



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
Y AVIÓNICA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN Y AVIÓNICA**

**TEMA: “IMPLEMENTACIÓN DE UN CONTROL DE
MOVIMIENTO DE UN BRAZO ROBÓTICO MEDIANTE
ELECTROMIOGRAFÍA A TRAVÉS DE LA TECNOLOGÍA MYO
ARMBAND PARA EL LABORATORIO DE INSTRUMENTACIÓN
VIRTUAL”**

AUTOR: PUMASUNTA SACA EDISON BLADIMIR

DIRECTORA: ING. VIERA GARZON IRINA GABRIELA

LATACUNGA

2019



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  **TÉCNOLOGÍAS**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
& AVIÓNICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación **“IMPLEMENTACIÓN DE UN CONTROL DE MOVIMIENTO DE UN BRAZO ROBÓTICO MEDIANTE ELECTROMIOGRAFÍA A TRAVÉS DE LA TECNOLOGÍA MYO ARMBAND PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL”** realizado por el señor **PUMASUNTA SACA EDISON BLADIMIR**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **PUMASUNTA SACA EDISON BLADIMIR** para que lo sustente públicamente.

Latacunga, 14 de febrero del 2019

ING. VIERA GARZON IRINA GABRIELA
DIRECTORA



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  **TÉCNOLOGÍAS**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
& AVIÓNICA**

AUTORÍA DE RESPONSABILIDAD

Yo, **PUMASUNTA SACA EDISON BLADIMIR**, con cédula de identidad N° **0504129180**, declaro que este trabajo de titulación “**IMPLEMENTACIÓN DE UN CONTROL DE MOVIMIENTO DE UN BRAZO ROBÓTICO MEDIANTE ELECTROMIOGRAFÍA A TRAVÉS DE LA TECNOLOGÍA MYO ARMBAND PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 14 de febrero del 2019

PUMASUNTA SACA EDISON BLADIMIR

C.C 0504129180



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  **TÉCNOLOGÍAS**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
& AVIÓNICA**

AUTORIZACIÓN

Yo, **PUMASUNTA SACA EDISON BLADIMIR**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“IMPLEMENTACIÓN DE UN CONTROL DE MOVIMIENTO DE UN BRAZO ROBÓTICO MEDIANTE ELECTROMIOGRAFÍA A TRAVÉS DE LA TECNOLOGÍA MYO ARMBAND PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 14 de febrero del 2019

PUMASUNTA SACA EDISON BLADIMIR

C.C 0504129180

DEDICATORIA

La obtención de este título va dedicada, a mis padres Edison y Elsa que formaron en mí, los valores de responsabilidad, humildad y respeto; quienes me dieron la oportunidad de realizar una carrera para mi futuro.

A mi hermana Selena, por creer en mí y brindarme su apoyo. A todas aquellas personas que nunca dudaron de mis capacidades.

Edison

AGRADECIMIENTO

Agradezco a Dios por guiar mi camino y darme una familia maravillosa. A mis padres quienes me enseñaron el valor del trabajo, esfuerzo y dedicación; gracias a ellos estoy en este punto muy importante de mi vida.

A todas aquellas personas, quienes me brindaban sus sabias palabras, para nunca desistir de mi meta propuesta.

Edison

INDICE DE CONTENIDOS

PORTADA	
CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
INDICE DE CONTENIDOS	vii
ÌNDICE DE FIGURAS	xi
ÌNDICE DE TABLAS	xiv
ÌNDICE DE ECUACIONES.....	xv
RESUMEN.....	xvi
ABSTRAC.....	xvii
CAPÍTULO I	1
1.1 Antecedentes.....	1
1.2 Planteamiento del problema	3
1.3 Justificación e importancia.....	3
1.4 Objetivos.....	4
1.4.1 General	4
1.4.2 Específicos.....	4
1.5 Alcance.....	5
CAPÍTULO II	6
2.1 Electromiografía.....	6
2.1.1 Beneficio de las señales EMG.	6
2.1.2 Origen de señales.	7
2.2 Elementos de control del brazo robótico.	7
2.2.1 Sensores mioeléctricos.	7

2.2.2	Myo Armband.....	8
2.2.2.1	Componentes del brazalete Myo Armband.	9
2.2.2.2	Especificaciones técnicas del brazalete Myo Armband.	10
2.2.2.3	Sensores musculares.....	11
2.2.2.4	Unidad de Medición Inercial (IMU)	12
2.2.2.5	Giroscopio de 3 ejes	12
2.2.2.6	Acelerómetro de tres ejes.	13
2.2.2.7	Magnetómetro de 3 ejes.	13
2.2.2.8	Procesador ARM Cortex M4.	14
2.3	Descripción de los estados de los LEDs del brazalete Myo Armband.....	14
2.4	Salida de datos del brazalete Myo Armband.....	16
2.5	Tarjetas de control electrónico.	17
2.5.1	Arduino.....	17
2.5.2	Arduino UNO.....	18
2.6	Comunicación Bluetooth.....	18
2.6.1	Bluetooth de baja energía.....	19
2.7	Fuente de alimentación.....	21
2.7.1	Tipos de fuentes de alimentación.....	22
2.7.2	Funciones esenciales de una fuente de alimentación lineal.	22
2.8	Motor DC.....	24
2.8.1	Reducciones mecánicas.....	25
2.9	Puente H para control de motores DC.....	26
2.9.1	Driver Puente H L298N.....	27
	CAPITULO III	29
3.1	Preliminares.....	29

3.2	Componentes para el desarrollo del control de movimiento de un brazo robótico.....	31
3.3	Carga del Firmware en el módulo Bluetooth HM-11.....	31
3.4	Desarrollo de los componentes de la maqueta para el control del brazo robótico.....	35
3.4.1	Fuente de alimentación de 5 V y 9 V independientes.....	35
3.4.1.1	Análisis matemático para el dimensionamiento de los capacitores electrolíticos.....	36
3.4.1.2	Materiales utilizados para la fuente de alimentación de 5 V y 9 V.....	39
3.4.1.3	Diseño de la placa de la fuente de alimentación.....	40
3.4.1.4	Grabado de las pistas en la placa de baquelita.....	40
3.4.1.5	Montaje de los componentes en la placa de la fuente de alimentación.....	41
3.4.2	Módulo divisor de voltaje.....	42
3.4.2.1	Materiales para el desarrollo del módulo divisor de voltaje.....	43
3.4.2.2	Diseño de la placa para el módulo del divisor de voltaje.....	43
3.4.2.3	Grabado de las pistas en la placa de baquelita.....	44
3.4.2.4	Montaje de los componentes en la placa del módulo divisor de voltaje.....	44
3.4.3	Módulo de indicación de movimientos.....	45
3.4.3.1	Materiales para el desarrollo del módulo de indicación de movimientos.....	45
3.4.3.2	Diseño de la placa del módulo de indicación de movimientos.....	46
3.4.3.3	Grabado de las pistas en la placa de baquelita utilizando el método del planchado.....	46
3.4.4	Base para el módulo Bluetooth HM-11.....	48

3.4.4.1	Materiales para el desarrollo de la base para el módulo Bluetooth HM-11.	48
3.4.4.2	Diseño de la placa para la base del módulo Bluetooth HM-11.....	48
3.4.4.3	Grabado de las pistas en la placa de baquelita.	49
3.4.4.4	Montaje de los componentes en la placa de la base para el módulo Bluetooth HM-11.	49
3.5	Etapas para el control de movimiento del brazo robótico.	50
3.5.1	Sincronización.	50
3.5.2	Bloqueo / Desbloqueo.	51
3.5.3	Reconocimiento de gestos.....	51
3.6	Estructura general del software.	57
3.7	Diagrama de flujo.....	58
3.8	Programación de la tarjeta Arduino UNO.	59
3.8.1	Declarar librerías.....	59
3.8.2	Declarar puertos.....	59
3.8.3	Declarar variables.....	60
3.8.4	Void printConnectionStatus.	60
3.8.5	Void updateLockOutput.	60
3.8.6	Void updateControls.	61
3.8.7	Void setup.....	62
CAPITULO IV	64
4.1	Conclusiones y recomendaciones	64
4.1.1	Conclusiones.....	64
4.1.2	Recomendaciones.....	66
BIBLIOGRAFÍA	67
GLOSARIO DE TERMINOS	71
ANEXOS	72

ÍNDICE DE FIGURAS

Figura 1 Dispositivo Myo Armband.	8
Figura 2 Componentes del brazalete Myo Armband.	9
Figura 3 Giroscopio de 3 ejes.	13
Figura 4 Acelerómetro de 3 ejes.	13
Figura 5 Magnetómetro de 3 ejes.	14
Figura 6 Leds de indicación del brazalete Myo.	14
Figura 7 Placa Arduino UNO (genérico).....	17
Figura 8 Placa Arduino UNO (original).....	18
Figura 9 Módulo Bluetooth V4.0 HM-11 BLE.	19
Figura 10 Diagrama de bloques de una fuente de alimentación lineal.	22
Figura 11 Partes del motor DC.	25
Figura 12 Motores DC con caja reductora.	25
Figura 13 Puente H con interruptores.	26
Figura 14 Driver Puente H L298N.....	27
Figura 15 Brazo robótico.....	29
Figura 16 Módulo bluetooth HM-11, placa Arduino UNO y el brazalete Myo Armband.....	29
Figura 17 Maqueta para el control del brazo robótico.	30
Figura 18 Diagrama principal de conexiones.	30
Figura 19 Ventana CMD del firmware cargado.	31
Figura 20 Diagrama de conexiones del módulo Bluetooth HM-11 con la placa Arduino UNO.	32
Figura 21 Ventana CMD.	33
Figura 22 Comando de dirección del MyoBridge_CC2541.....	33
Figura 23 Información del comando de dirección del MyoBridge_CC2541. .	34
Figura 24 Tiempo de espera para habilitar la transmisión.	34
Figura 25 Carga completa del firmware.	35
Figura 26 Fuente de alimentación de 5 V y 9 V.	36
Figura 27 Diagrama de las etapas de la fuente de voltaje.....	36
Figura 28 Diseño de la fuente de alimentación de 5 V y 9 V en el software Proteus.	40

Figura 29 Grabado de las pistas de la fuente de alimentación de 5 V y 9 V.	41
Figura 30 Montaje de los componentes en la placa de la fuente de alimentación.	41
Figura 31 Esquema de conexión del Bluetooth HM-11.	42
Figura 32 Esquema del divisor de voltaje.	42
Figura 33 Diseño del divisor de voltaje en el software Proteus.	43
Figura 34 Grabado de las pistas del divisor de voltaje.	44
Figura 35 Montaje de los componentes en la placa del módulo divisor de voltaje.	44
Figura 36 Diseño del módulo de indicación de movimientos en el software Proteus.	46
Figura 37 Aplicación de calor para la transferencia de las pistas a la placa de baquelita.	47
Figura 38 Recipiente plástico con ácido férrico.	47
Figura 39 Módulo indicador de movimientos.	48
Figura 40 Diseño de la base para el módulo Bluetooth HM-11 en el software Proteus.	49
Figura 41 Grabado de las pistas de la base para el módulo Bluetooth HM-11.	49
Figura 42 Módulo Bluetooth HM-11 montado en la base.	50
Figura 43 Diagrama de bloques de las etapas para el control de movimiento del brazo robótico.	50
Figura 44 Gesto de puño.	51
Figura 45 Luz de indicación del gesto de bloqueo / desbloqueo.	51
Figura 46 Indicador de estado de lectura.	52
Figura 47 Visualización del gesto “izquierda” y el porcentaje de batería en el monitor serial del Software Arduino.	53
Figura 48 Módulo de indicación de movimientos.	53
Figura 49 Bloques del módulo indicador de movimiento.	54
Figura 50 Indicador de movimiento en rojo1.	54
Figura 51 Indicador de movimiento en rojo2.	54
Figura 52 Indicador de movimiento en amarillo1.	55
Figura 53 Indicador de movimiento en amarillo2.	55
Figura 54 Indicador de movimiento en verde1.	55

Figura 55 Indicador de movimiento en verde2.	56
Figura 56 Indicador de movimiento en naranja1.	56
Figura 57 Indicador de movimiento en naranja2.	56
Figura 58 Diagrama de bloques de la estructura general del software.....	57
Figura 59 Diagrama de flujo del programa principal.....	58

ÍNDICE DE TABLAS

Tabla 1 Áreas de aplicación de las señales EMG.	6
Tabla 2 Gestos prediseñados del brazalete Myo Armband.	9
Tabla 3 Especificaciones técnicas del brazalete Myo Armband.	10
Tabla 4 Estados de la batería del brazalete Myo Armband.	15
Tabla 5 Estados de conexión del brazalete Myo Armband.	15
Tabla 6 Características del Módulo Bluetooth V4.0 HM-11 BLE.	20
Tabla 7 Características eléctricas del Módulo Bluetooth V4.0 HM-11 BLE. .	20
Tabla 8 Pines del Módulo Bluetooth V4.0 HM-11 BLE.	21
Tabla 9 Reguladores fijos con su tensión de salida.	23
Tabla 10 Estados del puente H.	26
Tabla 11 Movimientos y acción del brazo robótico.	45

ÍNDICE DE ECUACIONES

Ecuación 1: Tensión antes del puente de diodos.	37
Ecuación 2: Etapa de rectificación.	37
Ecuación 3: Etapa de filtrado.	38
Ecuación 4: Dimensionamiento del capacitor para una corriente de 500 mA.	38
Ecuación 5: Dimensionamiento del capacitor para una corriente de 1 A.	39
Ecuación 6: Fórmula para el divisor de voltaje.	43

RESUMEN

El presente trabajo de titulación está enfocado hacia la implementación de un control de movimiento de un brazo robótico, mediante señales electromiográficas y haciendo uso de la tecnología Myo Armband, el mismo que servirá como material didáctico para el laboratorio de instrumentación virtual y de esta manera poder contribuir el mejoramiento del proceso de enseñanza y aprendizaje.

El control del brazo robótico se realizó haciendo uso del Brazaletes Myo Armband, el cual posee ocho sensores mioeléctricos que se posicionan alrededor del antebrazo, y que consta de un acelerómetro, un giroscopio y un magnetómetro. Dicho dispositivo nos permite obtener datos correspondientes a los tres ejes (X, Y y Z); los mismos que son generados por medio de impulsos eléctricos de acuerdo con el movimiento del antebrazo, para luego ser enviados por medio de un dispositivo Bluetooth V4.0 HM-11 hacia una tarjeta de control.

Con la realización de este proyecto se abre un sin número de posibilidades para diferentes aplicaciones, en este caso se optó por implementarlo en el control de movimiento de un brazo robótico, el cual no exige de mucho esfuerzo muscular por parte del usuario, por lo que se usa para indicar el inicio y el fin de la lectura de gestos, los mismos que son traducidos en movimientos preestablecidos en el programa principal.

PALABRAS CLAVE:

ELECTROMIOGRAFÍA

MYO ARMBAND

INSTRUMENTACIÓN VIRTUAL

BRAZO ROBÓTICO

IMPULSOS ELÉCTRICOS

ABSTRAC

The present research is focused towards the implementation of a motion control of a robotic arm, using electromyographic signals and making use Myo Armband technology, the same that will serve as didactic material for the laboratory virtual instrumentation and this way to be able contribute to the improvement of the teaching and learning process.

The control of the robotic arm was made using the Myo Armband Bracelet, which has eight myoelectric sensors that are positioned around the forearm, and consists of an accelerometer, a gyroscope and a magnetometer. This device allows us to obtain data corresponding to the three axes (X, Y and Z), which are generated by means of electrical impulses according to the movement of the forearm, to be sent by means of a Bluetooth V4.0 HM-11 device to a control card.

With the realization of this research opens a number of possibilities for different applications, in this case it was decided to implement it in the motion control of a robotic arm, which does not require much muscle effort by the user, so that it is used to indicate the beginning and end of the reading of gestures, the same ones that are translated in pre-established movements in the main program.

KEYWORDS:

ELECTROMYOGRAPHY

MYO ARMBAND

VIRTUAL INSTRUMENTATION

ROBOTIC ARM

ELECTRICAL IMPULSES

CHECKED BY:

**MARÍA ELISA COQUE
ENGLISH TEACHER UGT**

CAPÍTULO I

1.1 Antecedentes

Con el pasar del tiempo el desarrollo de la tecnología ha ido avanzando considerablemente y la creación de herramientas tecnológicas nos ha llevado a un mundo donde el límite es la imaginación, por lo que se necesita soluciones a procesos automatizados, económicos y la menor intervención humana posible.

En el auge de la tecnología, en el campo de la robótica y electrónica se han desarrollado diversos dispositivos a nuestra disposición, así como el uso de ordenadores personales dotados del hardware y software necesarios para monitorear y controlar los procesos.

Mediante la electromiografía se han realizado trabajos como los que se exponen a continuación:

Según Mithileysh Sathiyarayanan (de la universidad de Brighton, Reino Unido) y Sharanya Rajan (del Hospital Fortis, Bangalore), a través de su trabajo de investigación científica (paper): “BRAZALETE MYO PARA FISIOTERAPIA SANITARIA: UN CASO DE ESTUDIO MEDIANTE LA APLICACIÓN DE RECONOCIMIENTO DE GESTOS”, en donde se especifica algunas características del Brazalete Myo.

“El MYO está equipado con varios sensores que pueden reconocer la mano gestos y el movimiento de los brazos. Se caracteriza por usar un proceso llamado electromiografía (EMG); identificando el gesto del moviendo de los músculos del brazo. Basado en los impulsos eléctricos generados por los músculos, 8 sensores EMG son responsables de reconocer y realizar cada gesto.” (Sathiyarayanan & Rajan, 2016)

Un segundo trabajo corresponde a Federico Gaetani, Giovanni Antonio Zappatore (ambos de BionIT Labs Company), Paolo Visconti y Patrizio Primiceri (del Departamento de Ingeniería de Innovación, Universidad de Salento Italia), a través de su trabajo de investigación científica (paper): “DISEÑO DE UNA PLATAFORMA BASADA EN ARDUINO CON INTERFAZ BLUETOOTH LOW ENERGY Y BRAZALETE MYO PARA EL CONTROL DE UNA PRÓTESIS TRANSRADIAL INFRA-ACCIONADA”, en donde se especifica algunas características de la prótesis.

“El dispositivo está equipado con sensores y actuadores que simplifican y ayudan a los movimientos de la mano; en particular, el mecanismo en el que se basa puede activar 15 grados de libertad con un solo motor (en lugar de los cinco / seis motores utilizados convencionalmente en otros dispositivos protésicos disponibles comercialmente).” (Gaetani, Zappatore, Visconti, & Primiceri, 2018)

“Dos servomotores se utilizan para activar los movimientos de la muñeca. Las señales mioeléctricas utilizadas para controlar la prótesis se detectan a través del brazalete Myo, que integra ocho electrodos de Electromiografía (EMG) y una Unidad de Medición Inercial (IMU).” (Gaetani, Zappatore, Visconti, & Primiceri, 2018)

Un tercer trabajo corresponde a Diego Camacho, María Erazo, Erick Mera, Álvaro Velasco (todos del Departamento de Energía y Mecánica, Universidad de las Fuerzas Armadas ESPE), a través de su trabajo de investigación científica (paper): “CONTROL DE UN BRAZO ROBÓTICO LYNX AL5D EMPLEANDO ELECTROMIOGRAFÍA A TRAVÉS DE LA TECNOLOGÍA MYO ARMBAND”, en donde se especifica algunas características de la tecnología Myo ARMBAND.

“Al realizar un gesto, el cual es interpretado por el brazalete y enviado inalámbricamente por Bluetooth al computador, este es interpretado por MYODUINO quien se encarga de enlazar la señal entre MYO ARMBAND y Arduino que envía un pulso al servo correspondiente permitiendo el control de

la posición del Brazo Robótico en todos sus grados de libertad.” (Diego Camacho, 2016)

1.2 Planteamiento del problema

Actualmente en los laboratorios de la carrera de electrónica de la Unidad de Gestión de Tecnologías ESPE sede Latacunga existe una cantidad limitada de unidades didácticas de alta tecnología dedicadas a simular procesos mediante tareas de programación, comunicación y control. Los módulos existentes implementados resultan insuficientes para que todos los alumnos de la carrera puedan familiarizarse y manipular nuevas tecnologías y de esa manera poder acondicionar la señal y tratarla de manera que se pueda manipular como un proceso real.

De ahí que la implementación de nuevas tecnologías en la universidad es prácticamente nula. Sin embargo, esto no es impedimento para que se realice la implementación de esta tecnología dentro de las universidades.

El desarrollo de este proyecto servirá como material de apoyo para futuras prácticas, que desarrollen los estudiantes de la carrera de Electrónica Mención Instrumentación y Aviónica, en el laboratorio de instrumentación virtual, ayudando así a desarrollar las habilidades, destrezas y actitudes para, analizar los principios básicos de la automatización y control de procesos para poner en práctica en el campo laboral.

1.3 Justificación e importancia

Este proyecto de grado tiene la finalidad de contribuir al desarrollo de prácticas de los últimos niveles de la carrera de Electrónica mención Instrumentación y Aviónica, en lo referente a la implementación y el control del brazalete MYO como ayuda prioritaria dentro del desarrollo de los conocimientos, ya que los nuevos dispositivos tecnológicos encierran muchos tópicos y diferentes campos de aplicación entre ella la industria.

Es necesario que el laboratorio de instrumentación virtual de la Unidad de Gestión de Tecnologías ESPE cuente con dispositivos de última tecnología, para que de esta manera los estudiantes de la carrera de Electrónica Mención Instrumentación y Aviónica obtengan conocimientos teóricos y prácticos de una nueva tecnología para que así puedan incrementar los conocimientos tanto prácticos como teóricos.

1.4 Objetivos

1.4.1 General

Diseñar e implementar un control de movimiento de un brazo robótico mediante electromiografía a través de la tecnología MYO ARMBAND para el laboratorio de instrumentación virtual.

1.4.2 Específicos

- Analizar las características técnicas del brazalete MYO ARMBAND.
- Realizar una interfaz entre el brazalete Myo Armband y un módulo bluetooth V4.0 para la transmisión y recepción de datos.
- Desarrollar una programación en el software Arduino para realizar el control de movimiento del brazo robótico.
- Realizar las pruebas de funcionamiento para evaluar el comportamiento del brazo robótico.
- Construir e implementar una maqueta para el control de movimiento del brazo robótico.

1.5 Alcance

Este proyecto técnico está dirigido a los estudiantes, de la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías ESPE, con el principal propósito de incentivar a mejorar el nivel de conocimiento, en donde se realiza un estudio del funcionamiento del brazalete Myo Armband, para tener en consideración información relevante para el desarrollo del proyecto.

El brazalete Myo Armband puede conectarse a dispositivos que dispongan de comunicación bluetooth V4.0 o superior, para efectos de este proyecto se estableció una interfaz de comunicación directa entre el brazalete Myo Armband y la tarjeta controladora Arduino UNO mediante el uso del bluetooth V4.0 HM-11 de bajo consumo de energía, de modo que sea capaz de recibir todas las señales captadas por los 8 sensores mioeléctricos que posee el brazalete Myo Armband, y transmitir las al microcontrolador.

Myo Armband posee 5 movimientos o gestos preestablecidos, que son captados por los 8 sensores mioeléctricos, además utiliza un magnetómetro de 3 ejes, un acelerómetro de 3 ejes y un giróscopo de 3 ejes que permite detectar el movimiento del brazo en cualquier dirección. Con las señales obtenidas del brazalete Myo Armband, se procede a establecer patrones para un control de movimiento de un brazo robótico, para que estos puedan ser traducidos a movimientos específicos; para lo cual se genera un algoritmo de control en la plataforma Arduino, que utilice los datos obtenidos y permita el manejo de un sistema de motores DC que son los encargados de dar movimiento al brazo robótico.

CAPÍTULO II

2.1 Electromiografía

Los músculos del brazo reciben los comandos para el movimiento desde el cerebro a través de impulsos eléctricos denominados acciones de potencial. La medición de estas señales eléctricas se conoce como Electromiografía (de electo, electricidad; mío, músculo; y grafía, escritura), abreviado usualmente como EMG. (Silver, 2015)

La electromiografía (EMG) es un procedimiento de diagnóstico que se utiliza para evaluar la salud de los músculos y las neuronas que los controlan (neuronas motoras). Las neuronas motoras transmiten señales eléctricas que hacen que los músculos se contraigan. Una EMG convierte estas señales en gráficos, sonidos o valores numéricos que interpreta un especialista. (Cardona, 2018)

2.1.1 Beneficio de las señales EMG

Las señales EMG no solo están dedicadas al estudio fisiológico, también han sido aplicadas para la investigación, rehabilitación, entrenamiento deportivo e interacción con el ser humano. (Ivan, 2017) En la tabla 1 se muestran las áreas de aplicación de las señales EMG.

Tabla 1

Áreas de aplicación de las señales EMG

Investigación médica	Rehabilitación	Ergonomía	Ciencia del Deporte
Ortopedia	Post cirugía / accidentes	Análisis de demanda	Biomecánica
Cirugía	Rehabilitación neurológica	Prevención de riesgos	Análisis de movimientos

Neurología	Terapia física	Diseños ergonómicos	Entrenamiento de atletas
Análisis de postura	de Terapia de entrenamiento	de Certificación de productos	Deportes de rehabilitación

Fuente: (Ivan, 2017)

2.1.2 Origen de señales

Las señales de electromiografía (EMG), son generadas por el intercambio de iones a través de las membranas de las fibras musculares debido a una contracción muscular. (DALCAME, 2018)

Lo que la gente considera normalmente un músculo (por ejemplo, el bíceps de los brazos) consta de miles de células individuales, envueltas en tejido conectivo. Como las células musculares tienen una forma elongada, con frecuencia se llaman fibras. Las fibras musculares están dispuestas en haces llamados fascículos, que se encuentran envueltos en tejido conectivo. (DALCAME, 2018)

2.2 Elementos de control del brazo robótico

2.2.1 Sensores mioeléctricos

Para el control del brazo robótico se utilizan sensores portátiles que capten las señales mioeléctricas (EMG) de las extremidades, en este caso señales emitidas por los músculos del brazo, para luego accionar cierto dispositivo que proporciona el movimiento del brazo robótico.

Las señales EMG pueden obtenerse mediante el uso de electrodos superficiales, los cuales poseen la ventaja de no ser invasivos, pero presenta inconvenientes con otros factores tales como el ruido e interferencia, características de la piel (sudoración), variación de la posición del electrodo respecto al músculo considerado y así como cambios en la distancia, razones por las que se puede alterar las señales EMG. (Jorge Brazeiro, 2015)

Estas desventajas se pueden disminuir con el uso del brazalete Myo Armband, que es un dispositivo que capta con una mayor exactitud las señales mioeléctricas de los músculos del brazo.

2.2.2 Myo Armband

Myo es un dispositivo háptico desarrollado por Thalmic Labs en Canadá, para el reconocimiento del gesto de mano y el seguimiento de movimiento del brazo. Consta con ocho sensores de alta sensibilidad, que tienen la capacidad de reconocer e interpretar la actividad eléctrica que producen los músculos, y las cuales se traducen en instrucciones concretas para controlar una interfaz sin necesidad de cables, ya que la comunicación es vía bluetooth. (ThalmicLabs, Myo SDK, 2013)

El objetivo de Myo Armband es controlar de forma fácil y cómoda una PC, teléfonos inteligentes, juguetes y otros productos mediante el uso de gestos con las manos. Su principal ventaja es que no necesita cámara, además está equipado con un procesador ARM Cortex M4, un sistema de feedback háptico basado en vibraciones, conectividad bluetooth, indicadores led y cuenta con una batería de litio que se carga a través de micro USB. En la Figura 1, se muestra el aspecto físico del dispositivo Myo.



Figura 1 Dispositivo Myo Armband

Fuente: (ThalmicLabs, Myo SDK, 2013)

El brazalete Myo Armband, cuenta con un conjunto de gestos prediseñados que no se pueden cambiar debido a algoritmos de aprendizaje automático, estos cinco gestos se detallan en la tabla 2.

Tabla 2

Gestos prediseñados del brazalete Myo Armband

Nombre del gesto	Gesto	Movimiento	Acción muscular
Puño		Flexión de mano	Flexores de mano
Palma hacia adentro		Flexión de muñeca	Flexores de muñeca
Palma hacia afuera		Extensión de muñeca	Extensores de muñeca
Dedos extendidos		Extensión y abducción de dedos	Abducción de dedos
Doble pulso		Oposición del pulgar	Músculos del pulgar

Fuente: (ThalmicLabs, 2013)

2.2.2.1 Componentes del brazalete Myo Armband

En la figura 2, se muestran los principales componentes del Myo Armband.

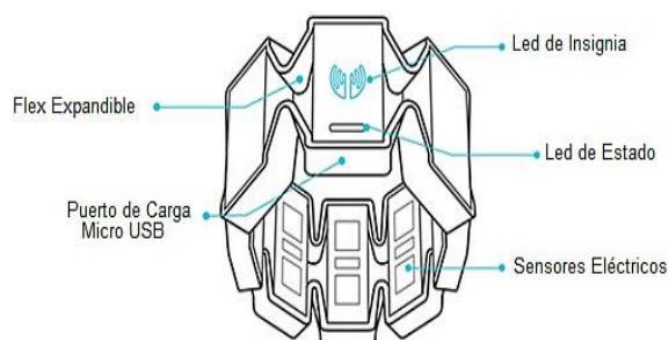


Figura 2 Componentes del brazalete Myo Armband

Fuente: (ThalmicLabs, Myo SDK, 2013)

Los sensores eléctricos se encuentran en los segmentos de la carcasa extensible, que permite que se expanda y contraiga con respecto al otro, de modo que el Myo puede encajar cómodamente en la fisiología de cada usuario. Los sensores eléctricos miden las señales eléctricas que viajan a través del brazo del usuario y el brazalete los traduce en poses y gestos. (ThalmicLabs, Myo SDK, 2013)

2.2.2.2 Especificaciones técnicas del brazalete Myo Armband

En la tabla 3 se muestra un resumen de las especificaciones técnicas del brazalete Myo Armband.

Tabla 3

Especificaciones técnicas del brazalete Myo Armband

Dimensionamiento, peso y dimensiones	Tamaño	Expansible entre 7.5-13 pulgadas (19-34 centímetros) circunferencia del antebrazo.
	Peso	93 gramos
	Espesor	0.45 pulgadas (1.14 centímetros).
Dispositivos compatibles	Windows	Windows 7 y superior (con adaptador USB Bluetooth incluido y OpenGL 2.1 o superior).
	Mac OS X y MacOS	Mac OS X 10.8 y superior (con adaptador USB Bluetooth incluido).
	iOS	iOS 7.0 y superior.
	Androide	Android 4.3 (Jelly Bean) y superior. El dispositivo debe tener soporte de radio Bluetooth para Bluetooth 4.0 Low Energy.

Hardware	Sensores	-Sensores EMG de acero inoxidable de grado médico. -IMU de nueve ejes altamente sensible que contiene giroscopio de tres ejes, acelerómetro de tres ejes, magnetómetro de tres ejes.
	LEDs	LED indicador doble.
	Procesador	Procesador ARM Cortex M4.
	Retroalimentación Háptica	Vibraciones cortas, medias, largas.
	Comunicación	Bluetooth inteligente
Potencia y batería	Cargando	Micro USB
	Batería	Batería de iones de litio recargable incorporada.
	Actuación	Un día completo de uso en una sola carga.

Fuente: (ThalmicLabs, 2013)

2.2.2.3 Sensores musculares

Para la detección de los gestos de mano el dispositivo Myo Armband utiliza tecnología basada en la electromiografía (EMG). Los músculos de los dedos están unidos al codo y este hecho, es utilizado por el dispositivo Myo Armband para detectar los movimientos. Myo Armband fue diseñado para ser colocado en el antebrazo, donde 8 sensores EMG especiales supervisan la actividad eléctrica de estos músculos que reaccionan al realizar un determinado gesto con la mano. (ThalmicLabs, Myo SDK, 2013)

2.2.2.4 Unidad de Medición Inercial (IMU)

El dispositivo Myo Armband está equipado con una unidad de medición inercial (IMU) que detecta el movimiento del brazo. La unidad de medición inercial (IMU) contiene un giroscopio de tres ejes, un acelerómetro de tres ejes y un magnetómetro de tres ejes.

Una Unidad de Medición Inercial o IMU (por las siglas en inglés de Inertial Measurement Unit) es un dispositivo electrónico cuyo objetivo es obtener mediciones de velocidad, rotación y fuerzas gravitacionales en forma autónoma. (Maureira, 2010)

Típicamente, una IMU está compuesta por un conjunto de acelerómetros y giróscopos, que obtienen datos de uno o más ejes ortogonales, enviándolos a algún sistema computarizado que realiza los cálculos necesarios para obtener las estimaciones de aceleración y velocidad de rotación requeridas. (Maureira, 2010)

2.2.2.5 Giroscopio de 3 ejes

El giroscopio es un dispositivo que sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo. Se basan en el principio de Coriolis; lo que dice este principio es que, si un cuerpo está girando y nosotros estamos en el centro queriendo salir en línea recta al exterior, tendremos que ir hacia adelante y al mismo tiempo de forma lateral para contrarrestar esta rotación. (Aran, 2012)

Un giroscopio electrónico emplea una tecnología microelectromecánica (MEMS) con un sensor de silicio, gracias a unas placas especiales que actúan a modo de condensadores. Los cambios de intensidad se convierten en datos digitales y eso es lo que lee el microprocesador.

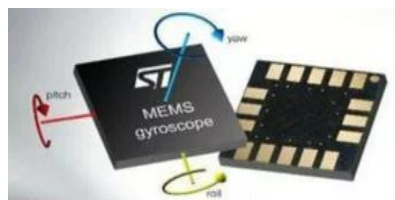


Figura 3 Giroscopio de 3 ejes.

Fuente: (Aran, 2012)

2.2.2.6 Acelerómetro de tres ejes

Los acelerómetros son dispositivos que miden la aceleración, que es la tasa de cambio de la velocidad de un objeto, funciona de una forma electromecánica a nivel microscópico, precisamente estos dispositivos son llamados MEMS (Microelectromechanical Systems) que es una tecnología de dispositivos microscópicos con partes mecánicas movibles. Los acelerómetros detectan las fuerzas de aceleración, ya sea estática o dinámica. Las fuerzas estáticas incluyen la gravedad, mientras que las fuerzas dinámicas pueden incluir vibraciones y movimiento. (5hertz., 2014)

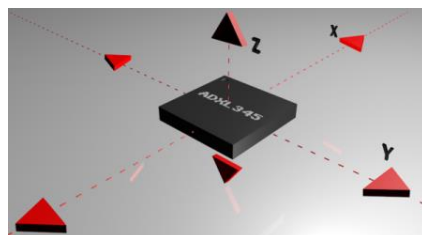


Figura 4 Acelerómetro de 3 ejes.

Fuente: (5hertz., 2014)

2.2.2.7 Magnetómetro de 3 ejes

El magnetómetro es un componente electrónico capaz de medir y cuantificar la cantidad de fuerza magnética de un objeto. O para lo que muchos dispositivos lo usan, como brújula, detectando el polo norte magnético (que como curiosidad no coincide con el polo norte geográfico). (W., 2015)



Figura 5 Magnetómetro de 3 ejes

Fuente: (5hertz., 2014)

2.2.2.8 Procesador ARM Cortex M4.

Es un procesador embebido de alto rendimiento para control de señales digitales de alta eficiencia con bajo consumo de energía y de bajo costo. Se caracteriza principalmente por proporcionar capacidades para el procesamiento de señal digital (DSP), funcionalidad matemática y una unidad opcional de punto flotante (FPU). (developer, 2016)

2.3 Descripción de los estados de los LEDs del brazalete Myo Armband

El dispositivo Myo Armband tiene dos LEDs (Figura 6), que representan el estado actual del brazalete. En la tabla 4 se detalla los diversos estados de la batería. (ThalmicLabs, 2013)

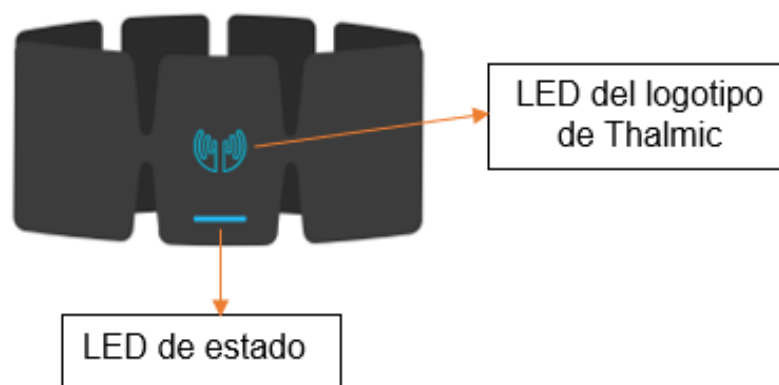


Figura 6 Leds de indicación del brazalete Myo

Fuente: (ThalmicLabs, 2013)

Tabla 4**Estados de la batería del brazalete Myo Armband**

Descripción	LED del logotipo de Thalmic	LED de estado
El dispositivo Myo Armband está conectado a una fuente de alimentación y se está cargando.	Apagado	Naranja pulso
El dispositivo Myo Armband está conectado a una fuente de alimentación y está completamente cargado.	Apagado	Verde constante
Muestra el indicador de batería baja cuando el dispositivo Myo Armband está en uso o inactivo.	Azul constante	Flash naranja
Muestra la batería del dispositivo Myo Armband está al 0%. Conectar a una fuente de alimentación.	Apagado	Apagado

Fuente: (ThalmicLabs, 2013)

En la tabla 5 se detalla los diversos estados de la conexión con otros dispositivos.

Tabla 5**Estados de conexión del brazalete Myo Armband**

Descripción	LED del logotipo de Thalmic	LED de estado
Muestra que el dispositivo Myo Armband está conectado por Bluetooth y se ha realizado el gesto de sincronización.	Azul constante	Azul constante
Muestra que el dispositivo Myo Armband está conectado por Bluetooth y no se ha realizado el gesto de sincronización.	Pulso azul	Azul constante

Muestra que el dispositivo Myo Armband no está conectado por Bluetooth y se ha realizado el gesto de sincronización. Myo no entrará en modo de suspensión mientras esté sincronizado.	Azul constante	Apagado
Muestra que el dispositivo Myo Armband no está conectado por Bluetooth y no se ha realizado el gesto de sincronización. Myo entrará en el modo de suspensión después de un corto período de tiempo.	Pulso azul	Apagado
Muestra que el dispositivo Myo Armband se está calentando. Vibrará cuando termine el período de calentamiento.	Flash azul	Azul constante

Fuente: (ThalmicLabs, 2013)

2.4 Salida de datos del brazalete Myo Armband

El dispositivo Myo Armband ofrece dos tipos de datos en la aplicación, los datos espaciales y los datos gestuales. (ThalmicLabs, Myo SDK, 2013)

- **Datos espaciales**

Los datos espaciales dan información de la orientación y movimientos del antebrazo. La UMI de MYO genera 9 ejes de datos espaciales que se obtienen con el SDK. Los datos obtenidos son: Un vector de orientación, que indica como Myo está localizado en relación con la flexión de hombro (pitch), aducción-abducción de hombro (yaw) y pronación-supinación de codo (roll). Un vector de aceleración del acelerómetro. Un vector de velocidad angular del giroscopio. (HUITZIL-VELASCO, 2017)

- **Datos gestuales**

Indica a la aplicación lo que el usuario está haciendo con sus manos. Proporciona datos gestuales en forma de una de las varias poses preestablecidas que representan una configuración particular de la mano del usuario. Por ejemplo, una pose representa la mano haciendo un puño, mientras que otra representa la mano estando en reposo con la palma abierta. (ThalmicLabs, Myo SDK, 2013)

2.5 Tarjetas de control electrónico

Son plataformas electrónicas basadas en hardware y software sencillos de usar; la plataforma basada en una placa de entradas y salidas simple y un entorno de programación libre es Arduino.

2.5.1 Arduino

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Consta de una placa principal de componentes eléctricos, donde se encuentran conectados los controladores principales que gestionan los demás complementos y circuitos ensamblados en la misma. Además, requiere de un lenguaje de programación para poder ser utilizado, programado y configurarlo de acuerdo con la necesidad del usuario. (Marmolejo, 2018)



Figura 7 Placa Arduino UNO (genérico)

Fuente: (Marmolejo, 2018)

2.5.2 Arduino UNO

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328 (Figura 8). Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa de Arduino UNO se puede conectar a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador. (JADIAZ, 2016)



Figura 8 Placa Arduino UNO (original)

Fuente: (JADIAZ, 2016)

2.6 Comunicación Bluetooth

Es un estándar global de comunicación inalámbrica, que posibilita la transmisión de datos entre diferentes equipos móviles y fijos, sin conexión a cables y conectores, utiliza la técnica FHSS (Frequency Hopping Spread Spectrum, espectro ensanchado por saltos de frecuencia). (Francisco José Bellido Outeiriño, 2014)

La conexión entre dispositivos Bluetooth se basa en el modo de operación maestro/esclavo, soportando hasta ocho dispositivos conectados simultáneamente en lo que es la estructura básica de comunicación, denominada piconet o picorred. El término piconet se utiliza para hacer

referencia a la red formada por un dispositivo y todos los dispositivos que se encuentran dentro de su rango.

Para recibir los datos del Myo Armband, se requiere establecer la comunicación bluetooth mediante dispositivo bluetooth de baja energía.

2.6.1 Bluetooth de baja energía

El bluetooth BLE (Bluetooth Low Energy), se presentó en la versión 4.0 en el año 2010, utiliza una pila simple que permite una comunicación asíncrona con dispositivos de bajo consumo, como sensores inalámbricos que envían volúmenes bajos de datos a intervalos infrecuentes.

El Módulo Bluetooth V4.0 HM-11 BLE, (Figura 9), se basa en el microprocesador CC2541, se puede usar para comunicarse con computadoras de escritorio, teléfonos inteligentes, computadoras de mesa e incluso otros sistemas integrados que ofrecen Bluetooth Smart, también conocido como Bluetooth Low Energy (BLE). (Donovan, 2011)



Figura 9 Módulo Bluetooth V4.0 HM-11 BLE.

Fuente: (Donovan, 2011)

En la tabla 6 se muestran las características del Módulo Bluetooth V4.0 HM-11 BLE.

Tabla 6**Características del Módulo Bluetooth V4.0 HM-11 BLE**

Características	Descripción
Protocolo Bluetooth	Especificación Bluetooth V4.0 BLE
Frecuencia de trabajo	Banda de 2,4 GHz ISM
Modo de interfaz	Un puerto serie
Entorno abierto	A 30 metros puede realizar comunicación entre módulos.
Enviar y recibir	Sin límite de bytes entre módulos.
Método de modulación	GFSK (codificación de cambio de frecuencia gaussiana).
Potencia de transmisión	- DBM, 23-6 DBM, 0 DBM, 6 DBM, puede ser modificado por el comando AT.
Microprocesador	TI CC2541, espacio de configuración de 256 KB.
Fuente de alimentación	+ 3.3 VDC 50 mA
Temperatura de trabajo	- 5 ~ + 65 grados centígrados

Fuente: (SEEDSTUDIO, 2017)

En la tabla 7 se muestran las características eléctricas del Módulo Bluetooth V4.0 HM-11 BLE.

Tabla 7**Características eléctricas del Módulo Bluetooth V4.0 HM-11 BLE**

Especificación	Min	Tipo	Max	Unidad
Voltaje de entrada máximo	-0.3		3.6	V
Voltaje de entrada de trabajo	2.0	3.3	3.6	V
Corriente de transmisión		15		mA
Recibir corriente		8.5		mA
Corriente de sueño profundo		600		uA
Temperatura de funcionamiento	-40		+65	DO

Fuente: (SEEDSTUDIO, 2017)

En la tabla 8 se muestran los pines del Módulo Bluetooth V4.0 HM-11 BLE.

Tabla 8

Pines del Módulo Bluetooth V4.0 HM-11 BLE

# Pin	Nombre	Descripción	CC254X
1	UART_RTS	UART Interface	P1_5
2	UART_TX	UART Interface	P1_6
3	UART_CTS	UART Interface	P1_4
4	UART_RX	UART Interface	P1_7
5	NC	NC	P1_1
6	NC	NC	P1_2
7	NC	NC	PIN2
8	NC	NC	PIN2
9	VCC	3.3 V	VCC
10	NC	NC or Vcc	
11	RESETB	Reset, in flow <100ms.	RESET_N
12	GND	Ground	GND
13	PIO3	Input/output line	P1_1
14	PIO2	Input/output line	P1_0
15	PIO1	System LED	P0_7
16	PIO0	System Key	P0_6

Fuente: (SEEDSTUDIO, 2017)

2.7 Fuente de alimentación

La fuente de poder o de alimentación, es el dispositivo que se encarga de transformar la corriente alterna de la línea eléctrica comercial que se recibe en los domicilios (110 volts en Ecuador) en corriente continua o directa; que es la que utilizan los dispositivos electrónicos, suministrando los diferentes voltajes requeridos por los componentes, incluyendo usualmente protección frente a eventuales inconvenientes en el suministro eléctrico, como la sobretensión. (Concepto.de, 2017)

2.7.1 Tipos de fuentes de alimentación

- **Fuentes lineales:** Utilizan un transformador para disminuir el nivel de tensión en la red eléctrica (reductor de tensión), rectificador (conversión de voltaje alterno a onda completa), filtro (conversión de onda completa a continua) y regulación (mantenimiento del voltaje de salida ante variaciones en la carga). (Comcepto.de, 2017)
- **Fuentes conmutativas:** Estas, en cambio, utilizan circuitos basados en transistores de potencia y bobinas trabajando en conmutación para reducir la tensión. (Comcepto.de, 2017)

2.7.2 Funciones esenciales de una fuente de alimentación lineal

Una fuente de alimentación lineal consta de varias funciones: transformación, rectificación, filtrado, regulación. En la figura 10 se puede observar un diagrama de bloques de una fuente de alimentación lineal.

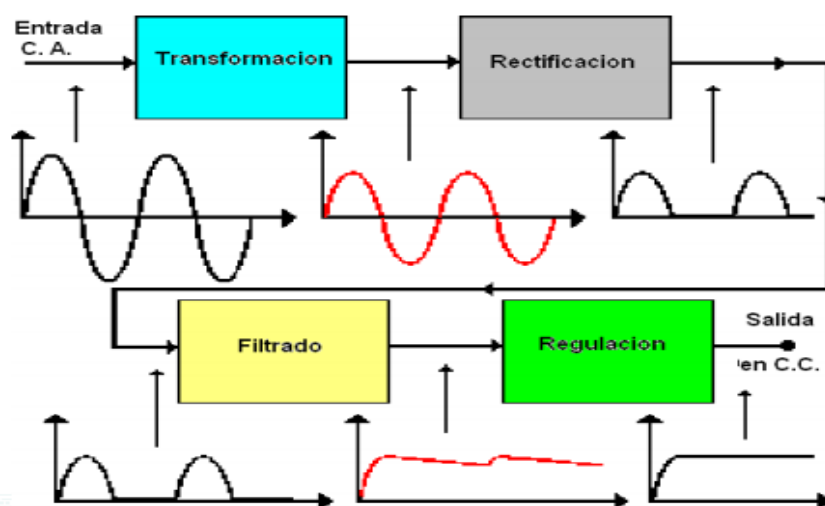


Figura 10 Diagrama de bloques de una fuente de alimentación lineal

Fuente: (Ramírez, 2018)

- **Transformación:** Reduce la tensión de entrada a la fuente (220 V o 120 V), que son las que suministra la red eléctrica. Allí participa un

transformador en bobina. La salida de este proceso generará de 5 a 12 voltios, dependiendo del transformador. (Comcepto.de, 2017)

- **Rectificación:** Esta etapa queda constituida por diodos rectificadores cuya función es de rectificar la señal proveniente del bobinado secundario del transformador. (Ramírez, 2018)
- **Filtrado:** Esta etapa está constituida por uno o varios capacitores que se utilizan para eliminar la componente de tensión alterna que proviene de la etapa de rectificación. Los capacitores se cargan al valor máximo de voltaje entregado por el rectificador y se descargan lentamente cuando la señal pulsante desaparece. Permitiendo lograr un nivel de tensión lo más continua posible. (Ramírez, 2018)
- **Regulación:** Esta etapa se encarga de reducir el rizado y de estabilizar la salida exacta a la que deseemos. De manera general existen dos tipos de reguladores de voltaje: reguladores fijos y reguladores variables.
 - **Reguladores fijos:** En la mayoría de las aplicaciones se requiere una tensión fija y estable de un determinado valor. La línea de reguladores ideales para este tipo de necesidades es la conocida como LM78XX. Las primeras letras y dos números corresponden a la denominación, mientras que las dos últimas XX deben ser reemplazados por la tensión de salida requerida. (Molina, 2018)

Las tensiones disponibles de observan en la tabla 9.

Tabla 9

Reguladores fijos con su tensión de salida

Número	Tensión de salida
LM7805	5 Voltios
LM7806	6 Voltios

LM7808	8 Voltios
LM7812	12 Voltios
LM7815	15 Voltios
LM7818	18 Voltios
LM7824	24 Voltios
LM7830	30 Voltios

Fuente: (Molina, 2018)

-Reguladores Variables: Permite manipular la salida de voltaje de acuerdo con las necesidades del usuario, siendo el regulador LM317 el más conocido.

El regulador de voltaje LM317 tiene la característica de tener una configuración variable, que es controlada mediante una resistencia variable. (E-Marmolejo., 2017)

2.8 Motor DC.

El motor de corriente continua (motor DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio. En algunas modificaciones, ejercen tracción sobre un riel. Estos motores se conocen como motores lineales. (Dea, 2015)

Una máquina de corriente continua (generador o motor) se compone principalmente de dos partes, un estator que da soporte mecánico al aparato y tiene un hueco en el centro generalmente de forma cilíndrica. En el estator además se encuentran los polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente de forma cilíndrica, también devanado y con núcleo, al que llega la corriente mediante dos escobillas. (Dea, 2015)

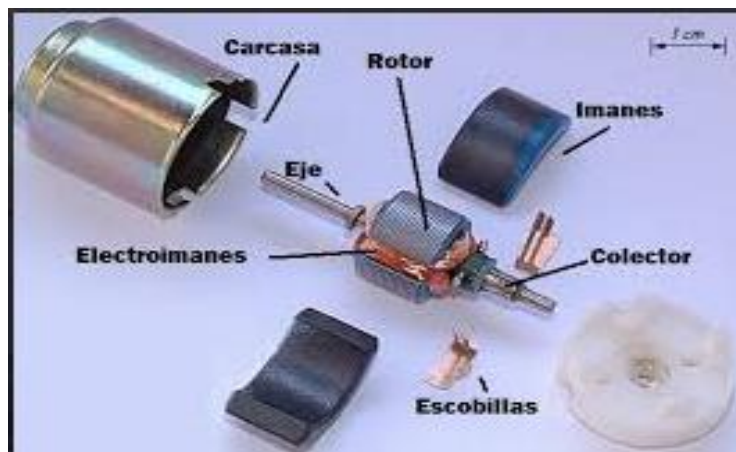


Figura 11 Partes del motor DC

Fuente: (Dea, 2015)

2.8.1 Reducciones mecánicas

Toda máquina cuyo movimiento sea generado por un motor (ya sea eléctrico, de explosión u otro) necesita que la velocidad de dicho motor se adapte a la velocidad necesaria para el buen funcionamiento de la máquina. Además de esta adaptación de velocidad, se deben contemplar otros factores como la potencia mecánica a transmitir, la potencia térmica, rendimientos mecánicos (estáticos y dinámicos). Esta adaptación se realiza generalmente con uno o varios pares de engranajes que adaptan la velocidad y potencia mecánica montados en un cuerpo compacto denominado reductor de velocidad, aunque también se le denomina caja reductora. (Dea, 2015)



Figura 12 Motores DC con caja reductora

Fuente: (Dea, 2015)

2.9 Puente H para control de motores DC

El puente H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder; los puentes H ya vienen hechos en algunos circuitos integrados, pero también se pueden construir a partir de componentes eléctricos y/o electrónicos. (Mecafenix, 2017)

Un puente H se construye con 4 interruptores (mecánicos o mediante transistores),(Figura 13). Cuando los interruptores S1 y S4 están cerrados (S2 y S3 abiertos) se aplica una tensión haciendo girar el motor en un sentido. Abriendo los interruptores S1 y S4 (cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

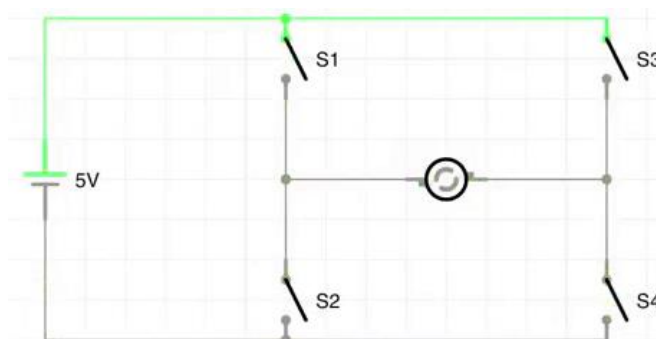


Figura 13 Puente H con interruptores

Fuente: (Mecafenix, 2017)

Un puente H no solo se usa para invertir el giro de un motor, también se puede usar para frenarlo de manera brusca, (ver tabla 10); al hacer un corto entre los bornes del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta. (Mecafenix, 2017)

Tabla 10

Estados del puente H

S1	S2	S3	S4	Resultado
1	0	0	1	Avanza
0	1	1	0	Retrocede

0	0	0	0	Se detiene bajo inercia propia
1	0	1	0	Frenado brusco

Fuente: (Mecafenix, 2017)

2.9.1 Driver Puente H L298N

El driver L298N es un puente H Dual, permite controlar el giro y la velocidad de 2 motores DC de hasta 2A o un Motor paso a paso bipolar/unipolar.(figura 14)

El módulo permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de microcontroladores y tarjetas de desarrollo como Arduino, Raspberry Pi o Launchpads de Texas Instruments. Tiene integrado un regulador de voltaje de 5V encargado de alimentar la parte lógica del L298N, el uso de este regulador se hace a través de un Jumper y se puede usar para alimentar la etapa de control. (Veloso, 2016)

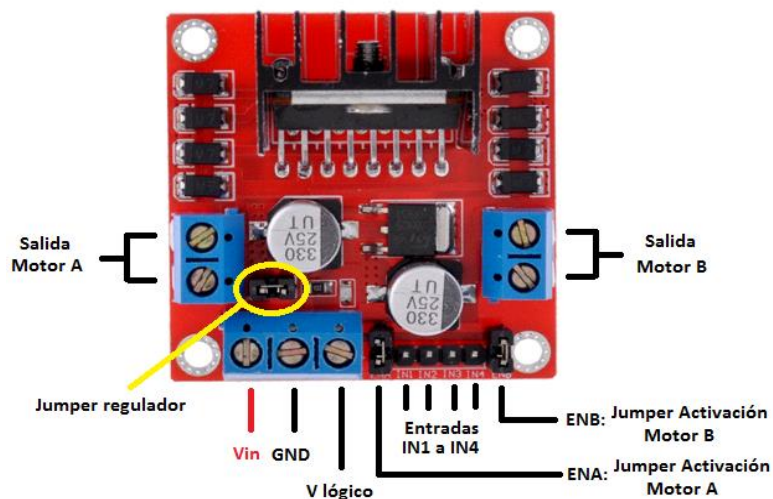


Figura 14 Driver Puente H L298N

Fuente: (Veloso, 2016)

Los pines ENA, IN1, IN2 corresponden a las entradas para controlar el MOTOR A (OUT1 y OUT2). De igual manera ENB, IN3, IN4 permiten controlar el MOTOR B (OUT3 y OUT4).

Los pines ENA y ENB, sirven para habilitar o deshabilitar sus respectivos motores, generalmente se utilizan para controlar la velocidad, ingresando una señal de PWM por estos pines. Si no se usan se deben de conectar los Jumper para que siempre estén habilitados.

CAPITULO III

4.1 Preliminares

En el presente capítulo se detalla paso a paso como se realizó el control de movimiento de un brazo robótico del laboratorio de instrumentación virtual utilizando la tecnología MYO Armband, el brazo robótico que se utilizó en el presente trabajo se muestra en la figura 15, mientras que el módulo bluetooth HM-11, placa Arduino UNO y el brazalete Myo Armband se muestra en la figura 16. La maqueta del control de movimiento de un brazo robótico que se realizó se muestra en la figura 17, también consta de una fuente de alimentación de 5 V y 9 V independientes, además de un módulo de indicación de los diferentes movimientos y un módulo divisor de voltaje. El diagrama de la figura 18, simboliza las conexiones realizadas en la maqueta para el control de movimiento del brazo robótico.

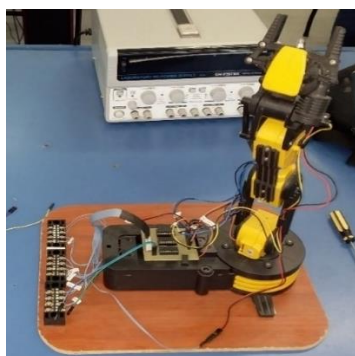


Figura 15 Brazo robótico

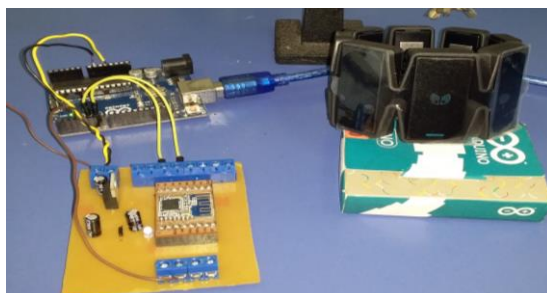


Figura 16 Módulo bluetooth HM-11, placa Arduino UNO y el brazalete Myo Armband



Figura 17 Maqueta para el control del brazo robótico

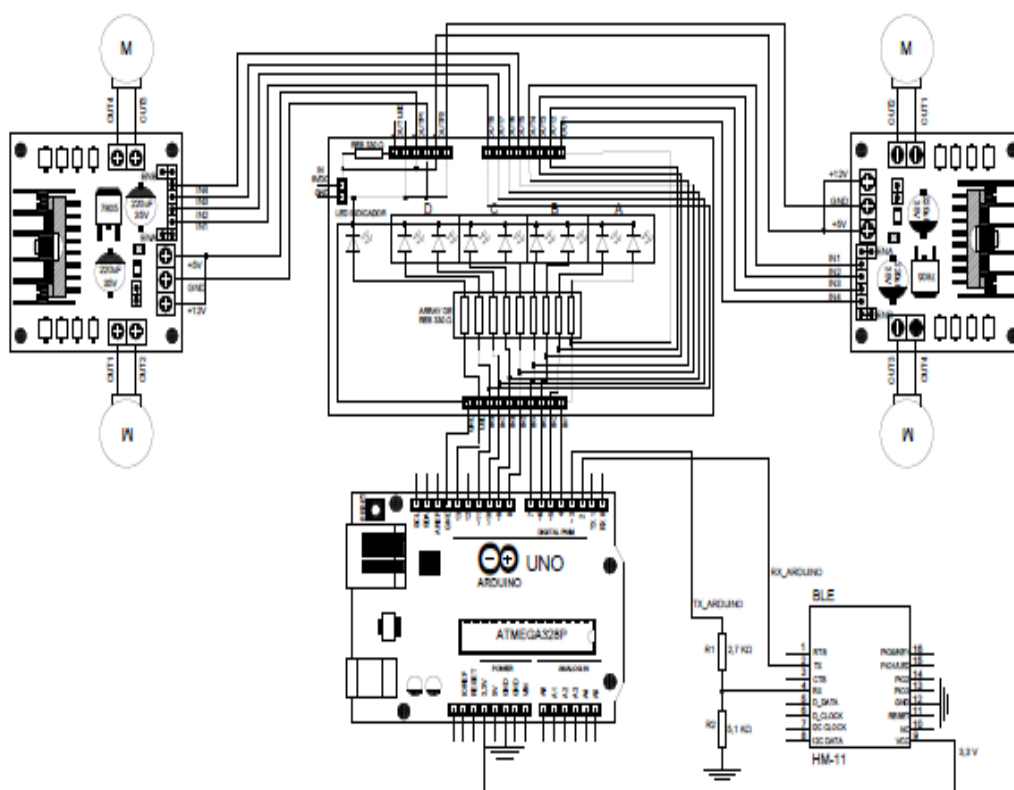


Figura 18 Diagrama principal de conexiones

En el anexo A, se muestra el esquema principal de conexiones y el LAYOUT interno de la maqueta del brazo robótico.

4.2 Componentes para el desarrollo del control de movimiento de un brazo robótico

Los componentes seleccionados para la implementación de un control de movimiento de un brazo robótico son los siguientes:

- Brazaletes MYO Armband.
- Módulo Bluetooth HM-11.
- Módulo de indicación de movimientos.
- Módulo divisor de voltaje.
- Fuente de alimentación de 5 V y 9 V independientes.
- Brazo robótico.
- Arduino UNO.
- Módulo Puente H L298N.
- Cables de conexión.

4.3 Carga del Firmware en el módulo Bluetooth HM-11

- **Descripción:** El Firmware es un pequeño programa que nos permite la comunicación entre el brazalete Myo Armband y el módulo Bluetooth HM-11.(Figura 19)

```

C:\WINDOWS\system32\cmd.exe
C:\CCloader-master\Windows>CCloader.exe 4 C:\CCloader-master\MyoBridge_CC2541.bin 0
Copyright (c) 2013 RedBearLab.com
CCloader.exe version 0.5
Comport : COM4
Bin file : C:\CCloader-master\MyoBridge_CC2541.bin
Device : Default (e.g. UNO)

Comport open!
<Baud:115200> <data:8> <parity:none> <stopbit:1> <DTR:off> <RTS:off>
File open!
[WARNING: File's size isn't the integer multiples of 512 bytes, and
the last block will be filled in up to 512 bytes with 0xFF!
Block total: 274
Enable transmission...
Request sent already!
/*****
* If there is no respond last for 3s, please press "Ctrl+C" to exit!
* And pay attention to :
* 1. The connection between computer and Arduino;
* 2. The connection between Arduino and CC2540;
* 3. Whether the device you using is Leonardo or not;
* 4. Other unexpected errors.
*****/
Waiting for respond from Arduino...
Uploading firmware...
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
3  34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
6  64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
9  94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
12 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
15 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
18 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
21 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214
24 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
27 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
30 263 264 265 266 267 268 269 270 271 272 273 274 Upload successfully!
File closed!
Comport closed!

```

Figura 19 Ventana CMD del firmware cargado

La mayoría de los modelos de Bluetooth de bajo consumo de energía poseen características muy limitadas, y ninguno de estos son capaces de conectarse directamente con el brazalete Myo Armband. Este inconveniente se soluciona cargando un Firmware personalizado en el módulo Bluetooth HM-11.

Para cargar el Firmware en el Bluetooth HM-11 se utilizó la placa Arduino UNO en conjunto con el programa CCLoader. Por medio de los pasos que se detallan a continuación:

1. Se carga el programa CCLoader en el microcontrolador Arduino UNO que nos permite subir el Firmware en el módulo Bluetooth HM-11.
2. Para conectar el microcontrolador Arduino UNO con el módulo Bluetooth HM-11, se conecta los pines C_DATA, C_CLOCK, RESET del módulo Bluetooth HM-11 a los pines digitales 6,5 y 4 del microcontrolador Arduino UNO. (Figura 20)
3. Para energizar el módulo Bluetooth HM-11 se utiliza la salida de 3.3V y GND que posee el microcontrolador Arduino UNO. (Figura 20)

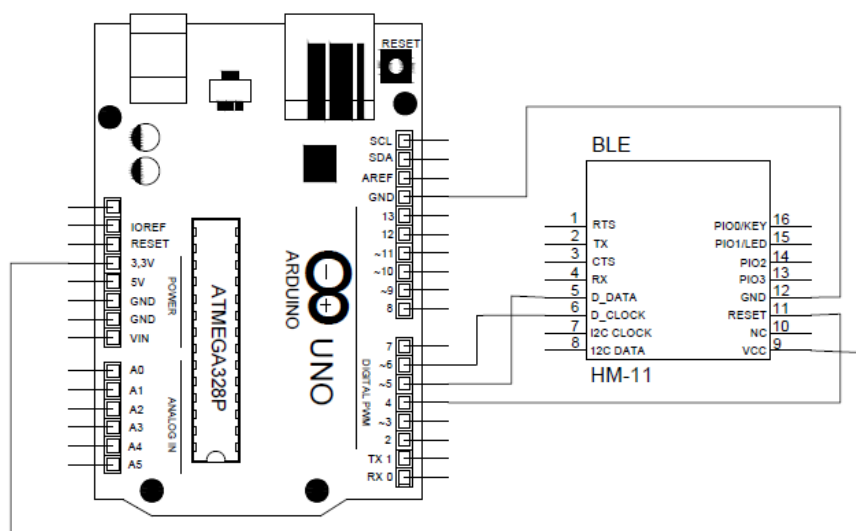


Figura 20 Diagrama de conexiones del módulo Bluetooth HM-11 con la placa Arduino UNO

- Una vez realizada las conexiones entre el módulo bluetooth HM-11 con el Arduino UNO, se debe conectar el Arduino UNO a la PC y abrir la ventana CMD. (Figura 21)

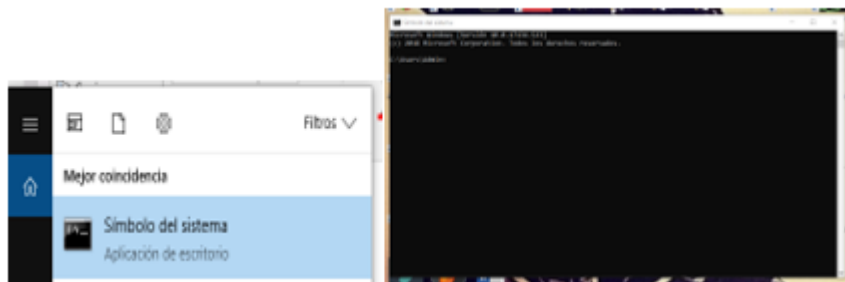


Figura 21 Ventana CMD

- Una vez abierto la ventana CMD, escribir: `C:\CCLoader-master\Windows>CCLoader.exe 4 C:\CCLoader-master\MyoBridge_CC2541.bin 0`, (Figura 22); que especifica la dirección del archivo `CcLoader.exe`, número de puerto COM donde se encuentra conectada la placa Arduino UNO y la dirección del archivo `.bin`; “1” en caso de utilizar un Arduino Nano o “0” en caso de utilizar un Arduino UNO. Esto permite conocer la dirección en donde se encuentra el `MyoBridge_CC2541` que se utilizó como un puente entre el brazaletes Myo y el Módulo Bluetooth HM-11 para controlar desde la placa Arduino.

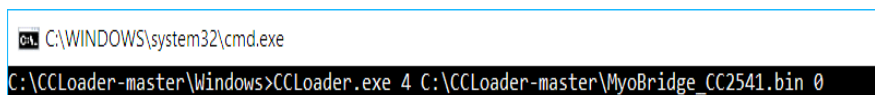
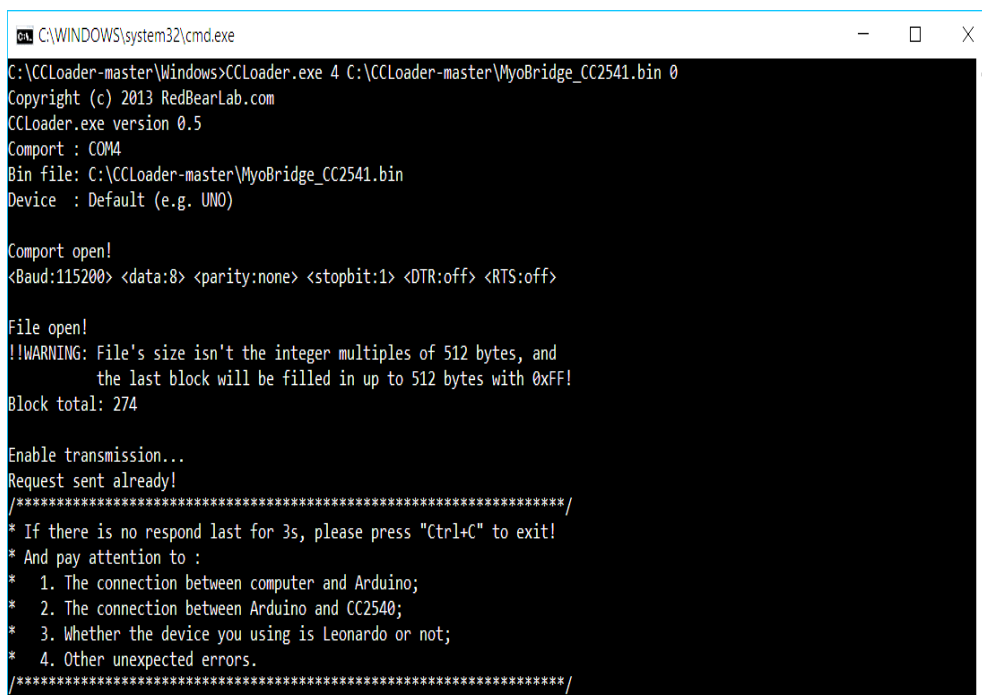


Figura 22 Comando de dirección del MyoBridge_CC2541

- Una vez que se escribe el comando de dirección del `MyoBridge_CC2541` se da un ENTER y aparece la siguiente información: (Figura 23)
 - Comport: COM4, (indica que el puerto COM4 está seleccionado).
 - Bin file: `C:\CCLoader-master\MyoBridge_CC2541.bin`, (es la dirección del archivo `MyoBridge_CC2541` que va a cargar).

- Device: Default (e.g. UNO), (nos indica que por defecto el dispositivo al que se conectara es el Arduino UNO).
- Comport open! <baud:115200> <data:8> <parity:none> <stopbit:1> <DTR:off> <RTS:off>, (indica que el puerto está abierto a una velocidad de 115200, 8 bit de datos, sin paridad, un bit de parada, no control de dirección de flujo y no control de dirección RTS).



```

C:\WINDOWS\system32\cmd.exe
C:\CCLoader-master\Windows>CCLoader.exe 4 C:\CCLoader-master\MyoBridge_CC2541.bin 0
Copyright (c) 2013 RedBearLab.com
CCLoader.exe version 0.5
Comport : COM4
Bin file: C:\CCLoader-master\MyoBridge_CC2541.bin
Device : Default (e.g. UNO)

Comport open!
<Baud:115200> <data:8> <parity:none> <stopbit:1> <DTR:off> <RTS:off>

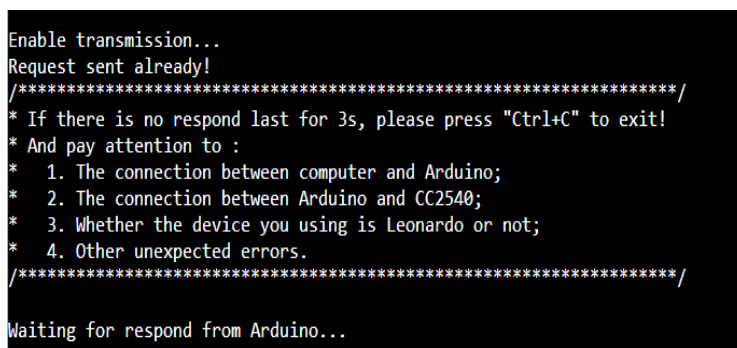
File open!
!!WARNING: File's size isn't the integer multiples of 512 bytes, and
           the last block will be filled in up to 512 bytes with 0xFF!
Block total: 274

Enable transmission...
Request sent already!
/*****/
* If there is no respond last for 3s, please press "Ctrl+C" to exit!
* And pay attention to :
* 1. The connection between computer and Arduino;
* 2. The connection between Arduino and CC2540;
* 3. Whether the device you using is Leonardo or not;
* 4. Other unexpected errors.
/*****/

```

Figura 23 Comando de dirección del MyoBridge_CC2541

7. Para habilitar la transmisión esperar 3 segundos,(Figura 24); en caso de que el programa no responda, presionar las teclas “Ctrl+C” para salir.



```

Enable transmission...
Request sent already!
/*****/
* If there is no respond last for 3s, please press "Ctrl+C" to exit!
* And pay attention to :
* 1. The connection between computer and Arduino;
* 2. The connection between Arduino and CC2540;
* 3. Whether the device you using is Leonardo or not;
* 4. Other unexpected errors.
/*****/
Waiting for respond from Arduino...

```

Figura 24 Tiempo de espera para habilitar la transmisión

8. Una vez cargado el programa, se despliega una secuencia de dígitos y al final un mensaje de carga terminada “Upload successfully”.(Figura 25)

```

Uploading firmware...
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
3 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
3 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
3 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142
143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214
215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262
263 264 265 266 267 268 269 270 271 272 273 274 Upload successfully!
File closed!
Comport closed!

```

Figura 25 Carga completa del firmware

4.4 Desarrollo de los componentes de la maqueta para el control del brazo robótico

La maqueta para el control del brazo robótico diseñada posee los siguientes componentes:

- Fuente de alimentación
- Módulo divisor de voltaje
- Módulo de indicación de movimientos
- Base para el módulo Bluetooth HM-11.

4.4.1 Fuente de alimentación de 5 V y 9 V independientes

Debido a que la alimentación de voltaje para la placa Arduino Uno, tiene un rango de 7 a 12 voltios; la alimentación del módulo de indicación de los diferentes movimientos y a los módulos puente H L298, necesitan un voltaje de alimentación de 5 voltios, se diseñó e implemento una fuente de alimentación de 5 V y 9 V independientes. (Figura 26)



Figura 26 Fuente de alimentación de 5 V y 9 V.

4.4.1.1 Análisis matemático para el dimensionamiento de los capacitores electrolíticos

Para el diseño de la fuente de alimentación, se debe tener en cuenta diversos factores, uno de ellos es la corriente que se va a suministrar. Los motores DC del brazo robótico consumen cada uno 250 mA con un voltaje de 5 V, (estos dispositivos son los que más corriente consumen). Por tal motivo la fuente de alimentación diseñada fue de 1 A, con una salida de 5 V utilizando un regulador de voltaje 7805 y otra salida de 9 V utilizando un regulador de voltaje 7809.

La regla empírica que se aplica para colocar el condensador suele estar sobre los 2000 μF por amperio de salida y el voltaje del doble del valor superior estándar al requerido. La Figura 27 muestra un diagrama de las etapas de la fuente de voltaje.

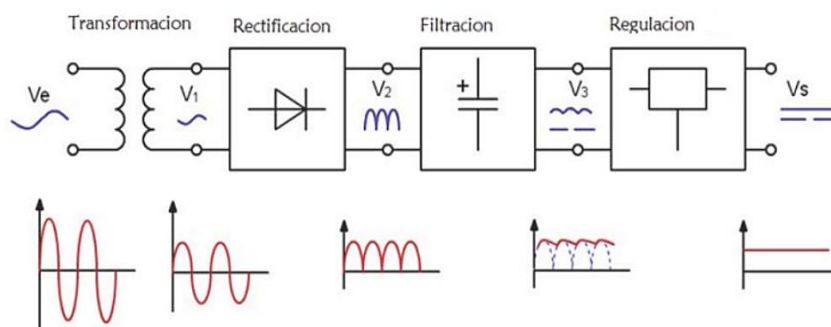


Figura 27 Diagrama de las etapas de la fuente de voltaje

Los siguientes datos del transformador, se utilizan para realizar los cálculos para el dimensionamiento de los condensadores:

$V_{in} = 120 \text{ Vac}$

$V_{out} = 12 \text{ Vac}$

$f = 60 \text{ Hz}$

Relación de transformación = 10:1

Donde:

V_{in} = Voltaje de entrada al bobinado primario del transformador.

V_{out} = Voltaje de salida del bobinado secundario de transformador.

f = Frecuencia.

Mediante las siguientes fórmulas se puede dimensionar los condensadores electrolíticos.

Ecuación 1: Tensión antes del puente de diodos

$$V_{maxs} = \frac{V_{rmsp} \times \sqrt{2}}{n_p} \times n_s$$

$$V_{maxs} = \frac{120 \text{ V}(\sqrt{2})}{10} \times 1$$

$$V_{maxs} = 16,97 \text{ V}$$

Donde:

V_{maxs} = voltaje máximo del bobinado secundario.

V_{rmsp} = voltaje raíz media cuadrática del bobinado primario.

n_p = número de espiras del devanado primario.

n_s = número de espiras del devanado secundario.

Ecuación 2: Etapa de rectificación

$$V_{dc} = \frac{2V_{maxs}}{\pi}$$

$$V_{dc} = \frac{2(16,97)}{\pi}$$

$$V_{dc} = 10,80 \text{ V}$$

Donde:

Vdc= Voltaje continuo en la etapa de rectificación.

Vmaxs= Voltaje máximo del bobinado secundario.

Ecuación 3: Etapa de filtrado

$$V_{dc} = \frac{V_{maxs}}{1 + \sqrt{3} \times \delta}$$

$$V_{dc} = \frac{12 \times \sqrt{2}}{1 + \sqrt{3} \times 0,03}$$

$$V_{dc} = 16,13 V$$

Donde:

Vdc= Voltaje continuo de la etapa de filtrado.

Vmaxs= Voltaje máximo del bobinado secundario.

δ = Factor de rizado.

Nota: el factor de rizado es la relación que existe entre el valor eficaz del voltaje de rizado y el voltaje de salida, se considera óptima una señal con un factor de rizado menor del 10%; en este caso se consideró un factor de rizado del 3%.

Ecuación 4: Dimensionamiento del capacitor para una corriente de 500 mA

$$C = \frac{I_{dc}}{4\sqrt{3} \times f \times V_{dc} \times \delta}$$

$$C = \frac{500 \times 10^{-3} A}{4\sqrt{3} \times 60Hz \times 16,13 V \times 0,03}$$

$$C = 2,485664 \times 10^{-3} F$$

$$C = 2485,66 \times 10^{-6} F$$

Donde:

C= Capacitor.

I_{dc}= Corriente de carga.

Vdc= Voltaje continuo de la etapa de filtrado.

f = Frecuencia.

δ = Factor de rizado.

Ecuación 5: Dimensionamiento del capacitor para una corriente de 1 A

$$C = \frac{I_{dc}}{4\sqrt{3} \times f \times V_{dc} \times \delta}$$

$$C = \frac{1 A}{4\sqrt{3} \times 60Hz \times 16,13 V \times 0,03}$$

$$C = 4,97132 \times 10^{-3} F$$

$$C = 4971,32 \times 10^{-6} F$$

Donde:

C= Capacitor.

I_{dc} = Corriente de carga.

V_{dc} = Voltaje continuo de la etapa de filtrado.

f = Frecuencia.

δ = Factor de rizado.

4.4.1.2 Materiales utilizados para la fuente de alimentación de 5 V y 9 V

- 2 puentes de diodos de 1 A.
- 2 borneras de dos salidas.
- 1 bornera de 3 salidas.
- 2 capacitores cerámicos de 0.1 uF (104).
- 1 capacitor electrolítico de 2200 uF a 25 V.
- 1 capacitor electrolítico de 1200 uF a 25 V.
- 2 resistencias de 330 ohm.
- 2 diodos led.
- 1 regulador 7805.
- 1 regulador 7809.
- 2 disipadores de calor.
- 1 transformador de 110 V a 12 V de TAP central.

- 1 placa de baquelita

4.4.1.3 Diseño de la placa de la fuente de alimentación

Para la construcción de la fuente de alimentación se realizó las diferentes pistas en el software Proteus, así como la simulación del funcionamiento,(Figura 28); el software Proteus nos permite generar un código (generate Gerber/Excellon Files), el cual sirve para generar un código G en el software COPPER CAM para realizar la placa en la máquina CNC del laboratorio de instrumentación virtual.

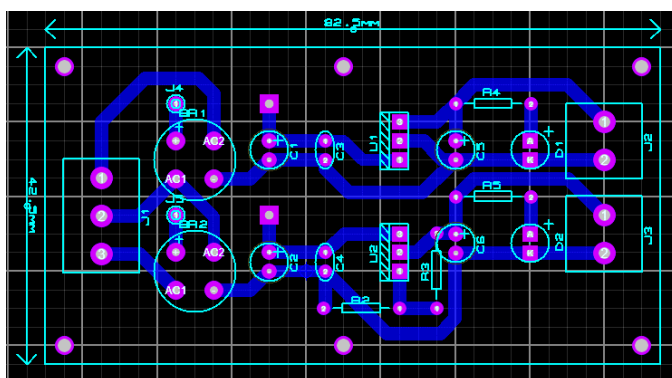


Figura 28 Diseño de la fuente de alimentación de 5 V y 9 V en el software Proteus

4.4.1.4 Grabado de las pistas en la placa de baquelita

Una vez generado el código G en el software COPPER CAM las pistas se graban en la máquina CNC utilizando el programa MACH3MILL, para el correcto uso de la máquina CNC del laboratorio de instrumentación virtual se debe tomar en cuenta las recomendaciones del manual y obtener un buen resultado en la placa evitando posibles averías. En la figura 29 se puede observar el grabado de la pista utilizando la máquina CNC.

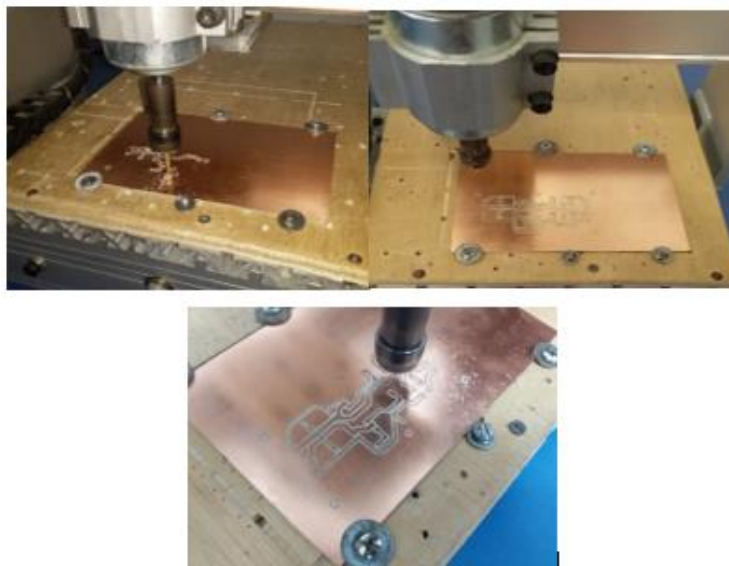


Figura 29 Grabado de las pistas de la fuente de alimentación de 5 V y 9 V

4.4.1.5 Montaje de los componentes en la placa de la fuente de alimentación

Una vez finalizada la placa en la máquina CNC, se sueldan los componentes en su respectivo lugar (Figura 30), para posteriormente realizar las respectivas pruebas de funcionamiento.



Figura 30 Montaje de los componentes en la placa de la fuente de alimentación

4.4.2 Módulo divisor de voltaje

Para la transmisión y recepción de datos entre el Bluetooth HM-11 con el Arduino UNO, se utilizó un divisor de voltaje o tensión el cual permitió reducir el voltaje del PIN de transmisión TX del Arduino UNO, esto evita posibles fallos o incluso el mal funcionamiento del Bluetooth HM-11, ya que el bluetooth HM-11 trabaja con un voltaje máximo de 3.3 V, y los pines de salida del Arduino entregan 5 V. En la figura 31 se muestra la conexión del Bluetooth HM-11 con la placa Arduino UNO.

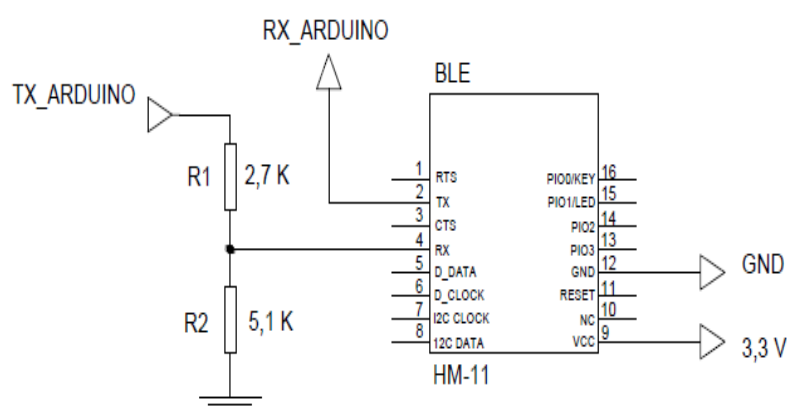


Figura 31 Esquema de conexión del Bluetooth HM-11

Con el cálculo indicado en la ecuación 6, se obtiene un voltaje de 3,26 V que es óptimo para el correcto funcionamiento del Bluetooth HM-11. En la figura 32 se muestra un esquema del divisor de voltaje.

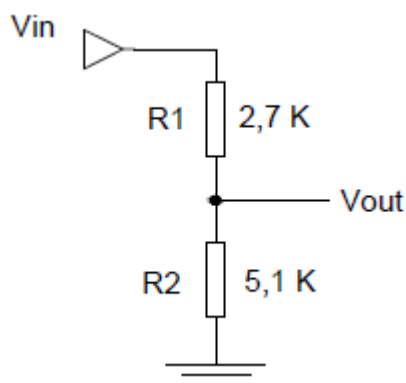


Figura 32 Esquema del divisor de voltaje

Ecuación 6: Fórmula para el divisor de voltaje

$$V_{out} = \frac{R2}{R1 + R2} \times V_{in}$$

$$V_{out} = \frac{2,7 K}{2,7 K + 5,1 K} \times 5 V$$

$$V_{out} = 3,26 V$$

4.4.2.1 Materiales para el desarrollo del módulo divisor de voltaje

- 3 resistencias de 2,7 K ohm.
- 3 resistencias de 5,1 K ohm.
- 6 borneras de dos salidas.
- 1 placa de baquelita.

4.4.2.2 Diseño de la placa para el módulo del divisor de voltaje

Para la construcción del divisor de voltaje se diseñó las diferentes pistas en el software Proteus (Figura 33), que se realizó en la máquina CNC del laboratorio de instrumentación virtual.

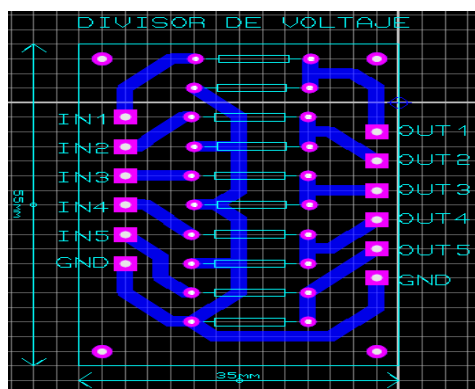


Figura 33 Diseño del divisor de voltaje en el software Proteus

4.4.2.3 Grabado de las pistas en la placa de baquelita

Después del diseño, Proteus genera un código, que sirve para generar un código G en el programa COPPER CAM el cual permite grabar las pistas en la placa de baquelita haciendo uso de la máquina CNC. En la figura 34 se puede observar el grabado de la pista utilizando la máquina CNC.



Figura 34 Grabado de las pistas del divisor de voltaje

4.4.2.4 Montaje de los componentes en la placa del módulo divisor de voltaje de voltaje

Una vez terminada la placa en la máquina CNC, se deben soldar los componentes (Figura 35), para posteriormente realizar las respectivas pruebas de funcionamiento.



Figura 35 Montaje de los componentes en la placa del módulo divisor de voltaje

4.4.3 Módulo de indicación de movimientos

El módulo de indicación de movimientos permite saber qué movimiento está ejecutando el brazo robótico cada vez que se realiza un movimiento preestablecido en el programa principal. En la tabla 11 se muestra los movimientos y la acción que realiza el brazo robótico.

Tabla 11

Movimientos y acción del brazo robótico

I T E M	PARTE DEL BRAZO ROBÓTICO	COLOR	MOVIMIENTOS	
			LED1	LED2
A	PINZA	ROJO	DERECHA	IZQUIERDA
B	ANTEBRAZO	AMARILLO	ARRIBA	ABAJO
C	BASE	VERDE	GIRO HORARIO	GIRO ANTIHORARIO
D	MUÑECA	NARANJA	CIRCULO ANTIHORARIO	CIRCULO HORARIO

4.4.3.1 Materiales para el desarrollo del módulo de indicación de movimientos

- 1 bornera de dos salidas.
- 9 resistencias de 330 Ohm.
- 2 leds de color rojo.
- 2 leds de color amarillo.
- 2 leds de color verde.
- 2 leds de color naranja.
- 1 led de color azul.
- 2 regletas de espadines macho.
- 1 placa de baquelita.
- 1 ácido férrico.

- 1 plancha.
- 1 esponjilla de acero.
- 1 taladro pequeño
- Brocas de 0,7 y 0,5 mm

4.4.3.2 Diseño de la placa del módulo de indicación de movimientos

Para la construcción del módulo de indicación de movimientos se realizó las diferentes pistas en el software Proteus (Figura 36), y posteriormente se exportó el diseño ya terminado en formato PDF; aplicando el método de planchado se grabó las pistas en la placa de baquelita.

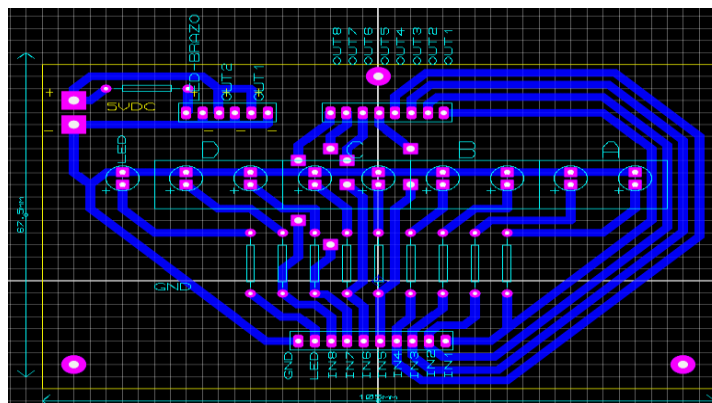


Figura 36 Diseño del módulo de indicación de movimientos en el software Proteus

4.4.3.3 Grabado de las pistas en la placa de baquelita utilizando el método del planchado

Después de exportar a PDF el circuito del módulo de indicación de movimientos se imprimió en color blanco y negro en una hoja de papel fotográfico utilizando una impresora a laser, luego se recortó el circuito dejando un borde de 1 cm más grande de la medida establecida y se colocó en la placa de baquelita para aplicar calor utilizando una plancha por un tiempo de 10 a 15 minutos.(Figura 37)



Figura 37 Aplicación de calor para la transferencia de las pistas a la placa de baquelita

Una vez que transcurridos los minutos recomendados, se colocó agua fría para poder retirar el papel con mayor facilidad y sin que exista el riesgo de dañar las pistas. Una vez retirado el papel de la placa de baquelita se colocó el ácido férrico en un recipiente de plástico (Figura 38), luego se sumergió la placa de baquelita con las pistas en el recipiente que contiene el ácido férrico en constante movimiento durante un tiempo de 5 minutos aproximadamente hasta que el cobre se va desprendiendo dejando solo las pistas que vamos a utilizar, una vez que el cobre se desprendió en su totalidad, se saca la placa de baquelita del ácido férrico, se debe lavar la placa y limpiar la tinta que se haya quedado en la placa utilizando una esponjilla de acero.



Figura 38 Recipiente plástico con ácido férrico

Con la placa completamente limpia, realizar los orificios usando un taladro pequeño y brocas de 0,7 y 0,5 mm según sea el caso, posteriormente se instalan los componentes en su lugar, y se deben soldar obteniendo el siguiente resultado.(Figura 39)

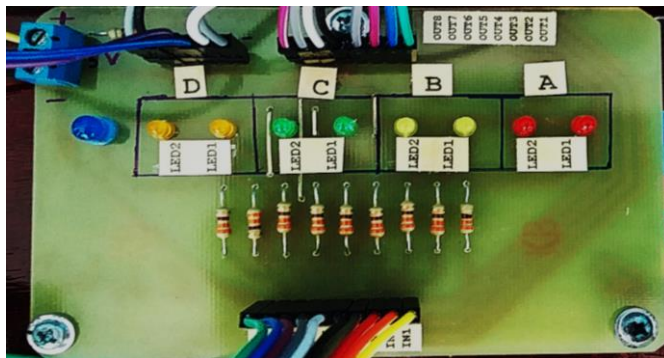


Figura 39 Módulo indicador de movimientos

4.4.4 Base para el módulo Bluetooth HM-11

La base para el módulo Bluetooth HM-11 nos permite realizar las conexiones de una mejor manera, esta base también permite reemplazar el Bluetooth HM-11 de forma rápida en caso de tener una falla.

4.4.4.1 Materiales para el desarrollo de la base para el módulo Bluetooth HM-11

- Módulo Bluetooth HM-11.
- 1 placa de baquelita de 10x10 cm.
- 7 borneras de dos salidas.
- 1 capacitor electrolito de 0,1 uF a 25V.
- 1 diodo led.

4.4.4.2 Diseño de la placa para la base del módulo Bluetooth HM-11

Para la construcción de la base para el módulo Bluetooth HM-11, se diseñó las diferentes pistas en el software Proteus,(Figura 40); que se realizará en la máquina CNC del laboratorio de instrumentación virtual.

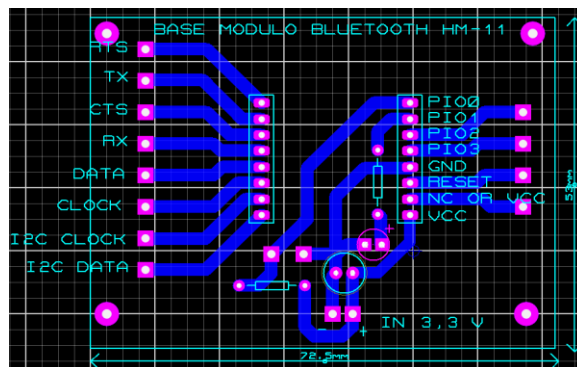


Figura 40 Diseño de la base para el módulo Bluetooth HM-11 en el software Proteus

4.4.4.3 Grabado de las pistas en la placa de baquelita

Después del diseño, Proteus genera un código usado para crear un código G en el programa COPPER CAM el cual permite grabar las pistas en la placa de baquelita haciendo uso de la máquina CNC. En la figura 41 se puede observar el grabado de la pista utilizando la máquina CNC.



Figura 41 Grabado de las pistas de la base para el módulo Bluetooth HM-11

4.4.4.4 Montaje de los componentes en la placa de la base para el módulo Bluetooth HM-11

Una vez terminada la placa en la máquina CNC, se sueldan los componentes, para posteriormente realizar las respectivas pruebas de funcionamiento. En la figura 42 se muestra el módulo Bluetooth HM-11

montado correctamente en la base que se elaboró en la máquina CNC del laboratorio de instrumentación virtual.

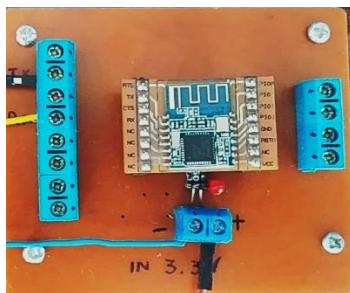


Figura 42 Módulo Bluetooth HM-11 montado en la base

4.5 Etapas para el control de movimiento del brazo robótico

En el diagrama de bloques de la figura 43, se muestra las etapas para el control de movimiento del brazo robótico.



Figura 43 Diagrama de bloques de las etapas para el control de movimiento del brazo robótico

4.5.1 Sincronización

En la etapa de sincronización el gesto de bloqueo / desbloqueo y la posición del brazalete son diferentes para cada usuario en términos de intensidad de señal. Es por ese motivo que se debe sincronizar antes de usar la detección de gestos, esta etapa comienza con una vibración prolongada, luego de esa larga vibración, se debe realizar un gesto fuerte con la mano (gesto de puño), como se muestra en la figura 44. Una breve vibración indicará el final de este procedimiento. Ahora puede bloquear o desbloquear la detección de gestos con el gesto de puño.



Figura 44 Gesto de puño

4.5.2 Bloqueo / Desbloqueo

Para indicar cuándo se desea grabar un gesto, primero se desbloquea la detección de gestos, para ello; se realiza el gesto de bloqueo / desbloqueo utilizado en la etapa de sincronización. El led de lectura indica que el desbloqueo se realizó correctamente, como se muestra en la Figura 45; ahora se puede generar el gesto que se desea grabar, finalmente se debe volver a realizar el gesto de bloqueo / desbloqueo para finalizar su comando. La librería ahora evaluará el gesto y dará una notificación, si se detectó un gesto.

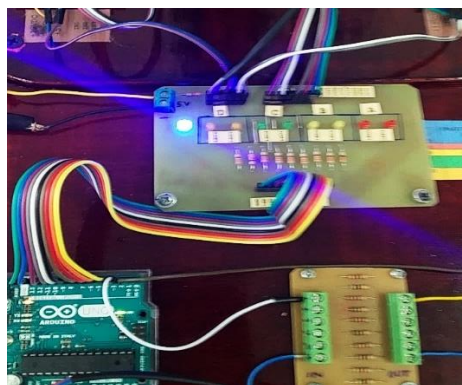


Figura 45 Luz de indicación del gesto de bloqueo / desbloqueo

4.5.3 Reconocimiento de gestos

El reconocimiento de gestos se lleva a cabo durante el estado de lectura, luego que el usuario haya realizado un gesto como movimientos de la muñeca

o una contracción muscular se generará una señal de actividad muscular del antebrazo que es detectado por los sensores mioeléctricos del brazalete Myo Armband, el estado de lectura para realizar el gesto esta temporizado una vez finalizado y si no se realizó ninguna acción se bloqueará de forma automática. Siempre para iniciar el estado de lectura se debe empezar y finalizar con el gesto de bloqueo / desbloqueo; para este lapso puede terminar o bloquear mediante la misma señal de actividad muscular que se realizó para desbloquear y realizar el gesto. Para visualizar el estado de bloqueo / desbloqueo, se hace uso de un led indicador de estado (Figura 46), el cual permanecerá encendido durante el lapso de adquisición de datos para el reconocimiento de gestos.

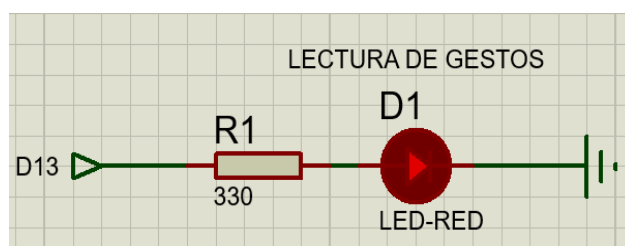


Figura 46 Indicador de estado de lectura

Al inicializar el estado de lectura, después de ejecutar la acción de bloqueo / desbloqueo, se debe realizar la acción que se desea guardar, esta acción es evaluada por el controlador y si el gesto realizado está dentro del tiempo del estado de lectura será reconocido, caso contrario se eliminará y bloqueará de forma automática.

En la figura 47 se puede visualizar utilizando el monitor serial del Software Arduino, el porcentaje de batería del brazalete Myo y el gesto “izquierda”.

```

COM7 (Arduino/Genuino Uno)
Searching for Myo...
4
5
6
connected!
% de Bateria: 91
1
0
Izquierda
1
0

```

Figura 47 Visualización del gesto “izquierda” y el porcentaje de batería en el monitor serial del Software Arduino

El monitor serial del software Arduino es una herramienta muy útil al momento de realizar simulaciones, en este caso para conocer la sincronización y el gesto realizado, pero únicamente se puede usar para simulaciones en un ordenador por esta razón se efectuó un módulo indicador de movimientos (Figura 48), que permite conocer la sincronización, el bloqueo / desbloqueo y los gestos que se realizan.

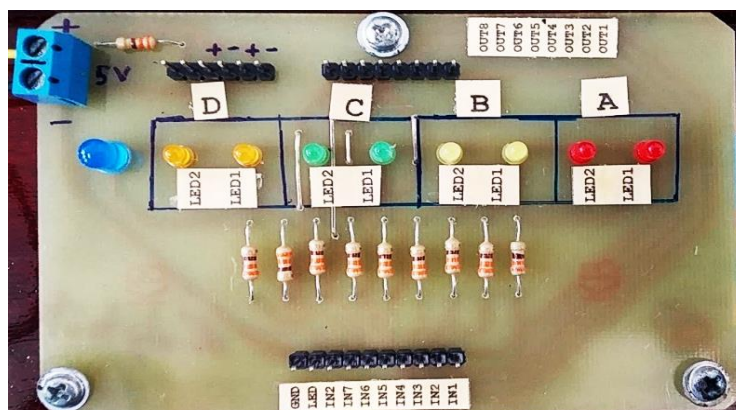


Figura 48 Módulo de indicación de movimientos

El módulo indicador de movimientos muestra los colores rojo, amarillo, verde y naranja que están divididos en 4 bloques (Figura 49), el bloque A que nos indica el control de la pinza, el bloque B que nos indica el control del antebrazo, el bloque C que nos indica el control de la base y el bloque D que

nos indica el control de la muñeca, además de un led de indicación de bloqueo / desbloqueo.

Indicación Bloqueo/desbloqueo	Bloque D Control muñeca	Bloque C Control base	Bloque B Control antebrazo	Bloque A Control pinza
----------------------------------	-------------------------------	-----------------------------	----------------------------------	------------------------------

Figura 49 Bloques del módulo indicador de movimiento

El led rojo1 del bloque A se enciende cuando se detecta que el brazo ha realizado un movimiento hacia la derecha.(Figura 50)

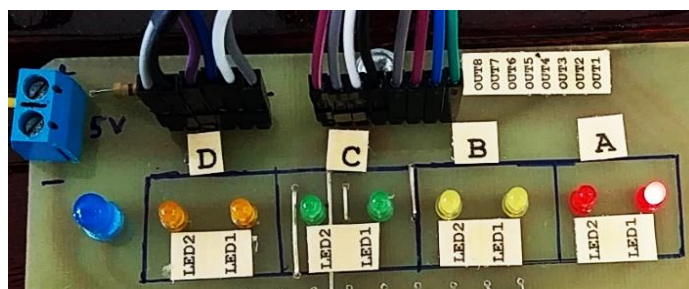


Figura 50 Indicador de movimiento en rojo1

El led rojo2 del bloque A se enciende cuando se detecta que el brazo ha realizado un movimiento hacia la izquierda.(Figura 51)

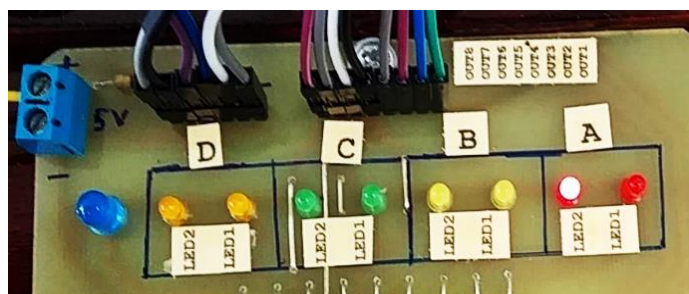


Figura 51 Indicador de movimiento en rojo2

El led amarillo1 del bloque B se enciende cuando se detecta que el brazo ha realizado un movimiento hacia arriba.(Figura 52)

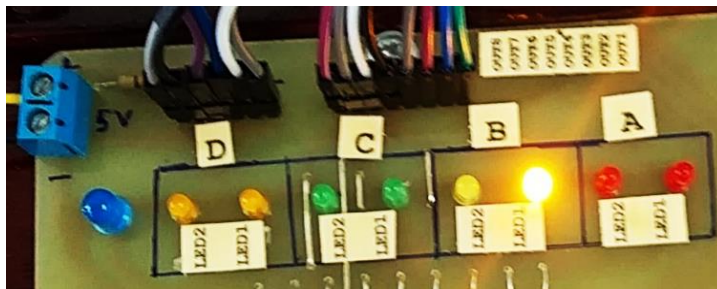


Figura 52 Indicador de movimiento en amarillo1

El led amarillo2 del bloque B se enciende cuando se detecta que el brazo ha realizado un movimiento hacia abajo.(Figura 53)

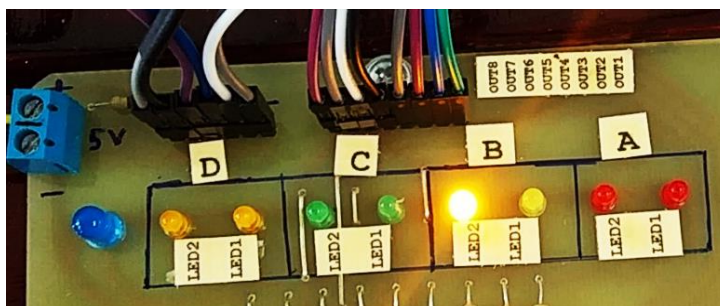


Figura 53 Indicador de movimiento en amarillo2

El led verde1 del bloque C se enciende cuando se detecta que el brazo ha realizado un movimiento de rotación de muñeca en sentido horario.(Figura 54)

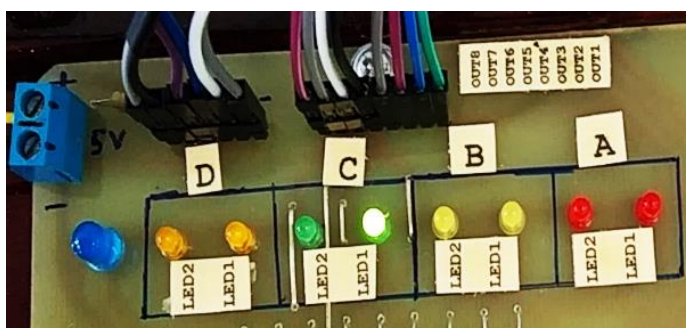


Figura 54 Indicador de movimiento en verde1

El led verde2 del bloque C se enciende cuando se detecta que el brazo ha realizado un movimiento de rotación de muñeca en sentido antihorario.(Figura 55)

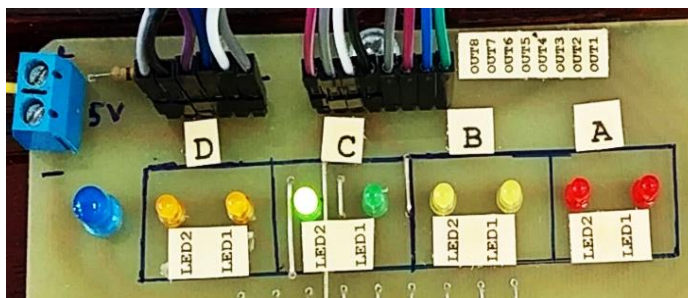


Figura 55 Indicador de movimiento en verde2

El led naranja1 del bloque D se enciende cuando se detecta que el brazo ha realizado un movimiento de círculo de muñeca en sentido antihorario.(Figura 56)



Figura 56 Indicador de movimiento en naranja1

El led naranja2 del bloque D se enciende cuando se detecta que el brazo ha realizado un movimiento de círculo de muñeca en sentido horario.(Figura 57)

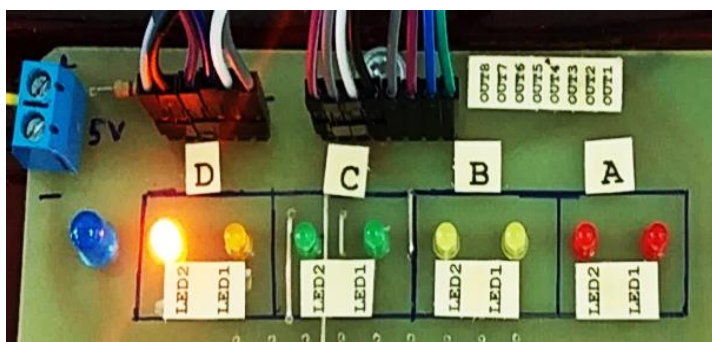


Figura 57 Indicador de movimiento en naranja2

4.6 Estructura general del software

La estructura general del software que se utilizó para realizar el control del brazo robótico está compuesta por dos librerías que se detallan a continuación:

- **Librería MyoBridge:** Esta librería nos permite básicamente la comunicación bidireccional entre el brazalete Myo Armband con la placa Arduino UNO.
- **Librería MyoIMUGestureController:** esta librería se encarga de la lectura y reconocimiento de gestos.

Además, se realizó un programa principal en el cual se establecen los parámetros y condiciones para la realización de movimientos en el brazo robótico. En la figura 58 se muestra un diagrama de bloques de la estructura general del software.

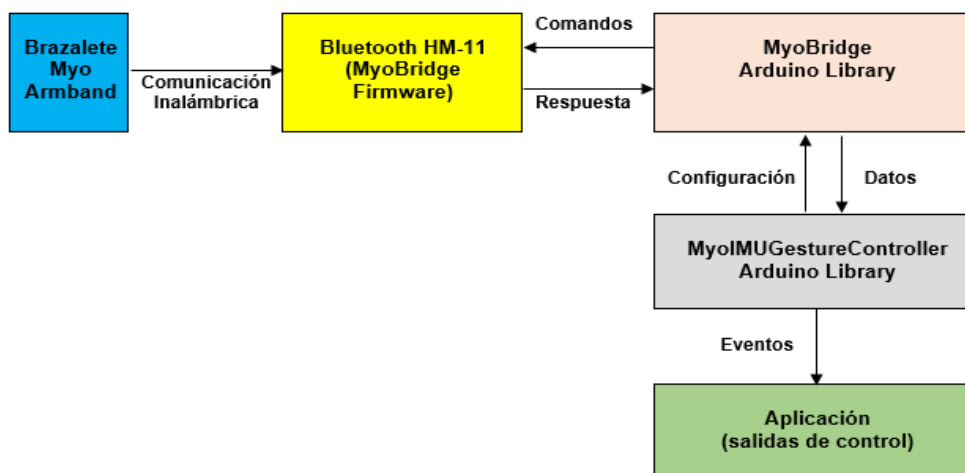


Figura 58 Diagrama de bloques de la estructura general del software

4.7 Diagrama de flujo

El diagrama de flujo del programa principal detalla la secuencia lógica de los pasos que realiza la tarjeta controladora Arduino UNO, para el control del brazo robótico.(Figura 59)

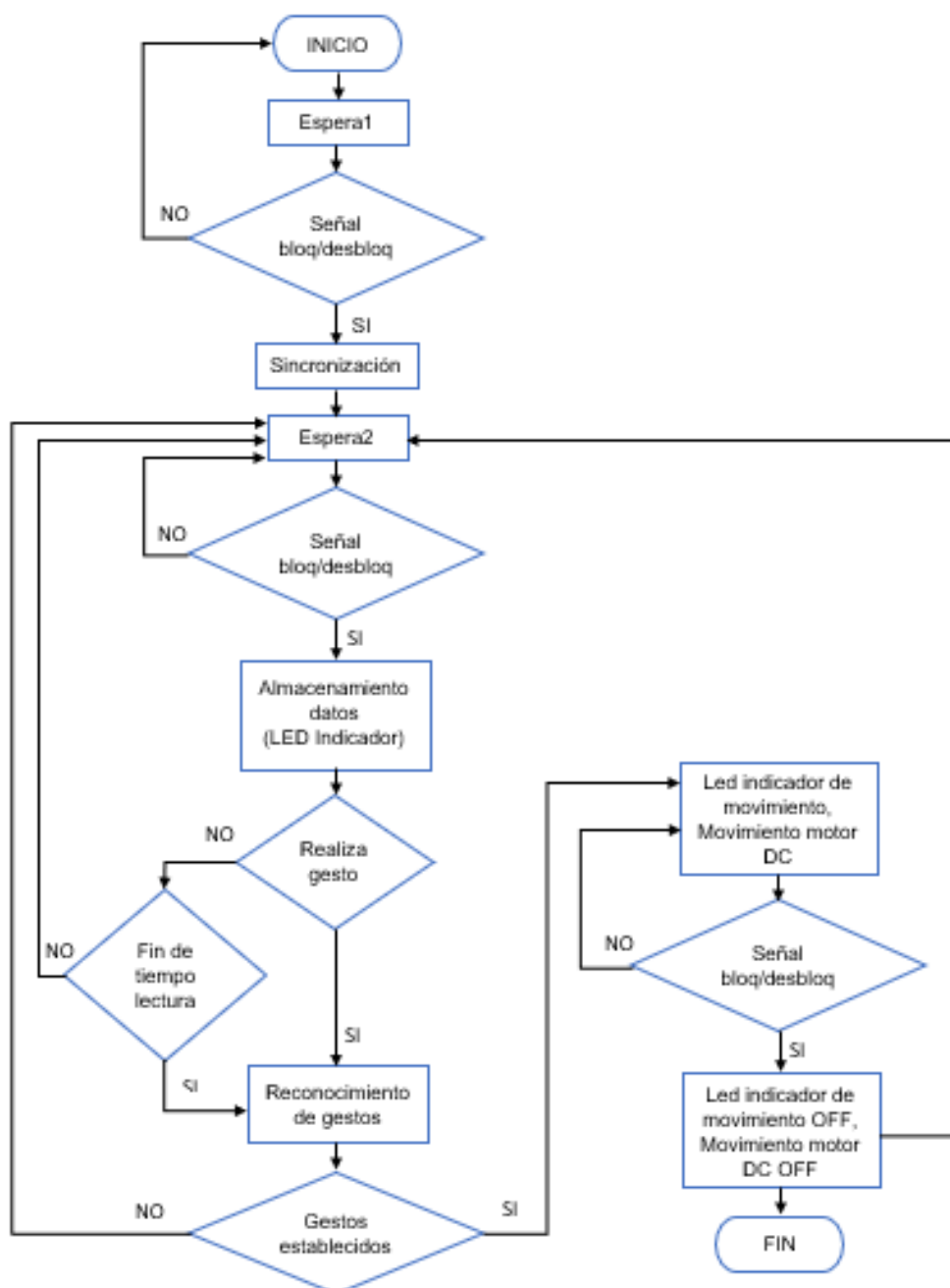


Figura 59 Diagrama de flujo del programa principal

4.8 Programación de la tarjeta Arduino UNO

Para realizar el programa principal del control de movimiento del brazo robótico se ejecutaron los siguientes pasos:

- Declarar librerías.
- Declarar puertos.
- Declarar variables.
- Void printConnectionStatus.
- Void updateLockOutput.
- Void updateControls.
- Void setup.

4.8.1 Declarar librerías

Para la programación se utilizó librerías para la comunicación con el brazalete Myo Armband.

```
#include <MyoBridge.h>
#include <MyoIMUGestureController.h>
#include <SoftwareSerial.h>
```

4.8.2 Declarar puertos

En esta parte se declararon los puertos para establecer la comunicación serial con el bluetooth HM-11 y la creación del puente de comunicación entre el brazalete Myo Armband y el Arduino UNO.

```
//SoftwareSerial connection to MyoBridge
SoftwareSerial bridgeSerial(2,3); //((2,3) (Rx arduino,Tx arduino)
//initialize MyoBridge object with software serial connection
//Iniciar el objeto MyoBridge con el software serial
MyoBridge bridge(bridgeSerial);
```

4.8.3 Declarar variables

En este paso se declara las variables para los motores del brazo robótico y el led de lectura de movimiento.

```
byte Led_Lectura = 13; // Salida pin de estado de lectura
bool bloqueo;        // bloqueo/desbloqueo
String GESTO;        // para valores de 1 o 0
byte M_PINZA1=4;     //motor de pinza
byte M_PINZA2=5;
byte M_BRAZO1=6;     //motor de brazo1
byte M_BRAZO2=7;
byte M_BASE1=8;      //motor de base
byte M_BASE2=9;
byte M_MUNIECA1=10; //motor de muñeca
byte M_MUNIECA2=11;
```

4.8.4 Void printConnectionStatus

La función void printConnectionStatus, permite imprimir las constantes de estado como string.

```
void printConnectionStatus(MyoConnectionStatus status) {
//print the status constant as string
Serial.println(status);
}
```

4.8.5 Void updateLockOutput

La función void updateLockOutput, es llamada cuando un bloqueo / desbloqueo se ha producido; al momento que la variable bloqueo (locked) es igual a "1L" esta desbloqueado, es decir el led de lectura se enciende y por consiguiente podemos realizar un movimiento con el antebrazo y si el

movimiento realizado es detectado se activa un motor que genera el movimiento de una determinada parte del brazo robótico. En caso de que la variable bloqueo sea igual a "0L" los motores permanecen apagados.

```
void updateLockOutput(bool locked) {
digitalWrite(Led_Lectura, !locked);
bloqueo=locked;    //Cuando la variable bloqueo=1 esta desbloqueado; es
decir, led del pin 13 encendido
Serial.println(bloqueo);
if(bloqueo==0)
{
digitalWrite(M_PINZA1,LOW);
digitalWrite(M_PINZA2,LOW);
digitalWrite(M_BRAZO1,LOW);
digitalWrite(M_BRAZO2,LOW);
digitalWrite(M_BASE1,LOW);
digitalWrite(M_BASE2,LOW);
digitalWrite(M_MUNIECA1,LOW);
digitalWrite(M_MUNIECA2,LOW);
}
}
```

4.8.6 Void updateControls

La función void updateControls, actualiza y detecta el gesto realizado por el movimiento del antebrazo, podemos saber si detectó el gesto si se siente una pequeña vibración; al momento de detectar el gesto se activará el motor y el brazaete se vuelve a bloquear, para detener el motor debemos realizar nuevamente el gesto de bloqueo / bloqueo.

```
void updateControls(GestureType gesture) {
GESTO = gestureToString(gesture);
//Serial.println(gestureToString(gesture));
```

```

//Serial.println(bloqueo);
bridge.vibrate(1);

if (gesture == ARM_UP && bloqueo==0) //Gesto ARM_UP detectado y
brazaletes bloqueados
{
Serial.println("Arriba");
digitalWrite(M_BRAZO1,HIGH);
digitalWrite(M_BRAZO2,LOW);
}

if (gesture == ARM_DOWN && bloqueo==0)
{
Serial.println("Abajo");
digitalWrite(M_BRAZO1,LOW);
digitalWrite(M_BRAZO2,HIGH);
}

```

4.8.7 Void setup

En la función void setup, se inicializa para la comunicación serial en donde se muestra el nivel de batería y los gestos que se realizaron, se configuran los pines como salida y también se configuran los controladores para los gestos.

```

void setup() {
  // put your setup code here, to run once:
  //initialize both serial connections
  Serial.begin(9600);
  bridgeSerial.begin(115200);
  //wait until MyoBridge has found Myo and is connected. Make sure Myo is not
  connected to anything else and not in standby!
  Serial.println(F("Searching for Myo..."));
  //initiate the connection with the status callback function

```

```
bridge.begin(printConnectionStatus);
Serial.println(F("connected!"));
//Obtiene el nivel y lo muestra
byte NivelBateria = bridge.getBatteryLevel();
Serial.print("% de Bateria: ");
Serial.println(NivelBateria);
pinMode(Led_Lectura,OUTPUT);
pinMode(M_PINZA1,OUTPUT);
pinMode(M_PINZA2,OUTPUT);
pinMode(M_BRAZO1,OUTPUT);
pinMode(M_BRAZO2,OUTPUT);
pinMode(M_BASE1,OUTPUT);
pinMode(M_BASE2,OUTPUT);
pinMode(M_MUNIECA1,OUTPUT);
pinMode(M_MUNIECA2,OUTPUT);
MyoIMUGestureController::begin(bridge,updateControls, updateLockOutput);
}
```

En el anexo B, se encuentra la programación extendida de la tarjeta Arduino UNO, encargada de controlar todo el sistema.

CAPITULO IV

5.1 Conclusiones y recomendaciones

5.1.1 Conclusiones

- La implementación del control de movimiento de un brazo robótico, que utiliza un controlador programable en software libre y la tecnología Myo Armband; nos permite tener acceso a una gran cantidad de aplicaciones que serán de utilidad para la formación de los estudiantes de la Unidad de Gestión de Tecnologías.
- El control del brazo robótico se realizó a través de la comunicación Bluetooth BLE V4.0 entre el brazalete Myo Armband y la tarjeta Arduino UNO, con lo cual se eliminó el uso de cables para la recopilación de las señales provenientes de los 8 sensores mioeléctricos del brazalete Myo Armband.
- El movimiento del antebrazo genera una actividad eléctrica que es producida por los músculos, los cuales son captados por los 8 sensores mioeléctricos que posee el brazalete Myo Armband, estos movimientos del antebrazo se logran con mucha facilidad; los cuales se traducen en instrucciones concretas para el Arduino y controlar una interfaz sin necesidad de cables, ya que la comunicación es vía bluetooth.
- Para el control de giro de los motores DC que posee el brazo robótico se utilizó unos módulos puente H L298, que nos permite que el motor gire en ambos sentidos (avance y retroceso).
- Para energizar todo el sistema de control se desarrolló una fuente de alimentación de 5 y 9 voltios independientes y una corriente de 1A, para lo cual se realizó algunos cálculos para el dimensionamiento de los

capacitores dependiendo de la carga que se va a utilizar, ya que el motor DC consume 250mA.

- Se desarrolló distintas placas electrónicas para la ejecución de este proyecto una de ellas fue la placa de indicación de los movimientos que nos ayudan a tener una visualización del movimiento que se está ejecutando en el brazo robótico, otra placa que se realizó fue un divisor de voltaje para limitar el voltaje de salida del pin TX de la placa Arduino UNO y una base para el Bluetooth HM-11 utilizando la máquina CNC del laboratorio de instrumentación virtual.
- La distancia promedio de la ejecución de los movimientos del brazo robótico es de 3 metros, ya que para manipular al brazo robótico el usuario tiene que tener línea de vista al brazo para poder manipularlo correctamente.
- Actualmente la página oficial del brazalete Myo Armband “ThalmicLabs”, se encuentra restringida debido a que el dispositivo Myo se encuentra sujeta a mejoras tanto en su diseño físico como en su diseño estructural, por lo que el usuario del brazalete Myo solo puede reportar errores de su dispositivo o tener un soporte técnico por parte de la página ThalmicLabs.

5.1.2 Recomendaciones

- Para la correcta carga del Firmware en el Bluetooth HM-11 se debe conectar los pines de alimentación del Bluetooth a la tarjeta Arduino UNO, ya que en un principio se conectó a una fuente externa de 3.3 V y generó un error de carga del Firmware.
- Antes de la manipulación del brazo robótico, es muy importante realizar la calibración con el gesto de Bloqueo / Desbloqueo, para que de esa manera evitemos fallas en la lectura de los movimientos que realicemos con el antebrazo.
- En caso de que el brazalete Myo Armband no se ajuste al antebrazo utilizamos los clips que vienen con el brazalete, esta acción permite que los sensores mioeléctricos del brazalete Myo Armband se ajusten adecuadamente y obtengan lecturas precisas y no erróneas.

BIBLIOGRAFÍA

- 5hertz. (22 de Octubre de 2014). 5hertz. Recuperado el 13 de Diciembre de 2018, de https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=2
- Aran. (23 de Enero de 2012). CochesRc. Recuperado el 13 de Diciembre de 2018, de CochesRc: <http://www.cochesrc.com/giroscopo-ii-giroscopos-electronicos-a1807.html>
- Cardona, H. J. (2 de Abril de 2018). Hospital Juan Cardona. Recuperado el 11 de Diciembre de 2018, de Hospital Juan Cardona: <http://www.hospitaljuancardona.es/blog/electromiograf%C3%ADa-tipos-de-pruebas-diagn%C3%B3sticas>
- Concepto.de. (Diciembre de 2017). Concepto.de. Recuperado el 14 de Diciembre de 2018, de Concepto.de: <https://concepto.de/fuente-de-alimentacion/>
- DALCAME. (s.f.). DALCAME Grupo de Investigación Biomédica. Recuperado el 11 de Diciembre de 2018, de <http://dalcame.com/emg.html#.XBawCGhKjIU>
- Dea. (21 de Abril de 2015). Geekbot Electronics. Recuperado el 14 de Diciembre de 2018, de Geekbot Electronics: <http://www.geekbotelectronics.com/motores-de-dc/>
- developer, A. (2016). ARM developer. Recuperado el 13 de Diciembre de 2018, de ARM developer: <https://developer.arm.com/products/processors/cortex-m/cortex-m4>
- Diego Camacho, M. E. (06 de JUNIO de 2016). Control de un Brazo Robótico LYNX AL5D Empleando Electromiografía a Través de la Tecnología MYO ARMBAND. PAPER. LATACUNGA, ECUADOR. Recuperado el 30 de agosto de 2018, de http://webltga.espe.edu.ec/sg-innovate/uploads/Manipulacion_electromiograf%C3%ADa.pdf
- Donovan, J. (1 de Diciembre de 2011). Digi-Key Electronics. Recuperado el 14 de Diciembre de 2018, de Digi-Key Electronics.:

<https://www.digikey.com/es/articles/techzone/2011/dec/bluetooth-goes-ultra-low-power>

E-Marmolejo., R. (1 de Diciembre de 2017). HETPRO. Recuperado el 6 de Febrero de 2018, de HETPRO: <https://hetpro-store.com/TUTORIALES/lm317/>

Farina, D. .. (2004). Biofísica de la generación de señales EMG. IEEE xplore, p. 81-105. Recuperado el 5 de Diciembre de 2018

Francisco José Bellido Outeiriño, J. L. (Octubre de 2014). Cogiti. Recuperado el 14 de Diciembre de 2018, de <http://www.tecnicaindustrial.es/TIFrontal/a-1434-Comunicacion-inalambrica>

Gaetani, F., Zappatore, G. A., Visconti, P., & Primiceri, P. (2 de julio de 2018). IEEE XPLORE. Recuperado el 30 de agosto de 2018, de <https://ieeexplore.ieee.org/document/8399787/>

HUITZIL-VELASCO, I. P.-C.-A. (Diciembre de 2017). Revista de Ciencias Ambientales y Recursos Naturales. Recuperado el 6 de Febrero de 2018, de Revista de Ciencias Ambientales y Recursos Naturales: https://www.ecorfan.org/spain/researchjournals/Ciencias_Ambientales_y_Recursos_Naturales/vol3num10/Revista_de_Ciencias_Ambientales_y_Recursos_Naturales_V3_N10_6.pdf

Ivan, C. (Enero de 2017). Universidad Veracruzana Reposorio institucional. Recuperado el 11 de Diciembre de 2018, de Universidad Veracruzana Reposorio institucional: <https://cdigital.uv.mx/bitstream/handle/123456789/47263/BastianChimallvan.pdf?sequence=1&isAllowed=y>

JADIAZ. (21 de Enero de 2016). MiArduino. Recuperado el 13 de Diciembre de 2018, de MiArduino: <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>

Jorge Brazeiro, S. P. (6 de Septiembre de 2015). Recuperado el 11 de Diciembre de 2018, de <http://iie.fing.edu.uy/publicaciones/2015/BPV15/BPV15.pdf>

Marmolejo, R. E. (s.f.). HeTPro. Recuperado el 13 de Diciembre de 2018, de HeTPro: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>

- Maureira, R. (24 de Diciembre de 2010). SmartDreams. Recuperado el 13 de Diciembre de 2018, de SmartDreams: <http://smartdreams.cl/unidad-de-medicion-inercial-imu/>
- Mecafenix, F. (3 de Mayo de 2017). Ingeniería mecafenix. Recuperado el 16 de Diciembre de 2018, de Ingeniería mecafenix: <http://www.ingmecafenix.com/electronica/puente-h-control-motores/>
- Molina. (s.f.). profesormolina. Recuperado el 6 de Febrero de 2018, de profesormolina: http://www.profesormolina.com.ar/electronica/componentes/fuente_reg/reg_int.htm
- Ramírez, D. (s.f.). Electrocomponentes S.A. Recuperado el 6 de Febrero de 2018, de Electrocomponentes S.A.: http://www.sase.com.ar/2011/files/2010/11/SASE2011-Fuentes_de_alimentacion.pdf
- Sathiyarayanan, M., & Rajan, S. (24 de MARZO de 2016). IEEE XPLORE. Recuperado el 29 de AGOSTO de 2018, de <https://ieeexplore.ieee.org/document/7439933/>
- SEEDSTUDIO. (s.f.). SEEDSTUDIO. Recuperado el 14 de Diciembre de 2018, de SEEDSTUDIO: http://wiki.seeedstudio.com/Bluetooth_V4.0_HM_11_BLE_Module/
- Silver, J. W. (15 de Abril de 2015). ELSEVIER. Obtenido de ELSEVIER: <https://www.elsevier.com/books/easy-emg/weiss/978-0-323-28664-0>
- ThalmicLabs. (23 de Febrero de 2013). ThalmicLabs. Recuperado el 13 de Diciembre de 2018, de ThalmicLabs: https://developer.thalmic.com/docs/api_reference/platform/getting-started.html
- ThalmicLabs. (s.f.). Myo support. Recuperado el 13 de Diciembre de 2018, de Myo support: <https://support.getmyo.com/hc/en-us/articles/202657596-Getting-starting-with-Myo-on-Windows>
- ThalmicLabs. (s.f.). Myo Support. Recuperado el 13 de Diciembre de 2018, de Myo Support: <https://support.getmyo.com/hc/en-us/articles/202657596-Getting-starting-with-Myo-on-Windows>
- Veloso, C. (6 de Mayo de 2016). ETOOLS. Recuperado el 16 de Diciembre de 2018, de ETOOLS:

<http://www.electrontools.com/Home/WP/2016/05/06/puente-h-con-driver-l298/>

W., A. (29 de Diciembre de 2015). appcessories. Recuperado el 13 de Diciembre de 2018, de appcessories: <http://www.appcessories.co.uk/thalmic-labs-myo-armband-gesture-control-remote/>

GLOSARIO DE TERMINOS

EMG: Electromiografía.

IMU: Inertial Measurement Unit (Unidad de Medición Inercial)

INTERFAZ: Capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.

LIBERIA ARDUINO: Trozos de código hechas por terceros que usamos en nuestro sketch (bosquejo). Esto nos facilita mucho la programación y hace que nuestro programa sea más sencillo de hacer y luego de entender.

FIRMWARE: Es un programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo, es el software que tiene directa interacción con el hardware.

MOTOR DC: Motor Direct Current (Motor de Corrientes Directa).

BRAZO ROBÓTICO: Un brazo robótico es un tipo de brazo mecánico, normalmente programable, con funciones parecidas a las de un brazo humano; este puede ser la suma total del mecanismo o puede ser parte de un robot más complejo.