



Ejemplos

Conexión del L298 a Arduino en el Kit

M₂

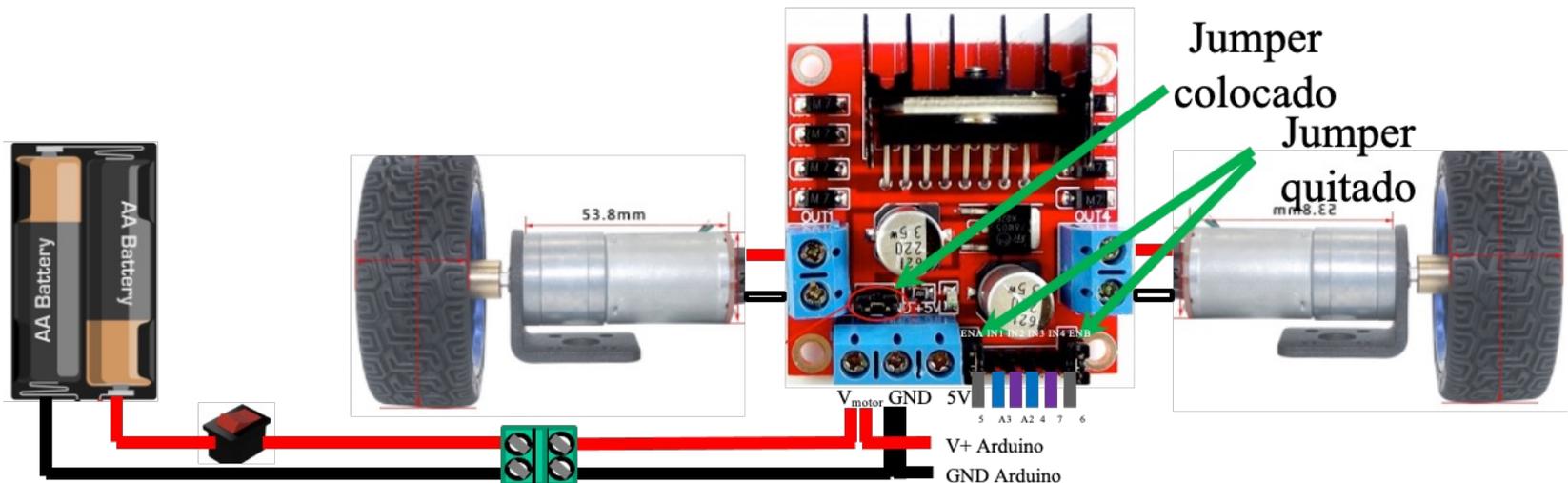
ControlMotores_1

- Jumper del regulador conectado

ENA = pin 5 ~ → Velocidad
EN1 = pin A3 → Sentido
EN2 = pin A2 → Sentido

EN3 = pin 4 → Sentido
EN4 = pin 7 → Sentido
ENB = pin 6 ~ → Velocidad

```
const uint8_t pin_L298_ENA = 5;  
const uint8_t pin_L298_IN1 = A3;  
const uint8_t pin_L298_IN2 = A2;  
const uint8_t pin_L298_ENB = 6;  
const uint8_t pin_L298_IN3 = 4;  
const uint8_t pin_L298_IN4 = 7;
```





Ejemplos

Ejemplo de control del robot a 6 Hilos

```
const uint8_t pin_L298_ENA = 5;
const uint8_t pin_L298_IN1 = A3;
const uint8_t pin_L298_IN2 = A2;
const uint8_t pin_L298_ENB = 6;
const uint8_t pin_L298_IN3 = 4;
const uint8_t pin_L298_IN4 = 7;

void configMotores(void) {
    pinMode(pin_L298_ENA,OUTPUT);
    pinMode(pin_L298_IN1,OUTPUT);
    pinMode(pin_L298_IN2,OUTPUT);
    pinMode(pin_L298_ENB,OUTPUT);
    pinMode(pin_L298_IN3,OUTPUT);
    pinMode(pin_L298_IN4,OUTPUT);

    digitalWrite(pin_L298_IN1, LOW);
    digitalWrite(pin_L298_IN2, LOW);
    digitalWrite(pin_L298_IN3, LOW);
    digitalWrite(pin_L298_IN4, LOW);
    analogWrite(pin_L298_ENA,0);
    analogWrite(pin_L298_ENB,0);
}
```

```
// Función que configura la velocidad y sentido de un motor
// velocidad toma valores entre +/- 100
void setVelocidadMotorD (float velocidad){

    // Comprueba los límites
    if (velocidad > 100) velocidad = 100;
    if (velocidad < -100) velocidad = -100;

    if (velocidad > 0) {
        // Si la velocidad es positiva
        digitalWrite(pin_L298_IN1, HIGH);
        digitalWrite(pin_L298_IN2, LOW);
        analogWrite(pin_L298_ENA, (int)((velocidad*255)/100));
    } else if (velocidad < 0) {
        // Si la velocidad es negativa
        digitalWrite(pin_L298_IN1, LOW);
        digitalWrite(pin_L298_IN2, HIGH);
        analogWrite(pin_L298_ENA, (int)((-velocidad*255)/100));
    } else {
        // Si la velocidad es positiva
        digitalWrite(pin_L298_IN1, LOW);
        digitalWrite(pin_L298_IN2, LOW);
        analogWrite(pin_L298_ENA,0);
    }
}
```



Ejemplos

Ejemplo de control del robot a 6 Hilos

```
// Función que configura la velocidad y sentido de un motor
// velocidad toma valores entre +/- 100
void setVelocidadMotorI (float velocidad){

    // Comprueba los límites
    if (velocidad > 100) velocidad = 100;
    if (velocidad < -100) velocidad = -100;

    if (velocidad > 0) {
        // Si la velocidad es positiva
        digitalWrite(pin_L298_IN3, HIGH);
        digitalWrite(pin_L298_IN4, LOW);
        analogWrite(pin_L298_ENB, (int)((velocidad*255)/100));
    } else if (velocidad < 0) {
        // Si la velocidad es negativa
        digitalWrite(pin_L298_IN3, LOW);
        digitalWrite(pin_L298_IN4, HIGH);
        analogWrite(pin_L298_ENB, (int)((-velocidad*255)/100));
    } else {
        // Si la velocidad es positiva
        digitalWrite(pin_L298_IN3, LOW);
        digitalWrite(pin_L298_IN4, LOW);
        analogWrite(pin_L298_ENB, 0);
    }
}
```

ControlMotores_1

```
void setup() {
    configMotores();
}

void loop() {
    int vel = 70;

    //Recto
    setVelocidadMotorI (vel);
    setVelocidadMotorD (vel);
    delay(1000);

    //Gira
    setVelocidadMotorI (vel);
    setVelocidadMotorD (-vel);
    delay(1000);

    // Atrás
    setVelocidadMotorI (-vel);
    setVelocidadMotorD (-vel);
    delay(1000);

    //Gira
    setVelocidadMotorI (-vel);
    setVelocidadMotorD (vel);
    delay(1000);
}
```



Ejemplos

Propuesta de actividades

– Actividad 2

- Modifica el programa de la actividad anterior para que, al recibir por el puerto serie un número entre -100 y 100, ponga uno de los motores a esa velocidad.
- Haz el programa más robusto haciendo que si el número es superior a 100 o inferior a -100, sature la salida a 100 o -100 respectivamente.

ControlMotores_2



Ejemplos

Ejemplo de control del robot a 6 Hilos

ControlMotores_2

```
void setup() {
  configMotores();
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  // print the three numbers in one string as hexadecimal:
  Serial.print("Introduzca valor de la velocidad del robot (-100 a 100):\n");
}

void loop() {
  while (Serial.available() > 0) {
    // lee el siguiente valor entero disponible.
    int value = Serial.parseInt();

    // Espera hasta que se haya recibido algún dato por el puerto serie.
    if (Serial.read() == ' ') {
      // Se espera leer valores entre -100 y 100
      setVelocidadMotorI (value);
      setVelocidadMotorD (value);
      Serial.print("Velocidad actual: ");
      Serial.println(value);
      Serial.print("Introduzca nuevo valor de la velocidad del robot (-100 a 100):\n");
    }
  }
}
```



Generación de señales de control

Propuesta de actividades

– Actividad 3

- Modifica el programa de la actividad anterior para que sea capaz de recibir la velocidad de los dos motores a la vez. Por ejemplo, "0 -20", "30 -30", "-20 80".

ControlMotores_3



Ejemplos

Ejemplo de control del robot a 6 Hilos

ControlMotores 3

```
void setup() {
  configMotores();
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  // print the three numbers in one string as hexadecimal:
  Serial.print("Introduzca nuevo valor de las velocidades separadas por espacio (-100 a 100):\n");
}

void loop() {

  while (Serial.available() > 0)
  {
    //Create a place to hold the incoming message
    static char message[MAX_MESSAGE_LENGTH];
    static unsigned int message_pos = 0;

    //Read the next available byte in the serial receive buffer
    char inByte = Serial.read();

    //Message coming in (check not terminating character) and guard for over message size
    if ( inByte != endChar && (message_pos < MAX_MESSAGE_LENGTH - 1) )
    {
      //Add the incoming byte to our message
      message[message_pos] = inByte;
      message_pos++;
    }
    //Full message received...
  }
  else
  {

```



Ejemplos

Ejemplo de control del robot a 6 Hilos

ControlMotores_3

```
//Full message received...
else
{
    int vel_der = 0;
    int vel_izq = 0;

    //Add null character to string
    message[message_pos] = '\0';

    //Print the message (or do other things)
    Serial.println(message);

    //Reset for the next message
    message_pos = 0;

    // Se lee las dos velocidades separadas por espacio
    sscanf(message,"%i %i", &vel_izq, &vel_der);

    setVelocidadMotorI (vel_izq);
    setVelocidadMotorD (vel_der);

    Serial.print("Velocidades seleccionadas: ");
    Serial.print(vel_izq);
    Serial.print(" ");
    Serial.println(vel_der);
    Serial.print("Introduzca nuevo valor de las velocidades separadas por espacio (-100 a 100):\n");
}
}
}
```



Generación de señales de control

Conexión del L298 a Arduino en el Kit

- Jumpers de ENA y ENB conectados (Motores activos)
- Jumper del regulador conectado

ENA = -----

EN1 = pin 6 ~ → Velocidad

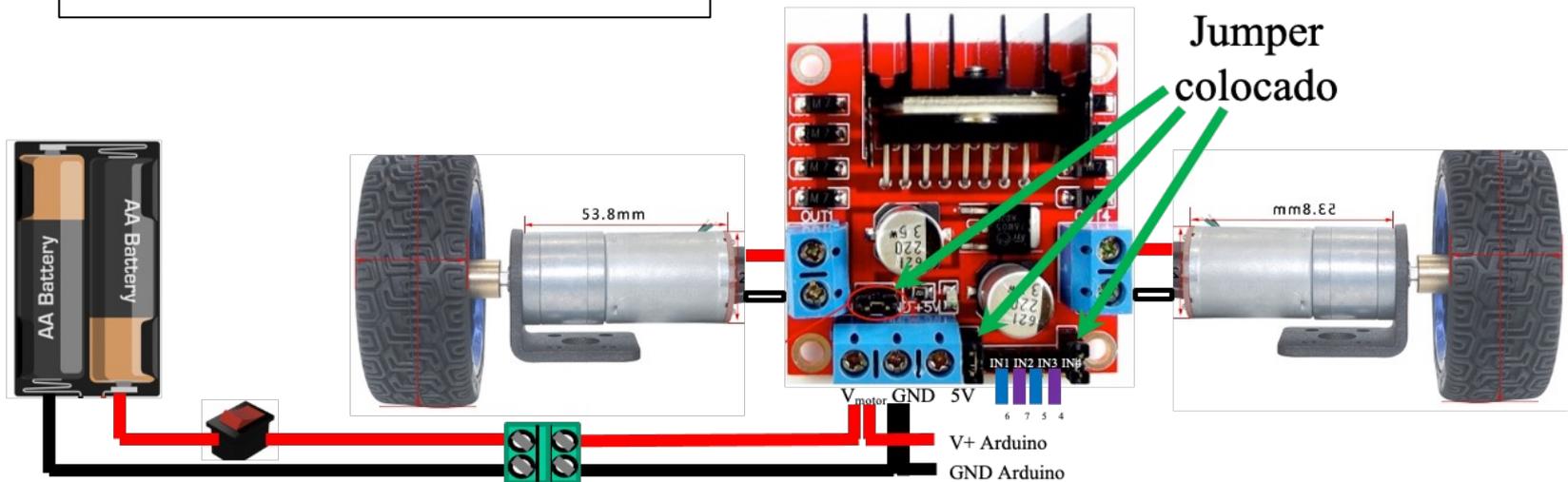
EN2 = pin 7 → Sentido

EN3 = pin 5 ~ → Velocidad

EN4 = pin 4 → Sentido

ENB = -----

```
const uint8_t pin_L298_IN1 = 6;  
const uint8_t pin_L298_IN2 = 7;  
const uint8_t pin_L298_IN3 = 5;  
const uint8_t pin_L298_IN4 = 4;
```

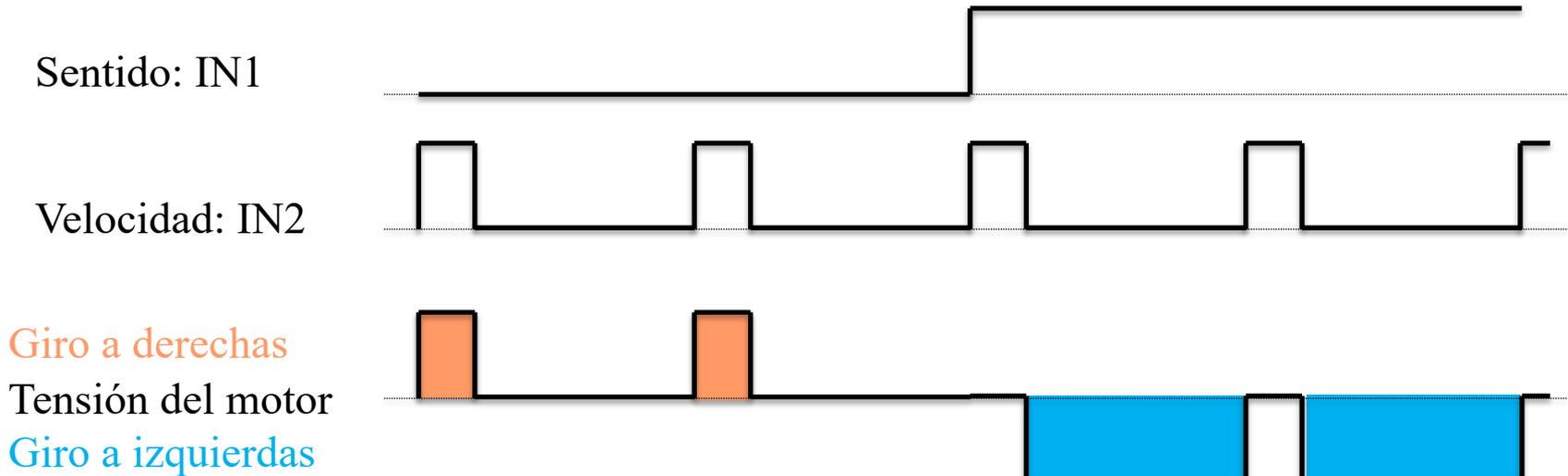




Generación de señales de control

Conexión del L298 a Arduino en el Kit

- Velocidad derecha: IN1 \rightarrow 6 Sentido derecha: IN2 \rightarrow 7
 - Velocidad izquierda: IN3 \rightarrow 5 Sentido izquierda: IN4 \rightarrow 4
- Jumpers de ENA y ENB conectados (Motores activos)
- Jumper del regulador conectado





Generación de señales de control

Ejemplo

- Conectar los motores al puente en H y éste a Arduino
- Función que recibe un % de velocidad de +100 a -100 y controla la velocidad de un motor

Pseudocódigo

```
SetVelocidad (velocidad)
```

```
  Si (velocidad > 0)
```

```
    Sentido = 0
```

```
    PWM = velocidad * 255 / 100
```

```
  Si (velocidad < 0)
```

```
    Sentido = 1
```

```
    PWM = 255 - velocidad * 255 / 100
```



Generación de señales de control

Conexión de un motor del L298 a Arduino a 4H

M_4

- Jumper de ENB conectado (Motores activos)
- Jumper del regulador conectado

ControlMotores_4

```
const uint8_t pin_L298_IN1 = 6;
const uint8_t pin_L298_IN2 = 7;
const uint8_t pin_L298_IN3 = 5;
const uint8_t pin_L298_IN4 = 4;

void configMotores(void) {
  pinMode(pin_L298_IN1,OUTPUT);
  pinMode(pin_L298_IN2,OUTPUT);
  pinMode(pin_L298_IN3,OUTPUT);
  pinMode(pin_L298_IN4,OUTPUT);

  analogWrite(pin_L298_IN1, 0);
  digitalWrite(pin_L298_IN2, LOW);
  analogWrite(pin_L298_IN3, 0);
  digitalWrite(pin_L298_IN4, LOW);
}
```

ENA = -----
EN1 = pin 6 ~ → Velocidad
EN2 = pin 7 → Sentido

EN3 = pin 5 ~ → Velocidad
EN4 = pin 4 → Sentido
ENB = -----



Ejemplos

Ejemplo de control del robot a 4 Hilos

ControlMotores_4

```
void setVelocidadMotorD_4H (float velocidad){

    // Comprueba los límites
    if (velocidad > 100)  velocidad = 100;
    if (velocidad < -100) velocidad = -100;

    if (velocidad > 0) {
        // Si la velocidad es positiva
        analogWrite(pin_L298_IN1, (int)((velocidad*255)/100));
        digitalWrite(pin_L298_IN2, LOW);
    } else if (velocidad < 0) {
        // Si la velocidad es negativa
        analogWrite(pin_L298_IN1, (int)(255-(-velocidad*255)/100));
        digitalWrite(pin_L298_IN2, HIGH);
    } else {
        // Si la velocidad es positiva
        analogWrite(pin_L298_IN1, 0);
        digitalWrite(pin_L298_IN2, LOW);
    }
}
```



Ejemplos

Ejemplo de control del robot a 4 Hilos

ControlMotores_4

```
void setVelocidadMotorI_4H (float velocidad){

    // Comprueba los límites
    if (velocidad > 100) velocidad = 100;
    if (velocidad < -100) velocidad = -100;

    if (velocidad > 0) {
        // Si la velocidad es positiva
        analogWrite(pin_L298_IN3, (int)((velocidad*255)/100));
        digitalWrite(pin_L298_IN4, LOW);
    } else if (velocidad < 0) {
        // Si la velocidad es negativa
        analogWrite(pin_L298_IN3, (int)(255-(-velocidad*255)/100));
        digitalWrite(pin_L298_IN4, HIGH);
    } else {
        // Si la velocidad es positiva
        analogWrite(pin_L298_IN3, 0);
        digitalWrite(pin_L298_IN4, LOW);
    }
}
```



Ejemplos

Ejemplo de control del robot a 4 Hilos

ControlMotores_4

```
void setup() {
  configMotores();
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  // print the three numbers in one string as hexadecimal:
  Serial.print("Introduzca valor de la velocidad del robot (-100 a 100):\n");
}

void loop() {
  while (Serial.available() > 0) {
    // lee el siguiente valor entero disponible.
    int value = Serial.parseInt();

    // Espera hasta que se haya recibido algún dato por el puerto serie.
    if (Serial.read() == '\n') {
      // Se espera leer valores entre -100 y 100
      setVelocidadMotorD_4H (value);
      setVelocidadMotorI_4H (value);
      Serial.print("Velocidad actual: ");
      Serial.println(value);
      Serial.print("Introduzca nuevo valor de la velocidad del robot (-100 a 100):\n");
    }
  }
}
```



Generación de señales de control

Propuesta de actividades

– **Actividad**

- Hacer un programa que haga que el robot vaya en línea recta.
- Modificar las funciones de control de los motores para añadirles un factor de ajuste que permita que el robot vaya en línea recta cuando se pasa el mismo valor a los dos motores.



Ejemplos

Uso de librería L298N de Andrea Lombardo

M_4

– <https://github.com/AndreaLombardo/L298N>

```
// With Enable pin to control speed  
L298N myMotor(EN, IN1, IN2);
```

```
// Without Enable pin with jumper in place  
L298N myMotor(IN1, IN2);
```

```
// Create one motor instance  
L298N motor(EN, IN1, IN2);  
  
void setup()  
{  
    // Set initial speed  
    motor.setSpeed(70);  
}
```



Ejemplos

Uso de librería L298N de Andrea Lombardo

– <https://github.com/AndreaLombardo/L298N>

Method	Params	Description
setSpeed	unsigned short pwmVal	Set the PWM value used to determine the motor speed. Value from 0 to 255.
getSpeed	none	Get the speed previously set.
forward	none	Run motor in forward direction (depends on wiring).
forwardFor	unsigned long delay	Run motor in forward direction for a time specified by delay.
forwardFor	unsigned long delay, CallbackFunction callback	Run motor in forward direction for a time specified by delay, after moving execute a callback function.
backward	none	Run motor in backward direction (depends on wiring).
backwardFor	unsigned long delay	Run motor in backward direction for a time specified by delay.
backwardFor	unsigned long delay, CallbackFunction callback	Run motor in backward direction for a time specified by delay, after moving execute a callback function.



Ejemplos

Uso de librería L298N de Andrea Lombardo

– <https://github.com/AndreaLombardo/L298N>

run	uint8_t direction	Move motor. To specify the direction use one of <i>L298N::FORWARD</i> , <i>L298N::BACKWARD</i> or <i>L298N::STOP</i> .
runFor	unsigned long delay, L298N::Direction direction	Like <i>forwardFor</i> or <i>backwardFor</i> but more flexible. To specify the direction use one of <i>L298N::FORWARD</i> , <i>L298N::BACKWARD</i> or <i>L298N::STOP</i> .
runFor	unsigned long delay, L298N::Direction direction, CallbackFunction callback	Like previous with the ability to execute a callback function.
stop	none	Stop the motor.
reset	none	Used to re-enable motor movements after the use of <i>runFor</i> , <i>forwardFor</i> or <i>backwardFor</i> methods.
isMoving	none	Returns a boolean indicating if motor is running or not.
getDirection	none	Returns the current <i>L298N::Direction</i> .



Ejemplos

Uso de librería L298N de Andrea Lombardo

– <https://github.com/AndreaLombardo/L298N>

Same thing for L298NX2 variant

```
// With Enable pin to control speed

           motor A           motor B
           |-----| |-----|
L298NX2 myMotors(EN_A, IN1_A, IN2_A, EN_B, IN1_B, IN2_B);
```

```
// Without Enable pin and jumper in place

           motor A           motor B
           |-----| |-----|
L298NX2 myMotors(IN1_A, IN2_A, IN1_B, IN2_B);
```



Ejemplos

Uso de librería L298N de Andrea Lombardo

```
// Initialize both motors
L298NX2 motors(EN_A, IN1_A, IN2_A, EN_B, IN1_B, IN2_B);

void setup()
{
  // Set initial speed for both motors
  motors.setSpeed(80);
}

void loop()
{
  // Tell both motors to go forward (may depend by your wiring)
  motors.forward();
  delay(3000);

  // Stop
  motors.stop();
  delay(3000);

  // Change individual speeds
  motors.setSpeedA(255);
  motors.setSpeedB(90);
}
```



Ejemplos

Uso de librería L298N de Andrea Lombardo

– <https://github.com/AndreaLombardo/L298N>

Method	Params
setSpeed	unsigned short pwmVal
forward	none
forwardFor	unsigned long delay, CallbackFunction callback
forwardFor	unsigned long delay
backward	none
backwardFor	unsigned long delay, CallbackFunction callback
backwardFor	unsigned long delay
run	uint8_t direction
runFor	unsigned long delay, L298N::Direction direction
runFor	unsigned long delay, L298N::Direction direction, CallbackFunction callback
stop	none
reset	none



Referencias

Referencias

- Maxon DC motor and Maxon EC motor. Key information
 - https://www.maxongroup.com/medias/sys_master/root/8815460712478/DC-EC-Key-Information-14-EN-42-50.pdf?attachment=true
- How to select a DC Micromotor. Faulhaber
 - <http://fab.cba.mit.edu/classes/961.04/topics/actuator/DCmicromotor.pdf>
- El módulo controlador de motores L298N. Prometec.net
 - <https://www.prometec.net/l298n/>
- Arduino y L298N: montaje y conexión
 - <https://www.prometec.net/coche-arduino-l298n/>
- Controlar motores de corriente continua con Arduino y L298N. Luis Llamas
 - <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>
- Tutorial del uso del módulo L298N. naylampmechatronics.com
 - https://naylampmechatronics.com/blog/11_tutorial-de-uso-del-modulo-l298n.html
- Puertos Analógicos Arduino Avanzado
 - <https://aprendiendoarduino.wordpress.com/2017/09/05/puertos-analogicos-arduino-avanzado/>



Referencias

Referencias

- Secrets of Arduino PWM
 - <https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM>