Control de un robot movil con el μ L ESP32





tallerrobotica.fi@gmail.com









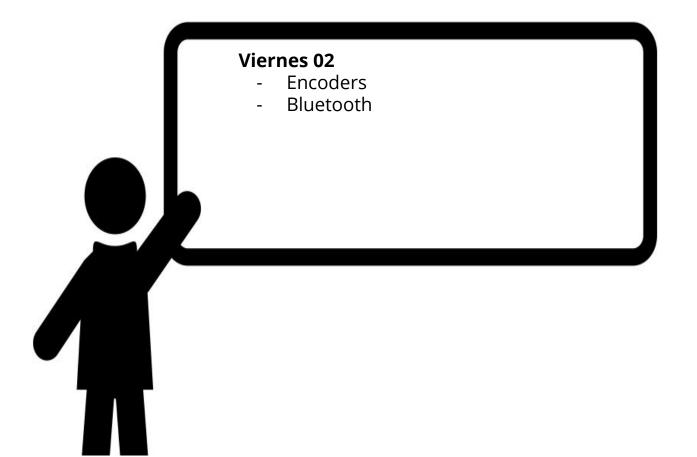


DÍA 5: ENCODERS Y BLUETOOTH



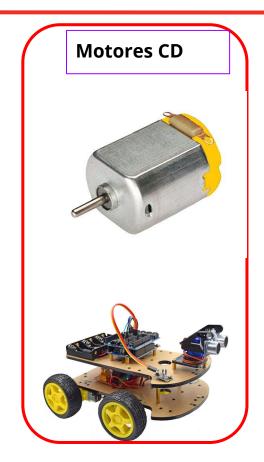
Contenido:





Motores





Servomotores



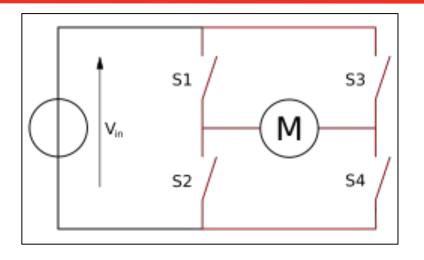
Motores paso a paso





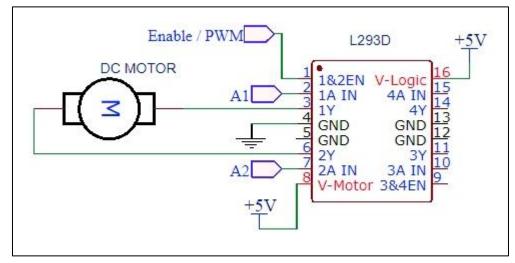
Control de un motor de CD







proporciona: 600mA (por canal)



Control de un motor de CD





Corriente a rotor bloqueado (stall): 6.5 A

CYTRON MDD10A

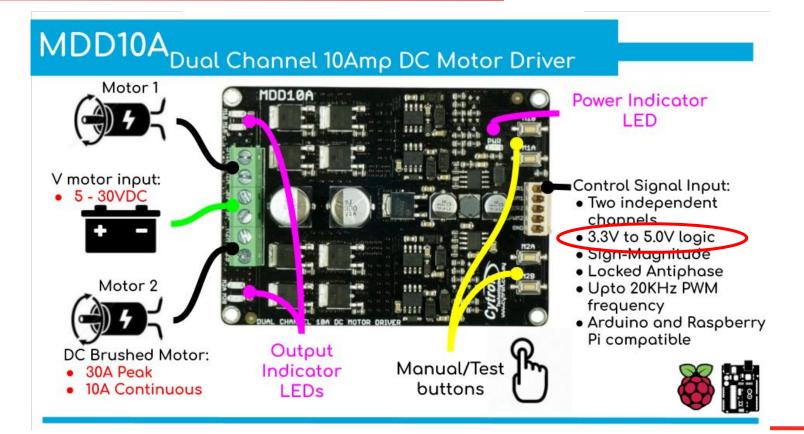
10Amp 5V-30V DC Motor Driver (2 Channels)



https://www.cytron.io/p-10amp-5v-30v-dc-motor-driver-2-channels

Pines del controlador de motores



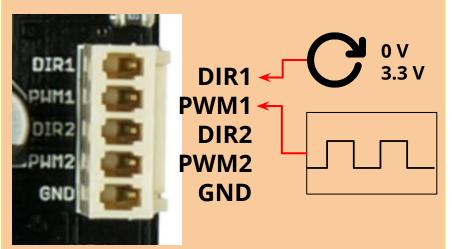


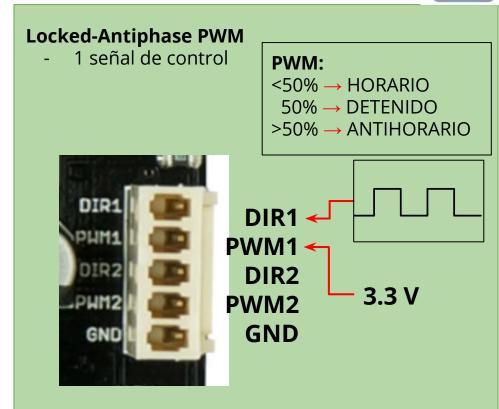
Cytron MDD10A: 2 tipos de control PWM



Sign-Magnitude PWM

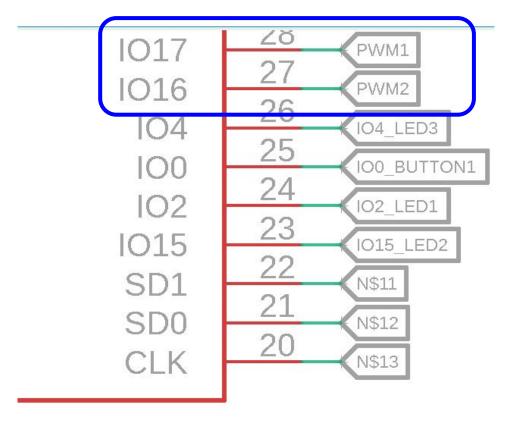
- 2 señales de control para controlar la Velocidad y Dirección





Conexión de motores en el robot móvil





17 - Llanta DERECHA

16 - Llanta IZQUIERDA

Ejemplo: D4_02_Prueba_motores

WiFi [introducción]



3.5 WiFi

ESP32 implementa el protocolo TCP/IP y full 802.11 b/g/n Wi-Fi MAC. Soporta el "Basic Service Set" (BSS) STA y SoftAP bajo operaciones de Función de Control Distribuido (DCF). El manejo de energía se maneja con interacción con el "host" mínima para minimizar el periodo de ciclo-activo.

3.5.1 WiFi Radio y "Baseband"

Soporta las siguientes características:

- 802.11b/g/n
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32 (RX)
- 802.11n 0.4 s guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2×1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity

Servidor Web



¿Qué es un servidor web y cómo funciona?

El servidor web es un lugar que almacena, procesa y entrega páginas web a clientes web. El cliente web no es más que un navegador web en nuestras computadoras portátiles y teléfonos inteligentes. La comunicación entre el cliente y el servidor se lleva a cabo utilizando un protocolo especial llamado Protocolo de transferencia de hipertexto (HTTP).



En este protocolo, un cliente inicia la comunicación haciendo una solicitud para una página web específica usando HTTP y el servidor responde con el contenido de esa página web o un mensaje de error si no puede hacerlo (como el famoso Error 404). Las páginas entregadas por un servidor son principalmente documentos HTML.

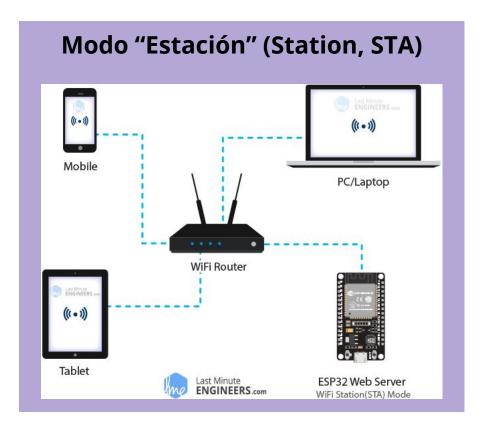
Modos de operación de la ESP32

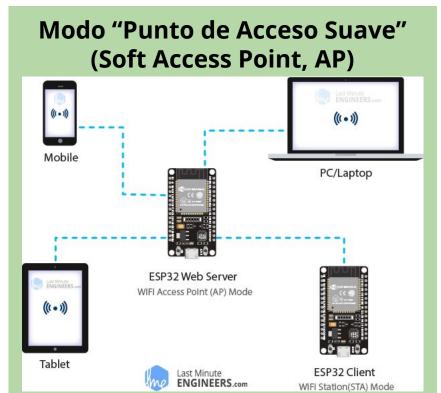


Una de las mejores características que ofrece ESP32 es que no solo puede conectarse a una red WiFi existente y actuar como un servidor web, sino que también puede configurar una red propia, permitiendo que otros dispositivos se conecten directamente a ella y accedan a las páginas web. Esto es posible porque ESP32 puede operar en tres modos diferentes: modo de estación, modo de punto de acceso suave y ambos al mismo tiempo. Esto proporciona la posibilidad de construir redes de malla.

Modos de operación de la ESP32

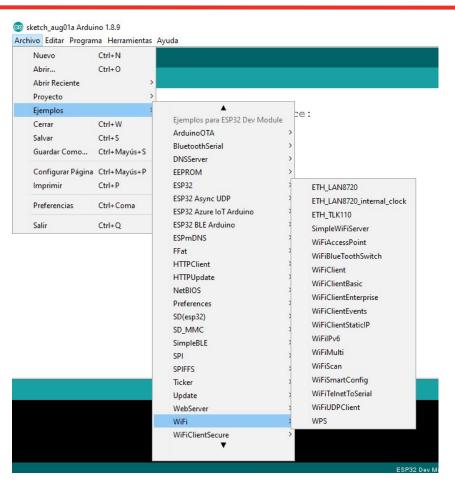






WiFi [Ejemplos]





WiFi [Ejemplo para conectarse a una red]



```
#include <WiFi.h>
const char* ssid = "ardwifi";
                                   // change this for your own network
const char* password = "ardwifi987"; // change this for your own network
                                                                                       Ejemplos:
void setup()
                                                                                       D4 03 wifi connection
                                                                                       D4 04 WiFiClient mod
  Serial.begin (115200);
                                                                                       D4 05 WiFi Server
 delay(10);
 // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
                                               Serial.println("");
  Serial.print("Connecting to ");
                                               Serial.println("WiFi connected");
  Serial.println(ssid);
                                               Serial.println("IP address: ");
                                               Serial.println(WiFi.localIP());
 WiFi.begin(ssid, password);
                                             void loop()
 while (WiFi.status() != WL CONNECTED) {
   delay (500);
   Serial.print(".");
```