola
    if (Serial.available() > 0) { //si la consola esta disponible
        SerialBT.write(Serial.read());
    }

    if (SerialBT.available()) {
        SerialBT.print("Recibido:"); //Escribir en BT app
        SerialBT.println(receivedChar); //Escribir en BT app
        Serial.print("Imprimiendo en el monitor");
        Serial.println(receivedChar); //Imprimir en el monitor
        if (receivedChar == turnON) Iniciar();
        else if (receivedChar == turnOFF) Apagar();
        else if (receivedChar == 's') Secuencia();
        else if (receivedChar == 'r') Recibir();
    }
}

//Iniciar la secuencia
void Iniciar() {
    Tiempo = millis();
    motor = true;
}
```

12:12 12:12 09/12/2020

Google | Carlos A. envió un mensaje | Meet - izv-uite-qky | +

AliExpress Facebook Booking.com Google ¡Bienvenido a Facebook! YouTube - Broadcast... Juegos PC Descargas... Cinemex Materias GitHub - vicjar/topi... »

vanessa rodriguez está presentando

A4.1.2 Arduino 1.8.5 Archivo Editar Programa Herramientas Ayuda

```
void setup() {
    SerialBT.begin("ESP32"); //nombre del dispositivo BT
    Serial.println("El dispositivo se enciende");
    pinsMode(enablePin, OUTPUT); // ponemos los pines del motor y del L239D como salida
    pinsMode(motor1, OUTPUT);
    pinsMode(motor2, OUTPUT);
    //configuración del pin del motor
    ledcSetup(pmChannel, freq, resolution);
    ledcAttachPin(enablePin, pmChannel);
    ledcAttachPin(motor1, pmChannel);
    ledcAttachPin(motor2, pmChannel);
    Serial.begin(115200); //iniciar consola
}

void loop() {
    receiveChar = (char)SerialBT.read(); // el valor de la consola
    if (Serial.available() > 0) { //si la consola esta disponible
        SerialBT.write(Serial.read());
    }

    if (SerialBT.available()) {
        SerialBT.print("Recibido:"); //Escribir en BT app
        SerialBT.println(receivedChar); //Escribir en BT app
        Serial.print("Imprimiendo en el monitor");
        Serial.println(receivedChar); //Imprimir en el monitor
        if (receivedChar == turnON) Iniciar();
        else if (receivedChar == turnOFF) Apagar();
        else if (receivedChar == 's') Secuencia();
        else if (receivedChar == 'r') Recibir();
    }
}

//Iniciar la secuencia
void Iniciar() {
    Tiempo = millis();
    motor = true;
}

//Apagar el motor
void Apagar() {
    motor = false;
    Serial.println("El motor se detuvo");
}

//Secuencia
void Secuencia() {
    if (motor) {
        if (millis() - Tiempo > 1000) {
            motor = false;
            Serial.println("El motor se detuvo");
        }
    } else {
        if (millis() - Tiempo > 1000) {
            motor = true;
            Serial.println("El motor se encendió");
        }
    }
}

//Recibir
void Recibir() {
    if (motor) {
        if (millis() - Tiempo > 1000) {
            motor = false;
            Serial.println("El motor se detuvo");
        }
    } else {
        if (millis() - Tiempo > 1000) {
            motor = true;
            Serial.println("El motor se encendió");
        }
    }
}
```

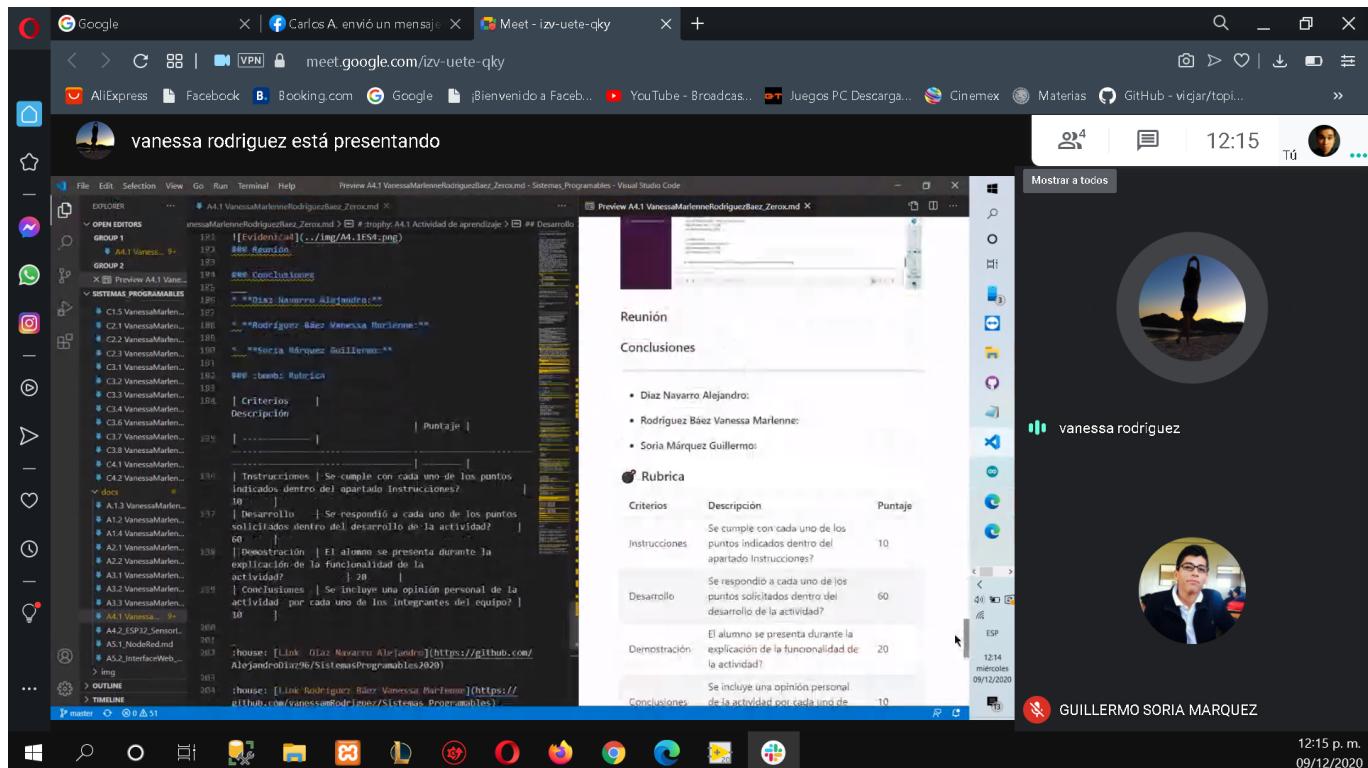
12:13 12:13 09/12/2020

A.4.1 Circuito de control para activar y desactivar un motor DC, utilizando NodeMCU ESP32

3 visualizaciones • 8 dic 2020

SUSCRIBIRSE

12:14 p.m.
09/12/2020



Conclusiones

- Díaz Navarro Alejandro:** Con esta práctica se logró comprender el manejo del motor con la incorporación y utilización de bluetooth, con apoyo de las prácticas anteriores se manejo la implementación de la señal pwm para generar el movimiento del motor, el cambio de sentido o la detención del mismo, entre las complicaciones de esta práctica fue realizar la programación y la conexión con bluetooth ya que al querer implementar la conexión con estos dos el motor dejaba de funcionar y no se lograba solucionar.
- Rodríguez Báez Vanessa Marlenne:** En esta practica se realizo algo similar con lo de la practica pasada de controlar algo por bluetooth, esta practica fue controlar un motor DC con un puente h o un L293D que funciona para poder controlar motores, yo había utilizado lo que es el puente h pero en placa por que utilice 2 motores DC, la dificultad de esta practica fue a la hora de hacer el código ya que nos basamos del materia de ayuda que venia en la practica pero al momento de incluir el bluetooth no funcionaba el motor lo cual eso fue lo mas complicado de esta practica, después de ver un poco los programas de nuestros compañeros pudimos resolver nuestro error.
- Soria Márquez Guillermo:** En esta práctica volvimos a utilizar el ESP32 con la implementación de un Motor DC, esta práctica en particular fue la más complicada que se no ha presentado ya que tuvimos mucho problemas con la codificación y sobre todo con la conexión con el bluetooth. En nuestro caso desarrollamos un código pero al momento de conectarlo con el bluetooth no encendía el motor por error de código. Después de investigar por nuestra parte no lo resolvimos en el momento ya que todavía no sabíamos porque ocurría el error de código. Tardamos un poco más de lo normal en terminar esta práctica pero al final logramos resolver los errores que teníamos.

Rubrica

Criterios	Descripción	Puntaje
-----------	-------------	---------

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10

 [Link Díaz Navarro Alejandro](#)

 [Link Rodríguez Báez Vanessa Marlenne](#)

 [Link Soria Márquez Guillermo](#)