

# Carro Robot controlado por señales mioeléctricas EMG (2020-1)

Jonathan Fuentes

Docente Politécnico Grancolombiano

*Ingeniería Sistemas, Arquitectura del computador,*

*Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia*

Juan Barbosa Montiel Estudiante

*Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia*

[jubarbosa3@poligran.edu.co](mailto:jubarbosa3@poligran.edu.co)

Andrés Gómez Estudiante

*Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia*

Víctor Sierra Estudiante

*Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia*

Mauro Martinez Estudiante

*Institución Universitaria Politécnico Grancolombiano, Bogotá, Colombia*

[mamartinez87@poligran.edu.co](mailto:mamartinez87@poligran.edu.co)

*Miembros del equipo de trabajo* **PROMETHEUS**

**Resumen** - En el presente trabajo se describe la construcción de un carro robot controlado por señales Mioeléctricas EMG, las cuales son tomadas por los electrodos que son conectados al brazo del usuario. Esas señales, son controladas por unos sensores superficiales de EMG. Los biopotenciales que pasan por los músculos del brazo son controlados por un microcontrolador marca Lolin referencia NodeMCU v3, en el cual dependiendo de la diferencia de potencial que circule por el músculo va activar los servomotores que controlan el carro robótico. Se utilizan 4 servomotores, manipulados por un puente H de referencia L298N y 4 sensores de EMG para darle movimiento al carro robot por en los cuatro hemisferios SUR, NORTE, ORIENTE, OCCIDENTE, por medio de un tráfico de datos y protocolos MQTT establecidos en un servidor maestro con tecnología NODEJS, un segundo mando de control (dispositivo Móvil) con tecnología Flutter, se estipula una página de consulta de movimientos del carro robótico el cual se debe visualizar por medio de tecnologías VueJS (frontend) y finalmente un almacenamiento de datos en sistemas no relacionales como lo es MongoDB.

**Abstrac** - This work describes the construction of a robot cart controlled by EMG Myoelectric signals, which are taken by electrodes that are connected to the user's arm. These signals are controlled by surface EMG sensors. Biopotentials passing through the arm muscles are controlled by a Lolin branded microcontroller reference NodeMCU v3, in which depending on the difference in

potential circulating through the muscle will activate the servo motors that control the robotic cart. 4 servo motors are used, manipulated by an L298N reference H-bridge and 4 EMG sensors to give the robot cart movement through the four southern, NORTH, ORIENTE, OCCIDENTE hemispheres, through data traffic and MQTT protocols established on a master server with NODEJS technology, a second control knob (Mobile device) with Flutter technology, stipulates a robotic cart movement query page which must be displayed by technologies VueJS (frontend) and finally a data warehouse on non-relational systems such as MongoDB.

**Índice de Términos** – NODEJS, Arduino, VueJS, Flutter, NodeMCU, Controlador, Carro Robot, EMG, Mioeléctricas, Rectificación Amplificación, noSQL, broker.

## I. INTRODUCCIÓN

Hay dos métodos para utilizar en EMG: Uno es el superficial, y el otro método es el intramuscular. Para llevar a cabo un EMG intramuscular, se usa una aguja hipodérmica que se inserta a través de la Piel hasta que entre al tejido muscular.

El método Intramuscular EMG puede ser considerado demasiado invasivo o innecesario en algunos casos. En su lugar, el método superficial emplea una superficie en la cual el electrodo se puede utilizar para controlar la imagen general de la activación muscular, a diferencia de la actividad de sólo unas pocas fibras como se observa utilizando un EMG intramuscular. Esta técnica se utiliza en una serie de ajustes, por ejemplo, en la fisioterapia, la activación muscular se controlará mediante EMG superficial y los pacientes tienen un estímulo auditivo o visual para ayudarles a saber cuándo se está activando el músculo.

Una unidad motora se define como un motor neurona y todas las fibras musculares que inerva. Cuando una unidad motora se activa, el impulso llamado potencial de acción se desplaza de la neurona motora hacia el músculo. El área donde el nervio hace contacto con el músculo se llama unión neuromuscular. Después de que el potencial de acción se transmite a través de la unión neuromuscular, se obtiene un potencial en todas las fibras musculares inervadas por la unidad motora particular. La suma de toda esta actividad eléctrica se conoce como un potencial motor de la acción de la unidad (MUAP). La actividad electrofisiológica de las múltiples unidades motoras es la señal que normalmente se evalúa durante un EMG. La composición de la unidad motora, el número de fibras musculares por unidad motora, el tipo metabólico de las fibras musculares y muchos otros factores afectan la forma de los potenciales de unidad motora en el miograma.

## II. OBJETIVOS

### Objetivo general

- Diseñar, programar e implementar una interfaz accesible permitiendo el control de un carro robot manipulado por señales mioeléctricas.

### Objetivos específicos

- Identificar las herramientas de hardware y software con las tecnologías vigentes que permitan implementar un prototipo de un control mioeléctricas en un carro robot.
- Implementar hardware para recibir señales externas y controlar elementos que permitan interactuar con el software mediante la tecnología bluetooth.
  - Desarrollar el algoritmo que permita controlar el comportamiento del carro robot con el uso de un control mioeléctricas.

## III. ESTUDIO MECÁNICO

### Materiales y Herramientas necesarias

- Kit de chasis de robot 4WD
- Modulo puente H LM298H
- Soporte de batería
- Mini protoboard
- Jumpers
- Soldador
- NodeMCU Lolin V3
- Power bank

Un carro robot consta de los siguientes componentes

#### • Estructura / chasis

Esto dependerá del diseño de cada uno, puede hacer, o comprar uno hecho. El chasis básicamente es en donde se va a poner los cuatro motores con llanta y los demás componentes.



#### • Actuadores

Básicamente 4 motores DC con caja reductora y su respectiva llanta, una velocidad promedio puede ser 100RPM, pero dependerá de la velocidad que uno desee.



El voltaje dependerá de la batería, Si alimentamos un motor con un voltaje inferior al nominal, su velocidad disminuye, en cambio si alimentamos con un voltaje superior podemos malograr el motor, por eso debemos de tener en cuenta las especificaciones técnicas del motor.

Esto tal vez sea lo más difícil de establecer, lo primero que hay que tener en cuenta es el voltaje. NodeMCU trabaja con un voltaje entre 3V y 5V, aparte de esto hay que tener en cuenta el voltaje del motor. También hay que considerar la corriente y la capacidad, por ejemplo, si usamos una pila económica de 9V nos durará menos de un minuto, se recomienda que la capacidad sea superior a 1500mAh y un voltaje ideal de 10V.

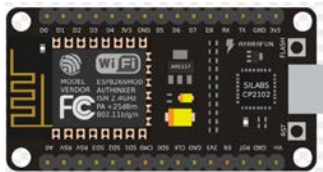
A continuación, mostramos las opciones para un Motor de 6V:

- **Opción 1:** 4 pilas AA para los motores y una de 9V para el Arduino
- **Opción 2:** 1 celda de litio (batería de celular) para los motores y una pila de 9V para el Arduino
- **Opción 4:** Cualquier batería superior a 6V (de suficiente corriente) y un step down para bajar el voltaje.  
(Recomendada)
- **Opción 4:** 2 Power Bank en serie cada una de un voltaje de 5V a 2500mAh la cual alimentara la tarjeta NodeMCU, Puente H y Motores.

- Controlador

### NodeMCU Lolin v3

La placa de desarrollo NodeMCU usa el módulo ESP-12E que contiene el chip ESP8266 que tiene un microprocesador LX106 RISC de 32 bits Tensilica Xtensa® que funciona a una frecuencia de reloj ajustable de 80 a 160 MHz y admite RTOS (Real Time Operating System) o sistema operativo de tiempo real.



El ESP8266 integra el transceptor Wi-Fi 802.11b / g / n HT40, por lo que no solo puede conectarse a una red WiFi e interactuar con Internet, sino que también puede configurar una red propia, permitiendo que otros dispositivos se conecten directamente a esta red. Esto hace que el ESP8266 NodeMCU sea aún más versátil.

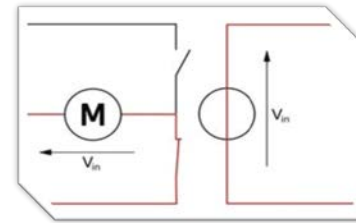
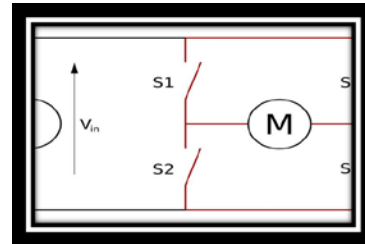
### NoceMCU Pines Y Periféricos

El ESP8266 NodeMCU tiene un total de 17 pines GPIO repartidos en los encabezados de pines a ambos lados de la placa de desarrollo. Estos pines se pueden asignar a todo tipo de tareas periféricas, que incluyen:

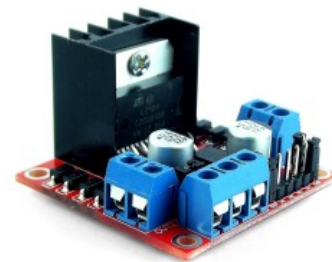
- Canal ADC: un canal ADC de 10 bits.
- Interfaz UART: la interfaz UART se usa para cargar código en serie.
- Salidas PWM: pines PWM para atenuar los LED o controlar motores.
- Interfaz SPI, I2C e I2S: interfaz SPI e I2C para conectar todo tipo de sensores y periféricos.
- Interfaz I2S: interfaz I2S si desea agregar sonido a su proyecto.

### Puente H L298N

El puente H es el que nos permite invertir el giro de un motor, necesitamos un puente H que tenga dos canales para poder controlar dos motores.



El puente H que usaremos es el L298N. Este módulo aparte de los pines para controlar la dirección tienen un pin (Enable ) por donde podemos enviar PWM y controlar la velocidad del carrito.

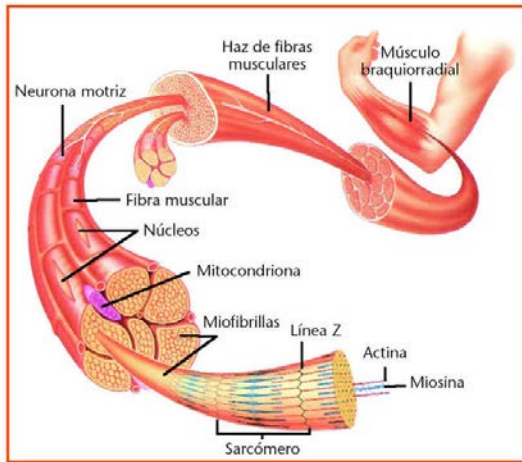


## IV. ESTUDIO EMG

Antes de continuar debemos conocer algo de anatomía:

### Músculos, corrientes eléctricas y proteínas

Los **músculos esqueléticos** están formados por fibras. Éstas se constituyen de miofibrillas, que a su vez están formadas por miofilamentos. Los miofilamentos son pequeños hilos hechos de dos proteínas distintas: la miosina y la actina. La actina se entrelaza con la miosina, formando una especie de “puentes” que más adelante servirán para tirar de los filamentos y contraer el músculo.

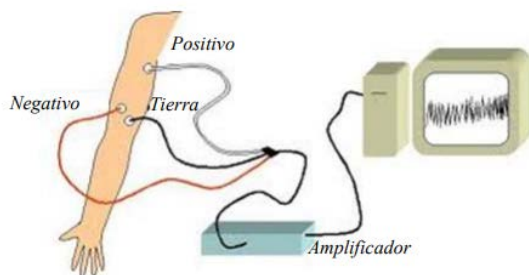


Los músculos se contraerán y se relajarán en función de cómo interactúen entre sí la miosina y la actina. Cuando el músculo está en lo que se llama “reposo relativo” (las fibras musculares nunca están totalmente en reposo) no hay interacción entre estas proteínas. Esto se debe a la acción de otras dos proteínas, la troponina y la tropomiosina, que bloquean la interacción entre la miosina y la actina.

Los impulsos nerviosos provenientes del cerebro liberan iones de calcio ( $\text{Ca}^{++}$ ), que cambia la configuración y desplaza a la troponina y la tropomiosina. Esto permite que los miofilamentos de actina formen puentes con los de miosina y se desplacen sobre ellos, acortando la longitud de las fibras musculares, lo que contrae el músculo.

### Materiales y Herramientas necesarias

- Power Bank
- Protoboard
- Jumpers
- NodeMCN Lolin V3
- Sensor Myowere



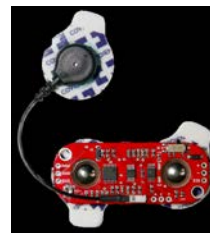
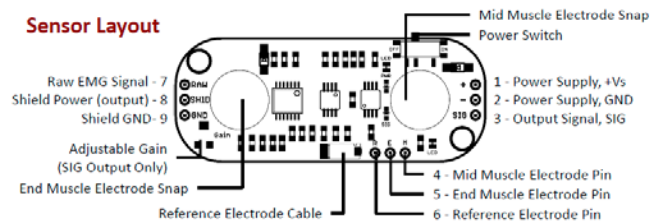
La captura de señales EMG consta de los siguientes componentes:

#### • Estructura / Electrodo

La Electromiografía (abreviado EMG) es una técnica para medir la actividad de las corrientes eléctricas de los músculos

esqueléticos. Gracias a esta técnica los médicos pueden diagnosticar enfermedades neuromusculares y desórdenes en el control motor.

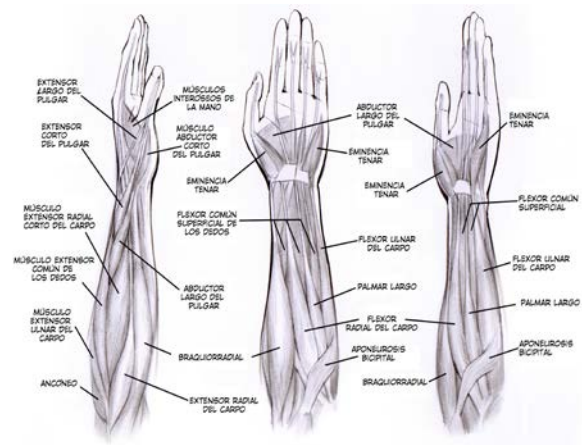
Dispositivos más simples constan de dos electrodos: un emisor y un GND, que miden la conductividad de la parte externa del músculo. Sin embargo, se suelen utilizar tres electrodos para eliminar ruido: dos emisores y un GND. Posteriormente, la señal de los electrodos se procesa: primero se amplifica y después le aplica un filtro de paso alto para eliminar interferencias; por esta razón usaremos el sensor Myowere.



#### • Actuador

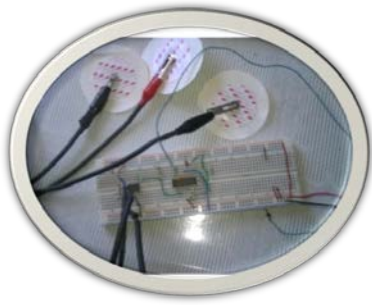
La mano humana es un instrumento exquisito, que te permite tomar un cautín y ponerlo con precisión para construir muy pequeñas placas electrónicas, y puede mantener una fuerza de agarre equivalente al peso de todo tu cuerpo. La anatomía muscular de la mano a fascina a médicos desde la antigüedad hasta el día de hoy.

En la siguiente figura puedes ver que la mano tiene músculos “por dentro” (intrínsecos) y “por fuera” (extrínsecos). Aquí estamos mirando el interior del antebrazo, que tiene los músculos que permiten flexión (flexión de los dedos, muñeca, agarre, etc).





## Montaje



## V. ESTUDIO SOFTWARE

*Este apartado se procederá a explicar la programación con la cual se ha conseguido realizar el control del carro robot. Para ello, el software cuenta con diversos apartados diferenciados, entre los que destacan los siguientes:*

- *Creación del servidor y microservicios mediante programación en Node.js*
- *Implementación de un broker en la aplicación emqx, la cual es un broker mqtt open source.*
- *Base de datos en mongoDB la cual es una base de datos distribuida.*
- *Google Cloud Platform, es una suite que contiene diversos servicios que funcionan en la misma infraestructura que utiliza Google de manera interna.*

*Para implementar el código se utilizarán varias herramientas de programación, como pueden ser las librerías que se emplearán para las diferentes comunicaciones que hay que realizar. Otras herramientas que se van a usar son las subrutinas, interrupciones, containers, mediante las cuales se llevarán a cabo diversas operaciones.*

### **Microservicios en node.js:**

*Para movimientos del carro robot, los creamos mediante 4 microservicios independientes bajo la arquitectura node.js express (left, right, back, down), los cuales se van a comunicar mediante protocolo HTTP, y haciendo uso de Async/Await.*

### **MongoDB:**

*Para la ase de datos utilizamos MongoDB la cual es una base de datos noSQL, la cual es una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. Es basada en la creación de documentos*

### **Arduino:**

Este código convertirá la señal EMG proveniente de los músculos en un número de pulsos que serán aplicados a los

motores del carro para que giren a un determinado movimiento por cada impulso de entrada. Mientras más actividad de tus músculos, el NodeMCU enviara más impulsos al motor, lo que resultará en que el Carro se mueva. El NodeMCU del Carro debe estar a la escucha de la información enviado desde el NodeMCU para saber que movimiento debe realizar, este ultimo por protocolo MQTT el cual se explicara mas adelante.

Para cargar el código al Arduino antes debemos hacer lo siguiente:

- En el IDE de Arduino, abra la ventana de preferencias e ingrese la URL: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) en el campo URL de Boards Manager adicional y haga clic en Aceptar.
- En el MENÚ, seleccione: Herramientas → Tablero → Administrador de tableros y desplácese hacia abajo para elegir esp8266 por ESP8266 Community y haga clic en INSTALAR. La instalación lleva unos 3 minutos.
- Instale el controlador USB (en este caso CH340).
- Reinicie Arduino IDE y seleccione nuestra placa de la opción de menú: Herramientas → Placa → NodeMCU 1.0 (Módulo ESP-12E).
- Luego, especificamos la frecuencia de CPU correcta (Herramientas → Frecuencia de CPU → 80MHz) y la Velocidad de carga (Herramientas → Velocidad de carga → 115200).
- Seleccione la opción correcta para el Puerto (Herramientas → Puerto → COM5).

¡Y ahora estamos listos para cargar nuestro código a NodeMCU!

### **MQTT:**

El conocimiento completo de un protocolo no es una tarea sencilla y requiere de bastante dedicación.

MQTT es un poderoso protocolo que se ha convertido en el líder dentro de la conectividad M2M (Machine to Machine), es particular dentro del IoT o Internet de las Cosas.

Este tipo de arquitecturas lleva asociada otra interesante característica: la comunicación puede ser de uno a uno o de uno a muchos.

los "topics" tienen una estructura jerárquica gracias a la cual podemos establecer relaciones padre-hijo y, al subscribirnos a

un topic padre, recibir también la información de sus hijos.

Vamos a hacer uso de un brokers no muy conocido que existen para MQTT: EMQX es un broker OpenSource ampliamente utilizado debido a su ligereza lo que nos permite, fácilmente, emplearlo en gran número de ambientes, incluso si éstos son de pocos recursos.

Una vez que tenemos instalado EMQX, podemos proceder a crear nuestro primer emisor y subscriptor siguiendo para ello una jerarquía en este caso Carrito – Express – EMG

Dentro de esta estructura, nuestros únicos "emisores" son el sensor EMG y frontend Flutter Cada uno de ellos, lo vamos a asignar a un topic propio de la siguiente forma:

- EMGcar/NODE => server Express Flutter
- EMGcar/EMG => NodeMCU Myoware

subscribirnos los topics:

- EMGcar/Move => Microservicios
- EMGcar/ROBOTCAR => NodeMCU carrito

## VI. ESTUDIO FINANCIERO

## VII. PRECIO

## VIII. VALOR AGREGADO

## IX. RECUPERACIÓN INVERSIÓN

## X. EDT

## XI. ENTREGABLES

## XII. RESULTADOS

## XIII. CONCLUSIONES

## XIV. BIBLIOGRAFÍA