



**UNIVERSIDAD ISRAEL**

## **TRABAJO DE TITULACIÓN**

**CARRERA:** ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**TEMA:** IMPLEMENTACIÓN DE UN BRAZO ROBÓTICO AUTÓMATA CONTROLADO POR UN SISTEMA HUMAN MACHINE INTERFACE (HMI) QUE TRANSPORTE CUBOS DE MADERA ENTRE DOS PUNTOS FIJOS, PARA LOS LABORATORIOS DE LA CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES DE LA UNIVERSIDAD ISRAEL EN LA CIUDAD DE QUITO.

**AUTOR:** CARLOS ALBERTO VÁSCONEZ ROMÁN

**TUTOR:** Ing. WILMER ALBARRACÍN Mg.

**AÑO 2015**

## UNIVERSIDAD ISRAEL

### APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**Implementación de un brazo robótico automático controlado por un sistema Human Machine Interface (HMI) que transporte cubos de madera entre dos puntos fijos, para los laboratorios de la Carrera de Electrónica y Telecomunicaciones de la Universidad Israel en la ciudad de Quito.**”, presentado por el Sr. Carlos Alberto Vásquez Román, estudiante de la carrera de Electrónica y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Abril del 2015

TUTOR

.....

Ing. WILMER ALBARRACÍN Mg.

## **UNIVERSIDAD ISRAEL**

### **AUTORÍA DEL TRABAJO DE TITULACIÓN**

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M. Abril del 2015

.....

Carlos Alberto Vásconez Román

CC: 1725752164

**UNIVERSIDAD ISRAEL**

**APROBACIÓN DEL TRIBUNAL DE GRADO**

Los miembros del Tribunal de grado, aprueban el trabajo de titulación para la graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M. Abril del 2015

Para constancia firma:

TRIBUNAL DE GRADO

.....

PRESIDENTE

.....

MIEMBRO 1

.....

MIEMBRO 2

## **AGRADECIMIENTO**

Agradezco a mi padre y madre quienes siempre me han apoyado en todo lo que he necesitado, quienes han sabido darme los mejores consejos para mi superación personal, quienes han estado en los momentos difíciles de mi vida, quienes me han dado la fuerza para poder terminar este proyecto y así culminar una etapa más en mi vida; a mis profesores que gracias a sus consejos y enseñanzas me han forjado y han permitido llenarme de conocimientos.

## **DEDICATORIA**

Dedico a mi padre y a mi madre quienes han guiado mi futuro desde un comienzo y me ha dado la fuerza necesaria para culminar con éxito una etapa más de mi vida, sobretodo me han sabido educar de la mejor manera para ser un hombre de bien y alcanzar cada meta propuesta; a mis hermanos quienes me han apoyado en todo el proceso de mi carrera y me han mantenido firme en el logro de mis objetivos; a mi familia y amigos quienes a pesar de la distancia en la que nos encontramos siempre están pendientes de mí.

## CONTENIDO

APROBACIÓN DEL TUTOR.....	II
AUTORÍA DEL TRABAJO DE TITULACIÓN .....	III
APROBACIÓN DEL TRIBUNAL DE GRADO .....	IV
AGRADECIMIENTO .....	V
DEDICATORIA .....	VI
CONTENIDO.....	VII
ÍNDICE DE FIGURAS .....	XI
ÍNDICE DE TABLAS .....	XIII
INTRODUCCIÓN .....	1
PROBLEMA INVESTIGADO .....	1
OBJETIVOS.....	2
Objetivo General .....	2
Objetivos Específicos.....	3
CAPÍTULO I.....	4
FUNDAMENTACIÓN TEÓRICA .....	4
1.1    Introducción .....	4
1.2    Marco Teórico .....	4
1.2.1    Robótica.....	4
1.2.2    Máquina Robot.....	5
1.2.3    Automatización y Robótica.....	6
1.2.4    Clasificación de los Robots .....	6

1.2.5	Tipos de Robots .....	7
1.2.6	Mecánica del brazo .....	8
1.2.7	Elemento Terminal .....	10
1.2.7.1	Sistema de Engranaje .....	10
1.2.8	Cinemática y Dinámica .....	10
1.2.9	Cinemática.....	10
1.2.9.1	Cinemática directa.....	11
1.2.9.2	Cinemática Inversa.....	11
1.2.10	Grado de Libertad .....	11
1.2.11	Dinámica .....	11
1.2.11.1	Dinámica Inversa y Directa .....	12
1.2.12	Servomotores .....	12
1.2.12.1	Fuente de alimentación ATX– 650 W .....	13
1.2.13	Microcontroladores.....	13
1.2.14	Arduino.....	14
1.2.15	HMI: Human Machine Interface.....	14
1.2.16	Labview .....	15
1.3	Marco conceptual .....	15
1.3.1	Comparación de Microcontroladores .....	15
1.3.1.1	Arduino Shield .....	17

1.3.2	Comparación de HMI .....	18
1.3.3	Comparación de Servomotores .....	20
CAPÍTULO II.....		23
BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO PARA LA IMPLEMENTACIÓN DE UN BRAZO ROBÓTICO QUE TRANSPORTE CUBOS DE MADERA ENTRE DOS PUNTOS FIJOS .....		23
CAPÍTULO III .....		25
PRESENTACIÓN DE RESULTADOS .....		25
3.1	Descripción .....	25
3.2	Análisis del diseño electrónico y estructural propuesto para la elaboración del brazo robótico automático.....	26
3.3	Implementación del brazo robótico autómatas con tecnología Arduino que transporte cubos de madera entre dos puntos fijos mediante el diseño propuesto.....	30
3.3.1	Construcción del Brazo Robótico .....	30
3.3.1.1	Elemento Terminal.....	31
3.3.1.2	Servomotores y Fuente de Poder Externa.....	32
3.3.2	Conexión Manipulador – Controlador Arduino.....	33
3.3.3	Comunicación Controlador Arduino – LabVIEW .....	34
3.4	Implementación del sistema Human Machine Interface (HMI) para el control del brazo robótico. ....	35
3.4.1	Movimientos del brazo robótico en el modo Manual.....	38

3.4.2	Movimientos del brazo Robótico en el Modo Automático .....	39
3.5	Validación del funcionamiento del sistema implementado. ....	43
3.5.1	Prueba 1 .....	43
3.5.2	Prueba 2 .....	44
3.5.3	Prueba 3 .....	45
3.6	Análisis de resultados .....	46
3.6.1	Análisis de resultados de la prueba 1 .....	46
3.6.2	Análisis de resultados de la prueba 2.....	47
3.6.3	Análisis de resultados de la prueba 3.....	48
CONCLUSIONES Y RECOMENDACIONES .....		49
Conclusiones.....		49
Recomendaciones .....		50
BIBLIOGRAFÍA.....		51
ANEXOS .....		53

## ÍNDICE DE FIGURAS

Figura 1.1. Comparación Brazo Robótico – Anatomía humana .....	9
Figura 1.2 Cadena Cinemática. ....	9
Figura 1.3 Variables dinámicas del robot. ....	12
Figura 1.4 Métodos dinámicos para un robot .....	12
Figura 1.5 Microcontrolador Arduino UNO.....	17
Figura 1.6 Shield Arduino Sensor V.5.....	18
Figura 1.7 Ventana de Inicio LabVIEW 2011.....	20
Figura 1.8 Servomotor Hitec Ms-311 .....	21
Figura 1.9 Servomotor SG90.....	22
Figura 3.1. Estructura propuesta del brazo robótico .....	26
Figura 3.2 Diseño Estructural del brazo robótico .....	28
Figura 3.3 Diagrama de flujo del sistema .....	30
Figura 3.4 Vista lateral del brazo robótico .....	31
Figura 3.5. Fuente de Alimentación ATX con salida de voltaje 5VDC .....	32
Figura 3.6. Numeración de referencia de los servomotores .....	33
Figura 3.7. Conexión Arduino Shield – Terminales de los Servomotores .....	34
Figura 3.8. Imagen Brazo Robótico colocada en la interfaz gráfica .....	35
Figura 3.9. Interfaz gráfica para el modo manual .....	36
Figura 3.10. Switch para activar el modo automático del brazo robótico .....	37
Figura 3.11. Interfaz gráfica para controlar el manipulador .....	38
Figura 3.10. Movimiento 1 base giratoria .....	39
Figura 3.11. Movimiento 2 hombro .....	40
Figura 3.12. Movimiento 3 codo .....	40

Figura 3.13. Movimiento 4 Muñeca.....	41
Figura 3.14. Movimiento 5 mano.....	42
Figura 3.15. Movimiento 6 pinza .....	42
Figura 3.16. Gráfica estadística para determinar el peso máximo a ser transportado por el brazo robótico .....	46
Figura 3.17. Gráfica estadística para determinar el tamaño de los cubos a ser transportados por el brazo robótico .....	47
Figura 3.18. Gráfica estadística para determinar el tiempo de transporte de los cubos en el modo automático .....	48

## ÍNDICE DE TABLAS

Tabla 1.1 Leyes de la Robótica. ....	5
Tabla 1.2 Elementos de un robot.....	6
Tabla 1.3 Clasificación de los Robots.....	7
Tabla 1.4 Tipos de Robots .....	8
Tabla 1.5. Especificaciones Técnicas Fuente ATX – 650W .....	13
Tabla 1.6 Tabla comparativa de los microcontroladores.....	16
Tabla 1.7 Características técnicas Arduino UNO .....	17
Tabla 1.8 Características Arduino Sensor Shield.....	18
Tabla 1.9 Tabla comparativa de HMIs .....	19
Tabla 1.10 Tabla comparativa de los servomotores.....	21
Tabla 1.11 Características técnicas del servomotor Hitec Ms-311 .....	22
Tabla 1.12 Características técnicas del servomotor SG-90 .....	22
Tabla 3.1. Rango de operación de los servomotores en el modo manual.....	39
Tabla 3.2. Cubos de madera de distinta masa transportados por el brazo robótico .....	44
Tabla 3.3. Cubos de madera de distinto tamaño transportados por el brazo robótico .....	45
Tabla 3.4. Tiempo de transporte de los cubos por el brazo robótico .....	45

## **INTRODUCCIÓN**

La evolución de la tecnología implica que el ser humano se vea obligado a modificar su forma de actuar dentro de la sociedad, la cual también se renueva conforme el avance del tiempo.

Un robot se define como una entidad virtual, un agente artificial mecánico, el cual mediante un programa es capaz de realizar tareas de forma automática. A partir de la definición expuesta, se puede establecer que las características de un robot deben ser; analizar datos, tomar decisiones y actuar sobre el entorno, ser programables. Dicho esto, es necesario tener una idea clara idea clara sobre las aplicaciones que se le puede dar a la robótica en el campo científico.

La creación del brazo robótico, una innovación que incorpora el conocimiento científico y tecnológico, está destinada a despertar el interés de estudiantes universitarios en el área de electrónica, haciéndolos formar parte del acelerado avance tecnológico.

### **PROBLEMA INVESTIGADO**

Resulta eficaz experimentar en base a los métodos científicos ya que se obtiene como respuesta el desarrollo de técnicas a través del conocimiento humano, entre dichas habilidades y destrezas se encuentra la tecnología. Es por este motivo que, dentro de los sistemas electrónicos que existen actualmente, se ha optado por elegir el dispositivo electrónico de código abierto Arduino, como la plataforma a utilizarse en la realización de este proyecto multidisciplinario.

Los Laboratorios de la Carrera de Electrónica y Telecomunicaciones de la Universidad Israel, ubicada en el Distrito Metropolitano de Quito, no cuentan con un brazo robótico automático con tecnología Arduino que transporte cubos de madera entre dos puntos fijos diseñado y elaborado en Ecuador.

En la actualidad, no se ha construido un brazo robótico automático con tecnología ARDUINO que transporte cubos de madera entre dos puntos fijos mediante un diseño elaborado por estudiantes de la Universidad Israel, por lo tanto, la creación de este proyecto despertará la iniciativa, la creatividad y la investigación de los estudiantes en el campo de la Electrónica y la Robótica.

## **OBJETIVOS**

### **Objetivo General**

- Implementar un brazo robótico autómata controlado por un sistema Human Machine Interface (HMI) que transporte de cubos de madera entre dos puntos fijos, para los Laboratorios de la Carrera de Electrónica y Telecomunicaciones de la Universidad Israel en Quito.

## **Objetivos Específicos**

- Analizar el diseño electrónico y estructural propuesto para la elaboración del brazo robótico automático.
- Implementar un brazo robótico autómata con tecnología ARDUINO que transporte cubos de madera entre dos puntos fijos mediante el diseño propuesto.
- Implementar un sistema Human Machine Interface (HMI) para el control del brazo robótico que cumpla con los requerimientos necesarios para el funcionamiento del mismo.
- Validar el correcto funcionamiento del sistema implementado.

# **CAPÍTULO I**

## **FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

Para la elaboración del presente proyecto, se debe tomar en cuenta algunos conceptos y fundamentos teóricos necesarios para la implementación del brazo robótico autómatas que transporte cubos de madera entre dos puntos fijos, y de esta manera se pueda comprender el objeto de la implementación del mismo.

Se concertará cuales son los elementos necesarios para un correcto funcionamiento a través del diseño electrónico y de software de programación propuesto para el presente proyecto.

### **1.2 Marco Teórico**

#### **1.2.1 Robótica**

De una forma u otra, la electricidad interviene en la mayoría de los aspectos de nuestra vida y, a medida que avanza la tecnología se hace cada vez más imprescindible. La robótica es una ciencia aplicada orientada a campos como diseño electrónico y mecánico, control, programación, electrónica, entre otros. (OLLERO B., 2001, pág. 1)

En la tabla 1.1 se muestran las leyes que Isaac Asimov, escritor y bioquímico de origen ruso estableció como medida de protección para los humanos, ante la posibilidad de una posible conspiración de las máquinas contra sus creadores. (Vásconez Carlos, 2015)

<b>Leyes de la Robótica</b>		
Isaac Asimov: “la imagen de un robot es la de una máquina bien diseñada y con una seguridad garantizada, que actúa de acuerdo con tres principios”	Primera Ley	Un robot no debe operar contra un ser humano o, mediante la pasividad, permitir que un ser humano sufra daños
	Segunda Ley	Un robot debe obedecer las órdenes dadas por los seres humanos, salvo que estén en conflicto con la primera ley.
	Tercera Ley	Un robot debe salvaguardar su objetividad, a no ser que éste entre en conflicto con las dos primeras leyes

Tabla 1.1 Leyes de la Robótica.  
Fuente: (OLLERO B., 2001, pág. 1)

La robótica se caracteriza por el desarrollo de sistemas cada vez más versátiles, flexibles y adecuados, mediante el uso de nuevos métodos de control y estructuras mecánicas. Entonces, se la puede definir como “el estudio de la construcción, ensamblaje, generación, programación y uso de los robots y autómatas en general”. (OCEANO, 1995, pág. 100)

### **1.2.2 Máquina Robot**

Se puede considerar a un robot como una máquina complementada con un computador, con dispositivos de entrada y salida sofisticados. Una definición aceptada de un robot está asociada a la existencia de un dispositivo de control digital, mediante la ejecución de un programa almacenado en memoria. En la tabla 1.2 se exhiben los tres elementos claves a tener en cuenta para el diseño y construcción de un robot. (OLLERO B., 2001, págs. 5-9)

<b>Componentes del robot</b>	<b>Descripción</b>
Programabilidad	Lo que significa disponer de capacidades de operación lógica y de gestión de símbolos (el robot es un computador).
Capacidad Mecánica	Que lo habilita para realizar funciones en su entorno y no ser un simple procesador de datos (el robot es una máquina).
Flexibilidad	Que el robot puede operar según un amplio rango de lenguajes de programación y tratar con determinado material u objetos de formas distintas.

Tabla 1.2 Elementos de un robot  
Fuente: (OLLERO B., 2001, págs. 5-9)

### **1.2.3 Automatización y Robótica**

Automatización y robótica son dos tecnologías relacionadas estrechamente. Las necesidades actuales de aumentar la producción y obtener productos de alta calidad, provocan que industrias busquen la manera de automatizar el trabajo basándose en sistemas complejos. En términos industriales se puede definir a la automatización como una tecnología relacionada con el desarrollo de sistemas mecánicos, electrónicos y control de producción. (ANGULO USATEGUI, 2000, pág. 5)

### **1.2.4 Clasificación de los Robots**

Las características operacionales en el proceso lógico del robot determinan la ganancia y ventajas dentro de las restricciones del diseño y el funcionamiento de los sensores. En la tabla 1.3 los robots han sido encasillados y descritos brevemente de acuerdo a su nivel de razonamiento, control, y programación. (BARRIENTOS A., PEÑÍN L., BALAGUER C., RAFAEL ARACIL., 1997, págs. 9-19)

<b>Robots</b>	<b>Descripción</b>
Play-back	Regeneran una secuencia de instrucciones grabadas. Estos robots comúnmente tienen un control de lazo abierto en cinemática.
Controlados por sensores	Se basan en los datos obtenidos por sensores o dispositivos externos y su control es mediante lazo cerrado.
Controlados por visión	Pueden manipular un objeto al utilizar información desde un sistema de visión.
Controlados adaptablemente	Sus instrucciones y rutinas son reprogramadas en base a los datos de los sensores para así obtener nuevas acciones de forma automática. Estos robots comúnmente tienen un control de lazo cerrado.

Tabla 1.3 Clasificación de los Robots

Fuente: (BARRIENTOS A., PEÑÍN L., BALAGUER C., RAFAEL ARACIL., 1997, págs. 9-19)

### 1.2.5 Tipos de Robots

Los robots, sobretodo los industriales son de varios tamaños y diferente configuración, haciendo referencia a la forma física que se le da a los brazos de robot. (ANGULO USATEGUI, 2000, pág. 12)

A continuación, en la tabla 1.4 se expone algunos tipos, (cartesiano, cilíndrico, polar, de brazo articulado, Antropomórfico, Poliarticulado o mixto) sobre todo los más importantes: (ANGULO USATEGUI, 2000, págs. 12-15)

<b>Tipo de Robot</b>	<b>Descripción</b>
Cartesiano	Utiliza tres dispositivos deslizantes perpendiculares entre sí, siendo que su movimiento en el espacio sea de tipo lineal.
Cilíndrico	Se basa en una columna vertical que gira sobre la base. Sus articulaciones son lineales en el movimiento de altura y radio.
Polar	Cuenta con dos articulaciones rotacionales y una lineal. Utiliza un brazo telescópico que puede oscilar en torno a un eje horizontal.
De brazo Articulado	Una columna que gira sobre su propia base con tres articulaciones rotacionales.
Antropomórfico	Constituido por dos componentes rectos que simulan el brazo o antebrazo humano sobre una columna giratoria.
Poliarticulado o Mixto	Diseñados para que su movimiento sea en un determinado espacio de trabajo. Posee varias articulaciones y es una combinación de los anteriores mencionados.

Tabla 1.4 Tipos de Robots

Fuente: (ANGULO USATEGUI, 2000, págs. 12-15)

### 1.2.6 Mecánica del brazo

Debido a su similitud con las extremidades superiores del cuerpo humano, los manipuladores también son denominados Brazos de robot o Brazos Robóticos. (Bueno, 2009)

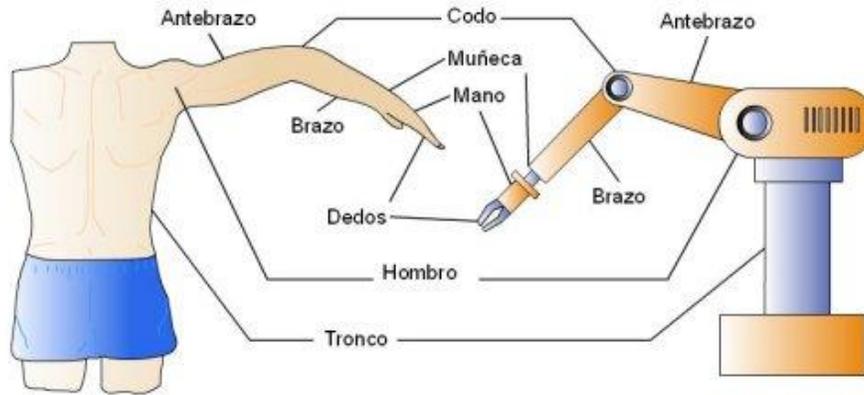


Figura 1.1. Comparación Brazo Robótico – Anatomía humana  
 Fuente: (Bueno, 2009)

Los robots también conocidos como manipuladores son por naturaleza brazos articulados, es decir, es una cadena cinemática abierta formada por una serie de eslabones concatenados interna y directamente a través de articulaciones. (Bueno, 2009)

En la figura 1.2 los eslabones  $C_0, C_1, \dots, C_n$  representan el número de eslabones que componen el manipulador. Una Articulación puede ser lineal, si un eslabón se desliza sobre un eje adherido al eslabón anterior; y rotacional, si un eje gira en torno a un eje adherido al eslabón anterior. “Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones” (OLLERO B., 2001, pág. 17).

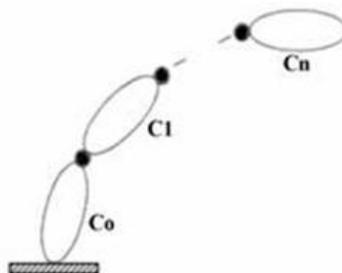


Figura 1.2 Cadena Cinemática.  
 Fuente: (OLLERO B., 2001, pág. 17)

### **1.2.7 Elemento Terminal**

En el extremo final del manipulador se conecta un elemento terminal que servirá para que se realice una aplicación en particular. Es decir que el manipulador será diseñado específicamente para dicha función. El punto central del elemento terminal se denomina punto terminal. Si fuera una pinza, su punto terminal sería el centro de represión de ésta. (Bueno, 2009)

#### **1.2.7.1 Sistema de Engranaje**

La transmisión del movimiento se da por un sistema de engranaje que se lo realiza en una forma directa, transmitiendo exactamente la misma potencia. El tipo que utiliza la pinza del brazo robótico es un sistema de engranaje con ejes paralelos, conocidos también como ejes rectos. Traspasando el movimiento de un eje a otro sin pérdida o aumento de potencia, por eso sólo se transmite. (ANGULO USATEGUI, 2000, pág. 35)

### **1.2.8 Cinemática y Dinámica**

Para poder controlar y determinar el estado de un manipulador se debe establecer la posición del punto terminal (o cualquier otro punto del brazo) en relación a un sistema de coordenadas externo y fijo. Y además analizar el movimiento del brazo cuando los elementos (articulaciones, eslabones, actuadores) aplican sus fuerzas y momentos, tanto desde la parte de la cinemática como de la dinámica. (SPONG, 1989, págs. 57,187)

### **1.2.9 Cinemática**

La Cinemática en los manipuladores se basa en las propiedades geométricas y temporales del movimiento del brazo articulado. A partir de dichos parámetros geométricos se

especifica la posición y orientación del manipulador mediante sistemas de referencia externos y objetos del entorno. “La cadena de cinemática abierta consiste cuando una secuencia de eslabones conecta los extremos de la misma. Mientras que la cinemática cerrada es cuando la secuencia de eslabones forma una trayectoria cerrada”. (SPONG, 1989, págs. 57-60)

#### **1.2.9.1 Cinemática directa**

Determina la posición y orientación del punto terminal del manipulador, con respecto a un sistema de coordenadas de referencia, conocidos los ángulos de las articulaciones y los parámetros geométricos de los demás elementos del brazo. (SPONG, 1989, págs. 61-65)

#### **1.2.9.2 Cinemática Inversa**

Determina la alineación que debe adoptar el manipulador para una posición y orientación del punto terminal conocido. (SPONG, 1989, pág. 79)

#### **1.2.10 Grado de Libertad**

El grado de libertad se refiere a cada una de las coordenadas independientes que son necesarias para analizar el estado mecánico del robot. En la cinemática abierta, cada par de eslabón – articulación tiene un sólo grado de libertad, ya sea de rotación o traslación. (SPONG, 1989, pág. 79)

#### **1.2.11 Dinámica**

La dinámica en la robótica es fundamental para el diseño de las leyes de control apropiadas para el robot y para la evaluación del diseño y estructura del brazo, utilizando

formulaciones matemáticas que analizan el comportamiento del movimiento del brazo mediante variables dinámicas que imponen el movimiento del robot como se muestra en la figura 1.3. (Abadía, 1997, pág. 40)

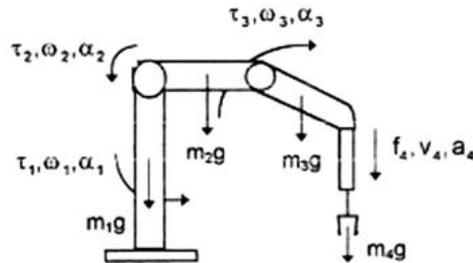


Figura 1.3 Variables dinámicas del robot.  
Fuente: (Abadía, 1997, pág. 40)

### 1.2.11.1 Dinámica Inversa y Directa

Los métodos dinámicos consisten en dos clases de soluciones: la dinámica inversa y la directa. La dinámica inversa consiste en obtener las fuerzas y torques actuantes en cada elemento y articulación del mecanismo, mientras que la dinámica directa calcula las velocidades y aceleraciones bajo condiciones de carga. (Abadía, 1997, pág. 41)

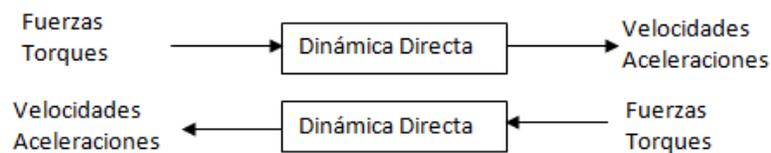


Figura 1.4 Métodos dinámicos para un robot  
Fuente: (Abadía, 1997, pág. 41)

### 1.2.12 Servomotores

Un servomotor es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. (labelec, 2009)

El control de un servo se reduce a indicar su posición mediante una señal cuadrada de voltaje. Para bloquear el servomotor en una posición es necesario enviarle continuamente una señal, de esta forma el servo conservará su posición y se resistirá a fuerzas externas que traten de cambiarlo de posición. Generalmente el rango de giro de un servomotor cubre entre 90° y 180° de la circunferencia total. (labelec, 2009)

### 1.2.12.1 Fuente de alimentación ATX– 650 W

Esta fuente de alimentación es un dispositivo que convierte el voltaje alterno a la red de suministros, en una o varias tensiones, habitualmente continuas, que alimentan los distintos circuitos del dispositivo electrónico al que se conecta.

	<b>Voltaje (V)</b>	<b>Corriente Continua (A) Carga Máxima</b>	<b>Frecuencia (Hz)</b>
<b>AC INPUT:</b>	110-220	30	50
<b>DC OUTPUT:</b>	5	40	
	12	24	
	3,3	3	
	(-)12	50	
	5	2	

Tabla 1.5. Especificaciones Técnicas Fuente ATX – 650W  
Fuente: Autor

### 1.2.13 Microcontroladores

El microcontrolador es un circuito integrado compuesto por tres partes fundamentales, la unidad central de proceso, memoria y unidades de entrada/salida, que en conjunto forman una microcomputadora, la cual necesita de un programa para realizar algún proceso concreto. (Canto, 2012)

Los microcontroladores leen y ejecutan los programas que el usuario escribe dentro de un programador, es por esto que la programación es de vital importancia cuando se diseñan sistemas con estos dispositivos electrónicos. (Canto, 2012)

#### **1.2.14 Arduino**

Arduino es una plataforma de Hardware que trabaja mediante código abierto, basada en una sencilla placa con entradas y salidas, y un entorno de desarrollo que implementa un lenguaje de programación de fácil manejo para el usuario, sin embargo, su condición de sistema libre ha propiciado tantas variaciones del mismo, que Arduino no es una pieza de hardware única. (Torrente, 2010)

Las funciones de Arduino pueden resumirse en tres: Primero, se tiene una interfaz de entrada, que puede estar unida a los periféricos, o conectarse a ellos por medio de puertos. Segundo, el microcontrolador, que es la pieza encargada de procesar los datos o información que la interfaz de entrada envía. Este microcontrolador varía debido a las necesidades del proyecto. Por último, está la interfaz de salida, la cual se encarga de transportar la información procesada a los periféricos encargados de mostrar la versión final de los datos. (Torrente, 2010)

#### **1.2.15 HMI: Human Machine Interface**

HMI es la abreviación en inglés de Interfaz Hombre Máquina, estos sistemas se los considera como “ventanas” de un proceso. Esta ventana puede estar en dispositivos especiales como paneles de operador o en una computadora. (Rojas, 2012)

Esta interfaz permite que el usuario, opere y maneje la máquina, observe el estado del equipo e intervenga en el proceso.

La información se proporciona por medio de paneles de control con señales luminosas, campos de visualización o botones, o por medio de un software que utiliza un sistema de visualización que se ejecuta en una terminal. (Rojas, 2012)

### **1.2.16 Labview**

LabView es un software Comercial propiedad de National Instruments, que tiene un entorno de programación gráfico, es decir se basa en un lenguaje “G”, el cual permite que los programas no se escriban, se dibujen. Es decir que este programa literalmente traslada los algoritmos en una forma gráfica, haciendo de esto una sencilla programación. (Sánchez A, 2010)

La elaboración de un sistema automatizado mediante el software LabVIEW requiere del uso de diferentes tipos de mecanismos, algunas de sus aplicaciones son la creación de software de control de procesos, y además la conexión de controladores y actuadores, sensores, motores, servomotores y demás. (Sánchez A, 2010)

## **1.3 Marco conceptual**

### **1.3.1 Comparación de Microcontroladores**

En la tabla 1.6 se muestran las ventajas y desventajas de tres tipos de microcontroladores:

<b>Microcontrolador</b>	<b>Ventajas</b>	<b>Desventajas</b>	<b>Valoración (1-5)</b>
ARDUINO	<ul style="list-style-type: none"> <li>• Lenguaje de programación simplificado.</li> <li>• Posee extensiones de Hardware para conexiones externas</li> <li>• Compilación y depuración directa del Programa al microcontrolador</li> </ul>	<ul style="list-style-type: none"> <li>• Costo de adquisición.</li> <li>• Sensibilidad en el manejo del equipo</li> </ul>	5
AVR	<ul style="list-style-type: none"> <li>• Costo de adquisición</li> <li>• Tamaño del equipo</li> </ul>	<ul style="list-style-type: none"> <li>• Lenguaje de programación extenso.</li> </ul>	3
PIC	<ul style="list-style-type: none"> <li>• Costo de adquisición</li> <li>• Tamaño del equipo</li> </ul>	<ul style="list-style-type: none"> <li>• Lenguaje de programación extenso.</li> </ul>	2

Tabla 1.6 Tabla comparativa de los microcontroladores  
Fuente: Autor

A través de la tabla 1.6 se comparan tres tipos de microcontroladores. Arduino tiene la mejor valoración de entre los tres debido a la facilidad de programación que tiene y las extensiones de hardware que se pueden adicionar al dispositivo para conexiones externas obteniendo mayores funcionalidades. Es por esto que se ha elegido a la plataforma Arduino para el desarrollo de la implementación del presente proyecto. (Vásconez C., 2015)



Figura 1.5 Microcontrolador Arduino UNO  
 Fuente: (Torrente, 2010)

Arduino UNO	Características técnicas
Microcontrolador	Atmega 328
Voltaje de funcionamiento	5V
Alimentación	7-12 V
Corriente DC (I/O pin)	50 mA
Memoria FLASH	32KB de los cuales 0.5KB son usados para arranque
SRAM	2KB
EEPROM	1KB
Velocidad del reloj	16Mhz

Tabla 1.7 Características técnicas Arduino UNO  
 Fuente: (Torrente, 2010)

### 1.3.1.1 Arduino Shield

Arduino Shield se encarga de mejorar mediante funcionalidades adicionales a la placa Arduino aumentando sus capacidades, por ejemplo, comunicación (con otras placas o el medio), o al momento de gestionar más sistemas. (webelectro, 2015)

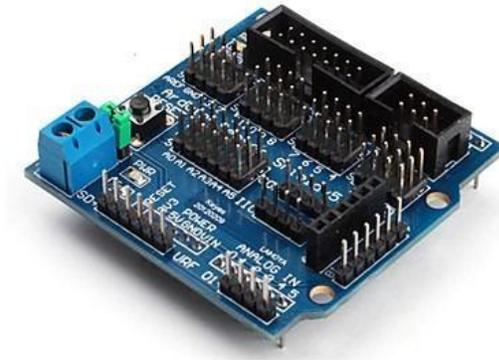


Figura 1.6 Shield Arduino Sensor V.5  
Fuente: (webelectro, 2015)

Arduino Sensor Shield V.5	Características
Controlador	Secuencia de interfaz analógica y digital
	Interfaz i2C
Interfaz	Controladora de 32 servos
	Comunicación del módulo Bluetooth
	APC220 WIRELESS
	rs232.
	Comunicación del módulo de tarjeta SD
	Comunicación APC220 inalámbrica módulo de RF.
	12864 LCD serie y paralelo
	Sensor de ultrasonidos.

Tabla 1.8 Características Arduino Sensor Shield  
Fuente: (webelectro, 2015)

### 1.3.2 Comparación de HMI

En la tabla 1.9 se muestran las ventajas y desventajas de tres tipos de HMI. Además se comparan tres tipos de HMIs. Debido a la facilidad de programación gráfica por diagrama de bloques y por la compatibilidad que tiene para trabajar conjuntamente con la plataforma

Arduino se utilizará el sistema HMI LabVIEW versión 2011 para realizar una interfaz de usuario que puede interactuar con el manipulador. (Vásquez C., 2015)

HMI	Ventajas	Desventajas	Valoración (1-5)
LabVIEW (National Instruments)	<ul style="list-style-type: none"> <li>• Interfaz para plataforma Arduino.</li> <li>• Programación gráfica por diagramas de bloques</li> </ul>	<ul style="list-style-type: none"> <li>• Se requiere Licencia de software</li> </ul>	5
LabWINDOWS (National Instruments)	<ul style="list-style-type: none"> <li>• Interfaz para plataforma Arduino.</li> </ul>	<ul style="list-style-type: none"> <li>• Costo de adquisición</li> <li>• Programación por código fuente.</li> <li>• Se requiere Licencia de software</li> </ul>	2
Blender	<ul style="list-style-type: none"> <li>• Software Libre</li> <li>• Imágenes en 3D</li> </ul>	<ul style="list-style-type: none"> <li>• Programación en Lenguaje Python</li> <li>• No cuenta con Interfaz para plataforma Arduino</li> </ul>	1

Tabla 1.9 Tabla comparativa de HMIs

Fuente: Autor

Hoy es posible diseñar sistemas de automatización y medida de bajo costo. La programación gráfica con Labview permite a los no programadores un método fácil para implementar aplicaciones complejas de test, medida y automatización. Con Labview el software define el sistema. (TutorialLabview.pdf)



Figura 1.7 Ventana de Inicio LabVIEW 2011

Fuente: (ni, 2012)

### 1.3.3 Comparación de Servomotores

En la tabla 1.10 se indican las ventajas y desventajas de tres tipos de servomotores y se compara tres tipos de servomotores, debido a sus características mecánicas y electrónicas los servomotores (Hitec Ms-311 y SG90) son los ideales para la implementación del brazo robótico. (Vásconez C., 2015)

Servomotores	Ventajas	Desventajas	Valoración (1-5)
Hitec Ms-311	<ul style="list-style-type: none"> <li>• Características mecánicas y electrónicas</li> </ul>	<ul style="list-style-type: none"> <li>• Costo de adquisición.</li> </ul>	5
SG90	<ul style="list-style-type: none"> <li>• Costo de adquisición</li> <li>• Características de precisión</li> </ul>	<ul style="list-style-type: none"> <li>• Potencia</li> </ul>	5
Hitec Hs-422	<ul style="list-style-type: none"> <li>• Características mecánicas y electrónicas</li> </ul>	<ul style="list-style-type: none"> <li>• Costo de adquisición.</li> <li>• Tamaño</li> </ul>	3

Tabla 1.10 Tabla comparativa de los servomotores  
Fuente: Autor



Figura 1.8 Servomotor Hitec Ms-311  
Fuente: (Servodatabase, 2009)

En la tabla 1.11 se indican las características técnicas del servomotor Hitec Ms-311:

<b>Servomotor Hitec Ms-311</b>	<b>Características</b>
Largo	39.9 mm
Ancho	19.8 mm
Alto	36.3 mm
Peso	43 g
Velocidad de giro 4.8 V	0.19 seg/60°
Velocidad de giro 6.0 V	0.15 seg/60°

Tabla 1.11 Características técnicas del servomotor Hitec Ms-311

Fuente: (Servodatabase, 2009)

Las características de precisión del servomotor SG90 son ideales para utilizarlo en el elemento terminal del brazo robótico. En ese punto no se requiere de una gran potencia.

(Vásconez C., 2015)



Figura 1.9 Servomotor SG90

Fuente: (Botsience, 2013)

<b>Servomotor SG90</b>	<b>Características</b>
Largo	21.8 mm
Ancho	22.6 mm
Alto	11.4 mm
Peso	19 g
Velocidad de giro	0.12 seg/60°
Voltaje de operación	4 V – 6 V

Tabla 1.12 Características técnicas del servomotor SG-90

Fuente: (Botsience, 2013)

## **CAPÍTULO II**

### **BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO PARA LA IMPLEMENTACIÓN DE UN BRAZO ROBÓTICO QUE TRANSPORTE CUBOS DE MADERA ENTRE DOS PUNTOS FIJOS**

Dentro del proceso investigado, el problema principal del presente proyecto se basa en que los Laboratorios de la Carrera de Electrónica y Telecomunicaciones de la Universidad Israel en Quito no cuentan con un brazo robótico automático con tecnología ARDUINO para transportar cubos de madera entre dos puntos fijos que pueda despertar la iniciativa, creatividad y la investigación de los estudiantes en el campo de la Electrónica y la Robótica.

Para la implementación del proyecto se propuso un objetivo general con la finalidad de elaborar un brazo robótico que transporte cubos de madera entre dos puntos fijos y así despierte la iniciativa, creatividad e investigación de los estudiantes de la Carrera de Electrónica de la Universidad Israel. Los objetivos específicos fueron propuestos para el análisis del diseño electrónico y estructural para poder implementar el brazo robótico.

La implementación de un brazo robótico en el laboratorio de la Carrera de Electrónica de la Universidad Israel despertará el interés de implementar nuevos proyectos acorde al avance tecnológico que existe en la actualidad, además se aplicarán nuevos conocimientos en la formación académica de los Estudiantes de la Carrera.

### **Variable Independiente**

Implementación de un brazo robótico en el laboratorio de la Carrera de Electrónica de la Universidad Israel

### **Variable Dependiente**

Despertar el interés de implementar nuevos proyectos acorde al avance tecnológico que existe en la actualidad.

Para el presente proyecto se utilizaron los siguientes métodos de investigación:

### **Método de Análisis y Síntesis**

El método de análisis y síntesis se utilizó para poder analizar el diseño electrónico y estructural propuesto y disponer de toda la información necesaria y útil para el sistema, que sirva para la implementación del brazo robótico.

### **Método Experimental**

Se usó el método experimental para la detección y corrección de errores y poder validar el correcto funcionamiento una vez implementado el brazo robótico.

## CAPÍTULO III

### PRESENTACIÓN DE RESULTADOS

#### **3.1 Descripción**

De acuerdo con lo analizado e investigado, para la implementación del brazo robótico se tomó en cuenta el diseño propuesto para el presente proyecto, ya que se basa en tres elementos importantes como lo son: HMI, PLATAFORMA ARDUINO Y MANIPULADOR (servomotores).

El brazo robótico será semejante a la extremidad superior del cuerpo humano, siendo capaz de realizar determinados movimientos dentro de un espacio de trabajo establecido. Al final del brazo, en la muñeca se colocará una pinza la cual sujetará y afianzará los cubos de madera para poder moverlos dentro del espacio en el que el robot trabaje.

La función principal que desempeñará el brazo robótico será la de transportar un cubo de madera de un punto a otro de forma automática controlado por una interfaz gráfica que especifique los movimientos que tendrá que realizar el brazo para poder llevar a cabo dicha tarea. Además, se puede controlar cada movimiento del brazo robótico de forma manual. Esto, con la finalidad de que el brazo obtenga mayores aplicaciones optimizando su funcionalidad.

Los movimientos automáticos del brazo robótico son programados mediante Arduino, plataforma que controla al manipulador de manera sistemática. Mediante una interfaz HMI el usuario puede especificar tareas (movimientos) adicionales que el brazo pueda ejecutar

de forma manual. Por lo cual tanto Arduino como el HMI reciben como datos de entrada, la posición angular de los servomotores para establecer el movimiento de cada parte del brazo robótico.

Mediante el código Arduino y la interfaz máquina humano Labview se ejecutan una serie de rutinas e instrucciones para poder analizar y calcular los movimientos que debe realizar el brazo robótico, en este caso a través de los servomotores implementados en los eslabones del manipulador.

El brazo robótico construido pertenece al grupo de los robots de arquitectura poliarticulada o mixta. A continuación se muestra la estructura principal para el funcionamiento del brazo:



Figura 3.1. Estructura propuesta del brazo robótico  
Fuente: Autor

### 3.2 Análisis del diseño electrónico y estructural propuesto para la elaboración del brazo robótico automático.

Para la implementación del brazo robótico controlado por un sistema Human Machine Interface que transporte cubos de madera entre dos puntos fijos se procedió al análisis

del diseño planteado por el estudiante Luis Mosquera Garcés, de la Carrera de Electrónica de la Universidad Israel.

Se observa que el diseño consta de tres etapas: Etapa de entrada, etapa de control y etapa de salida.

En la etapa de entrada se utiliza una interfaz gráfica realizada en LabVIEW, en donde el usuario a través del computador obtiene el control sobre el brazo. El software LabVIEW es el indicado a utilizar debido a la facilidad de programación gráfica por diagrama de bloques y por la compatibilidad que tiene para trabajar conjuntamente con la plataforma Arduino.

En la etapa de control se encuentra el microcontrolador Arduino UNO y el Arduino Shield conectado a los pines del mismo. Se utiliza este microcontrolador para la adaptación de interconexión entre pines para una mejora en el funcionamiento, además de guardar la programación para el control del brazo robótico.

En la etapa de salida los servomotores son ideales para la implementación del brazo robótico debido a sus características mecánicas y electrónicas tales como su capacidad de ubicarse en cualquier posición en su rango de operación útil para el movimiento de cada parte del brazo. Los servomotores a emplear son marca Hitec Ms-311 y SG90, los cuales poseen tres terminales que facilitan la conexión con la placa Arduino.

El diseño del hardware del brazo robótico es un prototipo estructural mediante medidas y dimensiones válidas para la implementación del presente proyecto. El diseño estructural del hardware del brazo robótico, es ideal para implementarlo en acrílico debido a que es un material liviano respecto a otros y beneficia a los servomotores al no tener que exigir su máxima capacidad evitando que se produzcan daños internos en los mismos. A continuación se muestra en la figura 3.2 el diagrama estructural del hardware del brazo robótico ensamblado con todas sus partes:

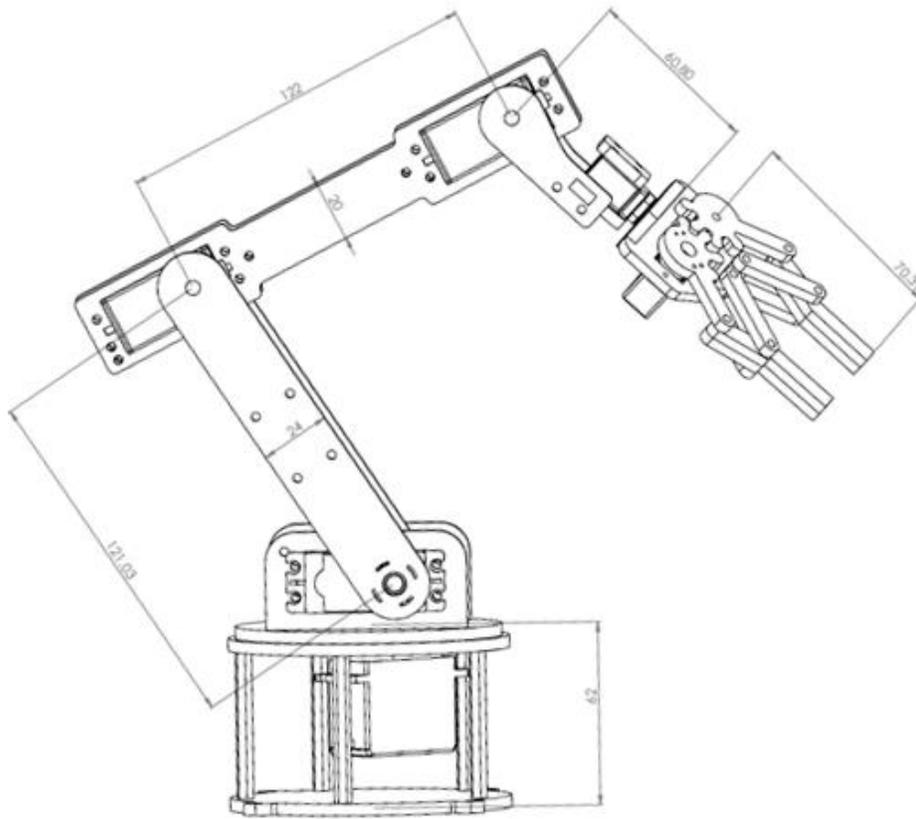


Figura 3.2 Diseño Estructural del brazo robótico

Fuente: (Mosquera Garcés, 2015)

Al final del brazo robótico se diseña una pinza la cual sujetará un cubo de madera y sostendrá el mismo al momento de transportarlo. Se considera que la pinza deberá soportar

una determinada capacidad de carga y para que al momento de transportarlo pueda sujetar adecuadamente el cubo de madera sin dejarlo caer.

Los servomotores utilizan un voltaje dc de 5V y una corriente de entre 0.5A a 1A. Debido a esto se ve que es necesario utilizar una fuente de alimentación externa de esas características.

Al analizar el flujograma de la figura 3.3 diseñada para el software del brazo robótico, se observa que el diseño cuenta con un flujo de control que permite que el código se ejecute repetidamente en base a una determinada condición. Se establecen dos condiciones para que el brazo robótico pueda ser controlado mediante dos formas. La primera condición (verdadera) para poder controlar el brazo robótico de forma manual, y la segunda condición (falsa) para que actúe de forma automática.

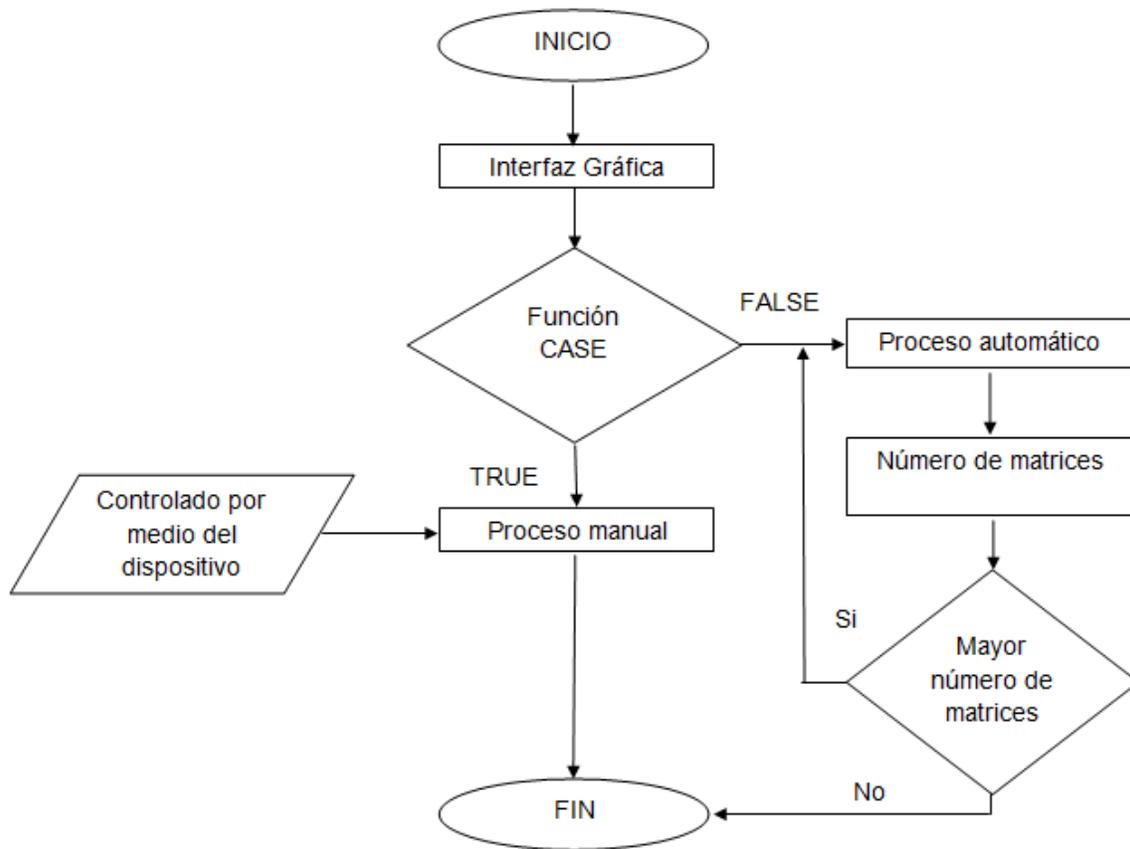


Figura 3.3 Diagrama de flujo del sistema  
Fuente: Autor

### 3.3 Implementación del brazo robótico autómatas con tecnología ARDUINO que transporte cubos de madera entre dos puntos fijos mediante el diseño propuesto.

#### 3.3.1 Construcción del Brazo Robótico

Mediante el diseño del manipulador propuesto se procede a la construcción física del mismo. Esto se lo realizó con material acrílico color azul para cada una de las partes tales como la base las articulaciones fijas y la pinza. Luego se procede a ensamblar cada parte del brazo junto con los servomotores en cada eje de los mismos para que el manipulador pueda obtener el movimiento deseado, semejante al de una extremidad superior del cuerpo

humano. A continuación se muestra en la figura 3.4 la construcción del manipulador, ensambladas sus partes junto con los servos en su respectiva posición:

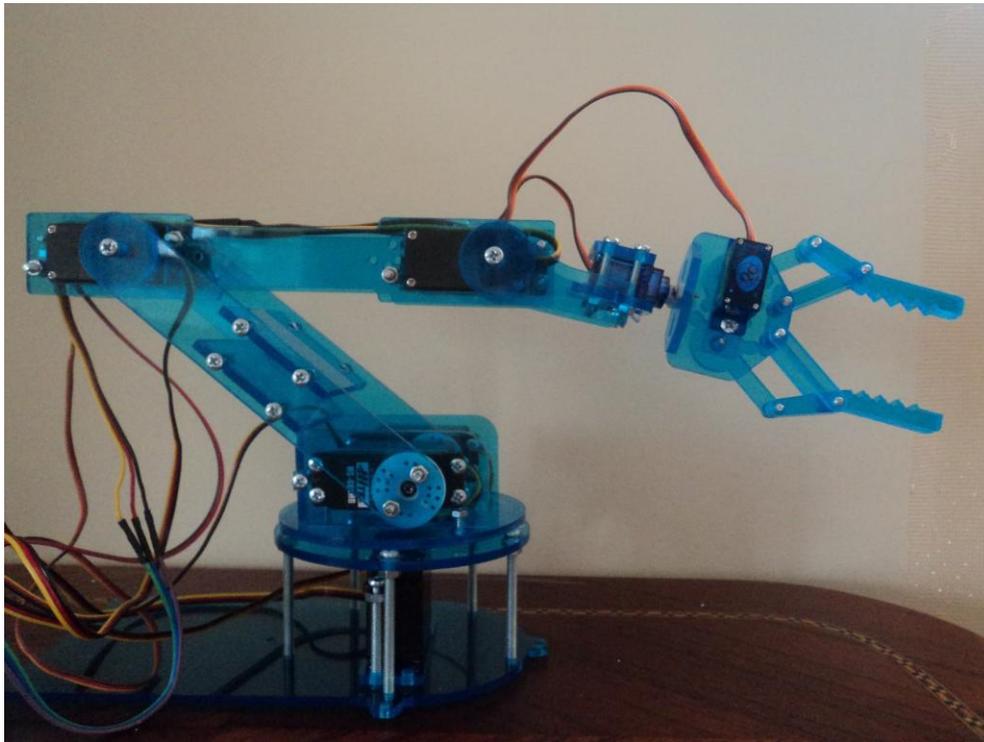


Figura 3.4 Vista lateral del brazo robótico

Fuente: Autor

### 3.3.1.1 Elemento Terminal

Para la pinza se mandó a confeccionar en material acrílico dos ganchos que simulen la forma de “dedos” y en su interior una hilera de puntas para una mejor sujeción del objeto. Se elaboró una base que conecte la pinza con la muñeca del brazo robótico (servomotor) y para sujetar el servomotor que articula los ganchos. Para dar el movimiento de abertura a los “dedos” se utiliza un sistema de dos engranes de igual tamaño en la parte inicial de los ganchos. El servomotor se conecta además con uno de los engranes en la base, dando la movilidad a dicho sistema.

Los “dedos” de la pinza se encuentran sujetos a la base mediante tornillos que permiten la movilidad de los ganchos; teniéndose una distancia máxima entre los “dedos” de 5cm, y una distancia mínima de 1cm.

### 3.3.1.2 Servomotores y Fuente de Poder Externa

La fuente externa que se implementó es una fuente de alimentación ATX de 600W con voltajes variables (figura 3.5). Los servomotores que dan el movimiento al brazo necesitan una alimentación de voltaje de 5VDC.

Esta fuente externa va conectada al igual que los servomotores en la placa Arduino Sensor Shield. Los 5VDC de la fuente de alimentación se conectan mediante el cable rojo (+) y negro (-) a la misma trama en la que se encuentran conectados los servomotores.



Figura 3.5. Fuente de Alimentación ATX con salida de voltaje 5VDC  
Fuente: Autor

### 3.3.2 Conexión Manipulador – Controlador Arduino

En la figura 3.6 se detalla el número de servomotor el cual sirve de referencia para la conexión al controlador Arduino y poder ubicarlos en los respectivos pines de entrada y salida que corresponden a su determinada instrucción en la programación de Labview 2011.

Ahora, se procede a conectar las terminales de los servomotores al controlador Arduino. Para ello se utilizó el Sensor Shield de Arduino, el cual facilita la conexión de las terminales de los servomotores permitiendo hacerlo de una manera directa.

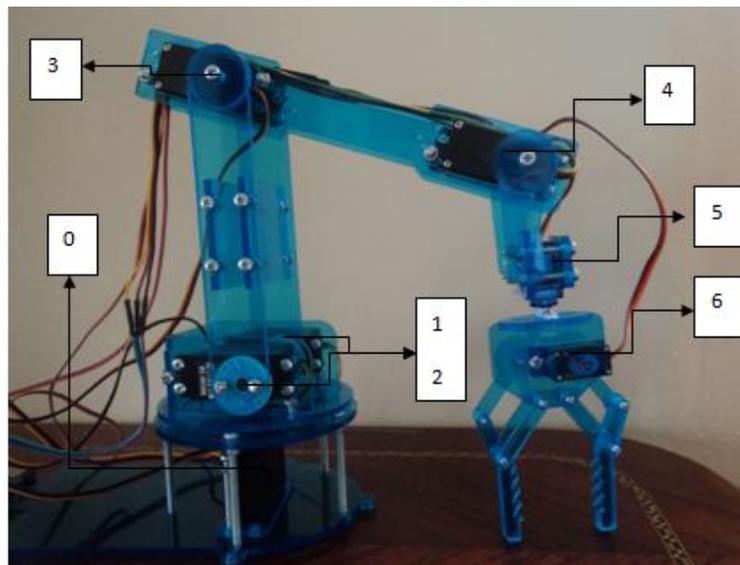


Figura 3.6. Numeración de referencia de los servomotores  
Fuente: Autor

En el Arduino Sensor Shield se encuentra una trama de conexiones de terminales de 3 pines para los servomotores como se observa en la figura 3.7.

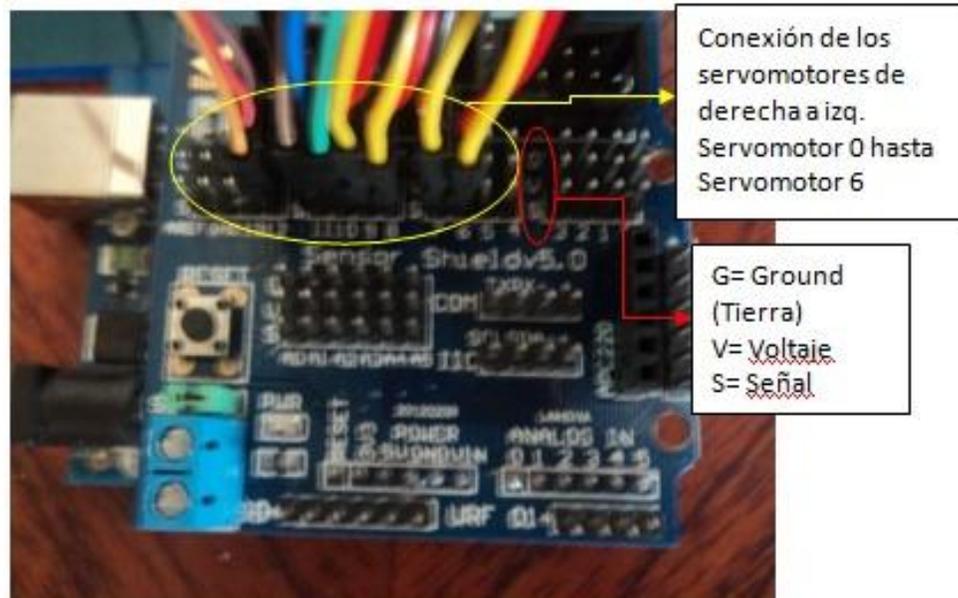


Figura 3.7. Conexión Arduino Shield – Terminales de los Servomotores  
Fuente: Autor

### 3.3.3 Comunicación Controlador Arduino – LabVIEW

A través de una interfaz USB se conecta el Arduino Uno a la computadora donde se encuentra instalado el programa Labview 2011 v11.0.

Para que Arduino trabaje y funcione correctamente desde Labview 2011 se instalan en dicho programa las herramientas funcionales y los drivers de Arduino que National Instrument (Empresa fabricante del programa Labview) proporciona. Esto se lo hace a través del Programa VI Package Manager y del programa NI VISA respectivamente.

### 3.4 Implementación del sistema Human Machine Interface (HMI) para el control del brazo robótico.

La interfaz gráfica está elaborada en el panel frontal del programa Labview que proporciona al usuario la capacidad de controlar cada parte del brazo robótico de forma manual y pueda activar el modo automático para que el manipulador realice el proceso de transportar un cubo de madera de un punto fijo a otro.

En la interfaz gráfica se coloca una imagen elaborada en el programa Solid Works donde se muestra el brazo robótico identificando cada una de sus partes. Las cuales podrán ser controladas por el usuario en el modo Manual, y así el usuario pueda saber de una manera más sencilla que parte del manipulador es la que está controlando.

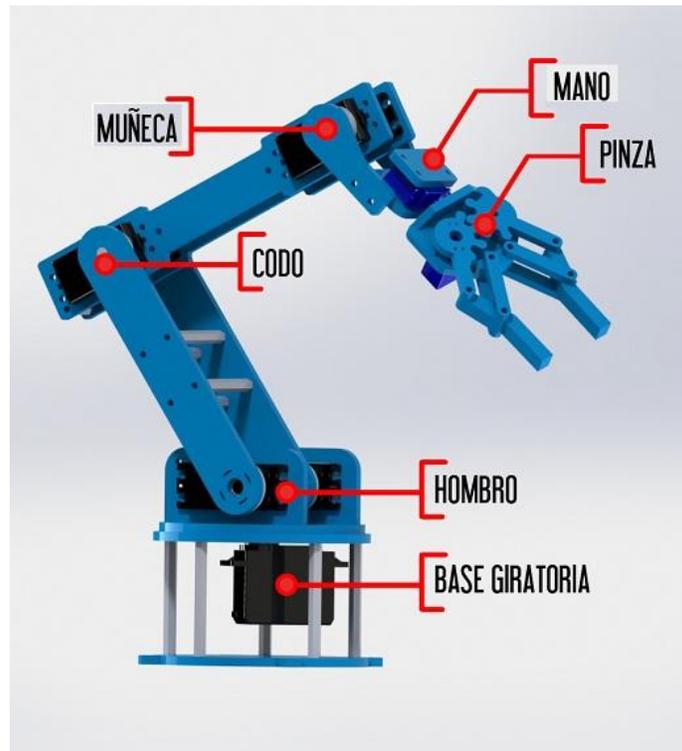


Figura 3.8. Imagen Brazo Robótico colocada en la interfaz gráfica  
Fuente: Autor

Para el control manual que podrá realizar el usuario a través de la interfaz gráfica se utilizaron deslizadores para aumentar o disminuir la posición angular de los servomotores, ubicados en cada parte del brazo y así obtener el movimiento deseado dentro del rango especificado en cada deslizador para cada parte del brazo robótico. A continuación se muestra en la figura 3.9 la interfaz gráfica en donde el usuario puede controlar manualmente el brazo robótico.

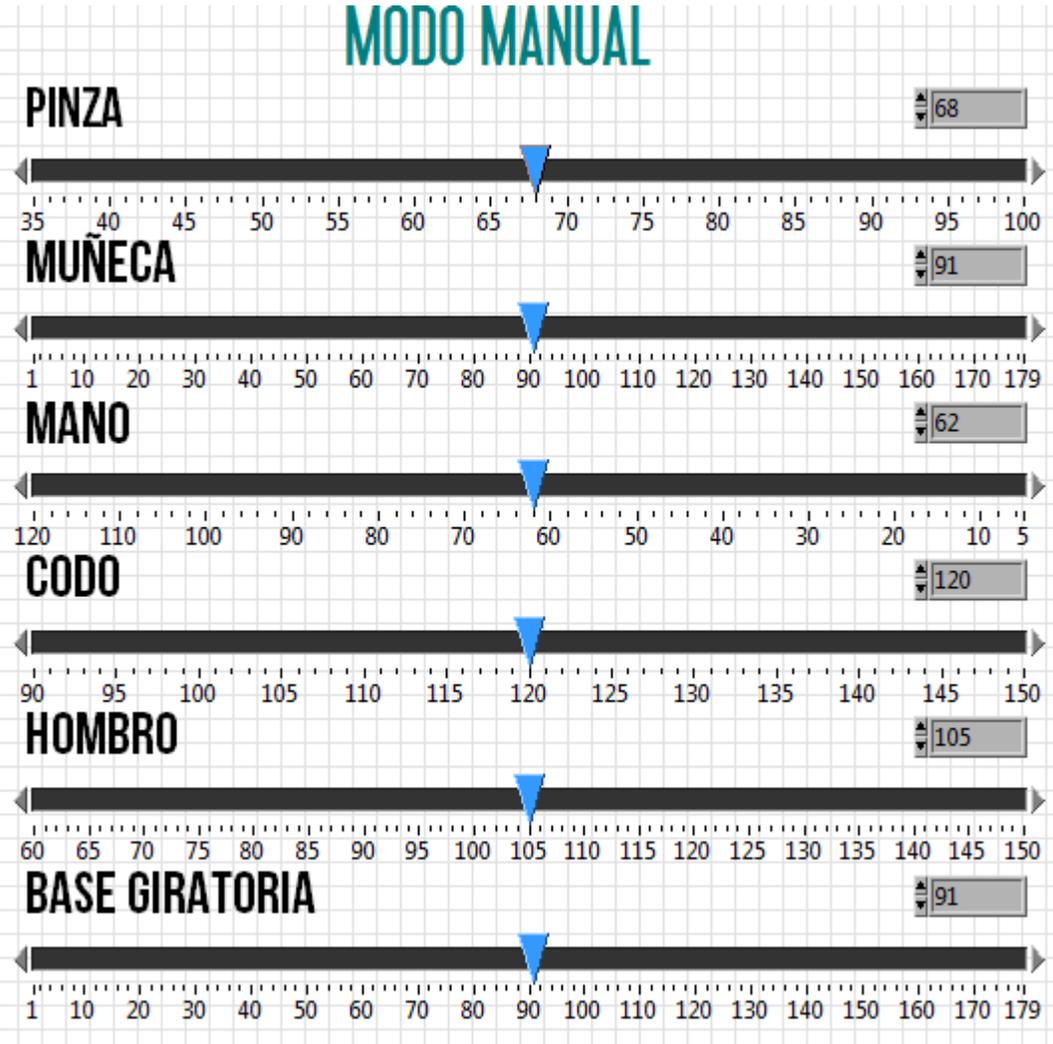


Figura 3.9. Interfaz gráfica para el modo manual  
Fuente: Autor

Para activar el modo automático se utiliza un switch deslizador, cuando está en la posición hacia arriba se encuentra en el modo manual, cuando el switch se encuentra en la posición hacia abajo el brazo robótico pasará al modo automático. Para que el usuario se dé cuenta en qué condición se encuentra el brazo robótico se colocó un indicador led, el cual estará en verde cuando esté activo el modo manual y en rojo cuando el brazo se encuentre trabajando de forma automática.

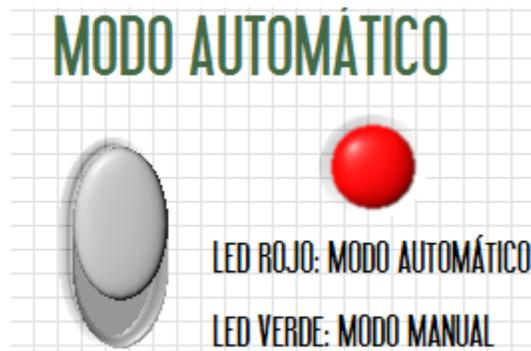


Figura 3.10. Switch para activar el modo automático del brazo robótico  
Fuente: Autor

Dentro de la interfaz gráfica se tiene un indicador de puerto COM en la parte superior derecha para que el usuario pueda saber a qué puerto de la computadora se conectó el cable USB del Arduino UNO.

Además el usuario puede detener el proceso automático si lo requiere mediante el botón STOP/RESTART ubicado en la parte inferior derecha de la interfaz gráfica. Y también detener completamente el funcionamiento del brazo a través del botón STOP o el botón de RESET directamente desde la placa Arduino.

A continuación se muestra en la figura 3.11 la interfaz gráfica; con los dos modos de funcionamiento del brazo, manual y automático.

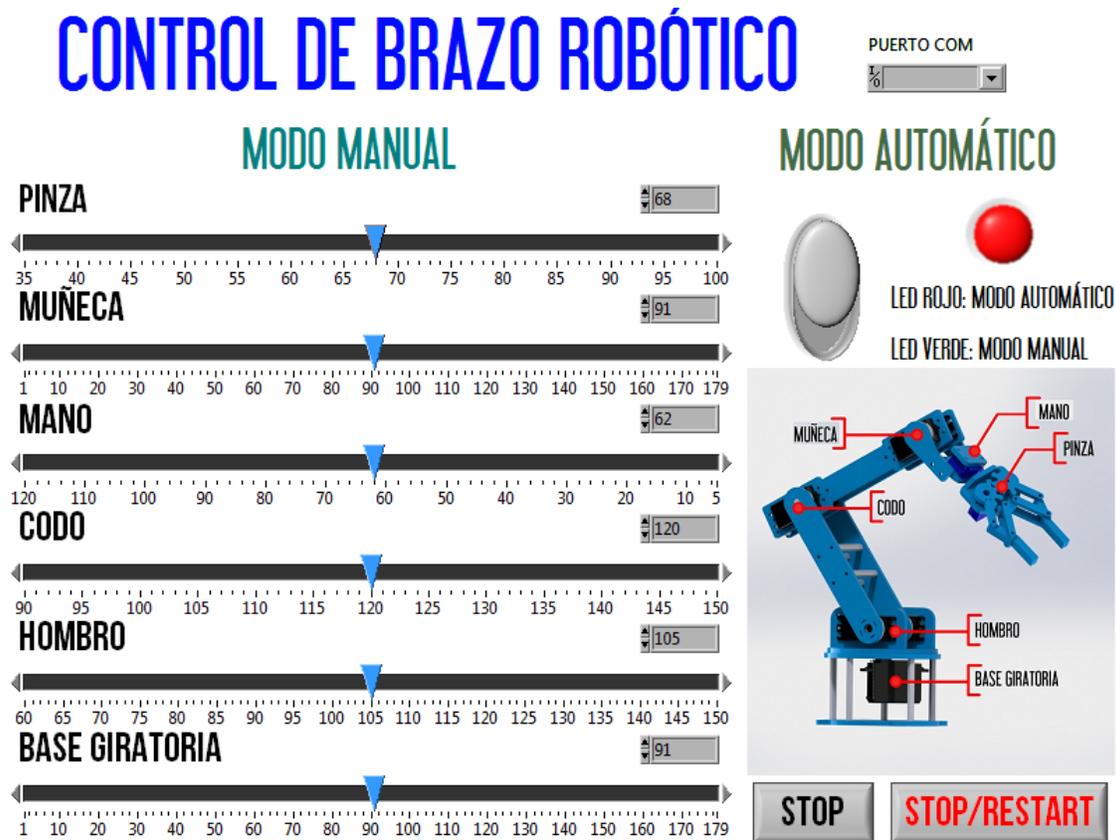


Figura 3.11. Interfaz gráfica para controlar el manipulador  
Fuente: Autor

### 3.4.1 Movimientos del brazo robótico en el modo Manual

A través de la interfaz gráfica el usuario puede manipular los movimientos de cada parte del brazo robótico dentro de los rangos establecidos. A continuación se muestra en la tabla 3.1 los rangos de posiciones angulares en los cuales cada servomotor trabaja para cada pieza del brazo robótico.

<b>PARTE DEL MANIPULADOR</b>	<b>RANGO DE OPERACIÓN (°)</b>
BASE GIRATORIA	1 – 179
HOMBRO	60 – 150
CODO	90 – 150
MUÑECA	5 – 120
MANO	1 – 179
PINZA	35 – 100

Tabla 3.1. Rango de operación de los servomotores en el modo manual

Fuente: Autor

### 3.4.2 Movimientos del brazo Robótico en el Modo Automático

**Movimiento 1 (Base giratoria):** La base realiza un giro de  $153^\circ$ . Cuando la base se encuentre en la posición  $0^\circ$ , quiere decir que el brazo está situado en el lugar donde cogerá el objeto a ser transportado, y cuando la base se encuentre en la posición de  $153^\circ$ , quiere decir que el brazo se encuentra en el lugar donde se depositará el objeto transportado. Una vez situado el objeto transportado la base regresa a la posición  $0^\circ$  para continuar con el proceso si se lo requiere.

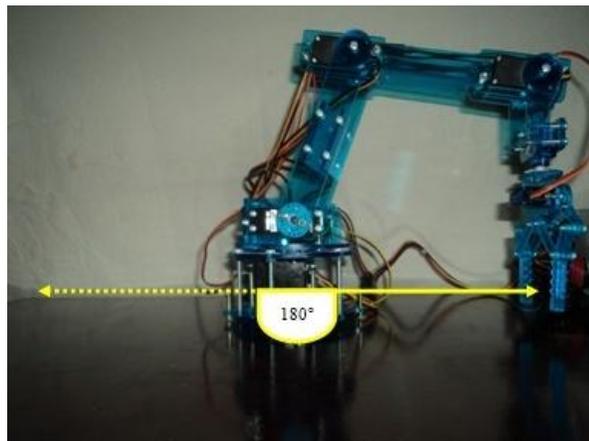


Figura 3.10. Movimiento 1 base giratoria

Fuente: Autor

**Movimiento 2 (Hombro):** El hombro del manipulador realiza un movimiento inclinado desde los  $82^\circ$  hasta los  $119^\circ$  aproximadamente en sentido horario y antihorario.

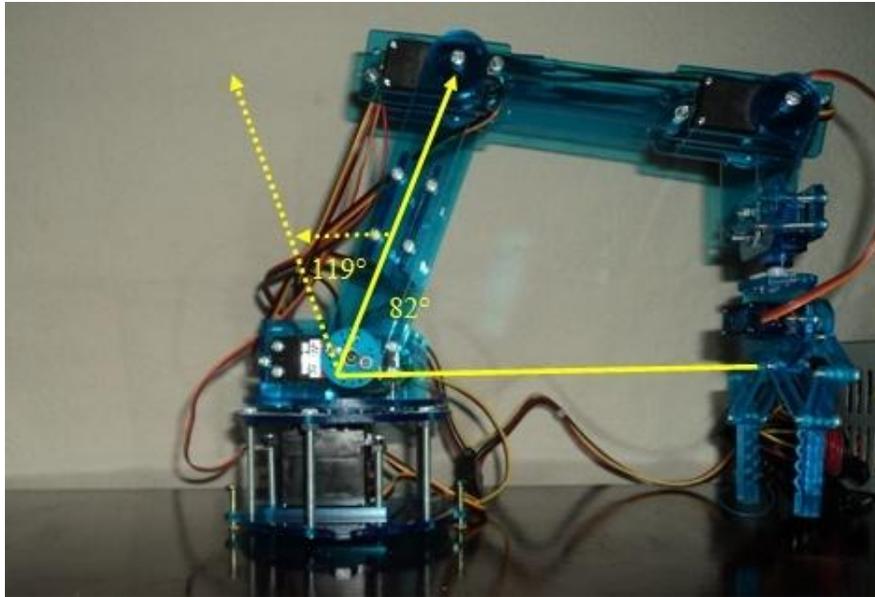


Figura 3.11. Movimiento 2 hombro  
Fuente: Autor

**Movimiento 3 (Codo):** El tercer movimiento se sitúa en la parte del codo y el hombro, el movimiento es vertical girando entre  $94^\circ$  hasta los  $122^\circ$  aproximadamente.

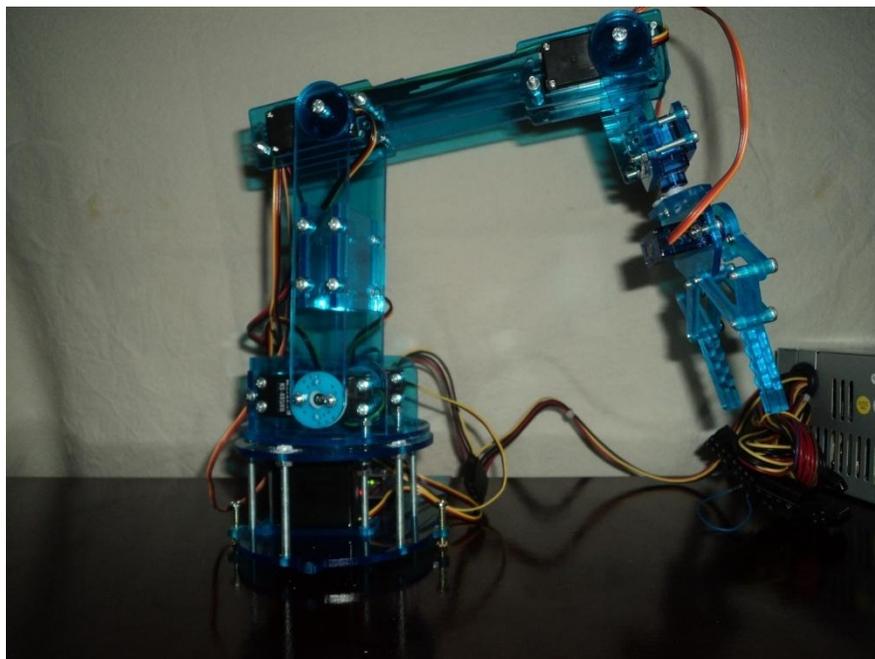


Figura 3.12. Movimiento 3 codo  
Fuente: Autor

**Movimiento 4 (Muñeca):** La muñeca también realiza un movimiento vertical desde los  $5^\circ$  hasta los  $77^\circ$ . Cuando se encuentre en los  $5^\circ$ , quiere decir que la pinza debe encontrarse lista para agarrar el objeto. Cuando se encuentra en los  $77^\circ$  la pinza debe encontrarse lista para depositar el objeto en el lugar fijo al cual se deseaba transportarlo.

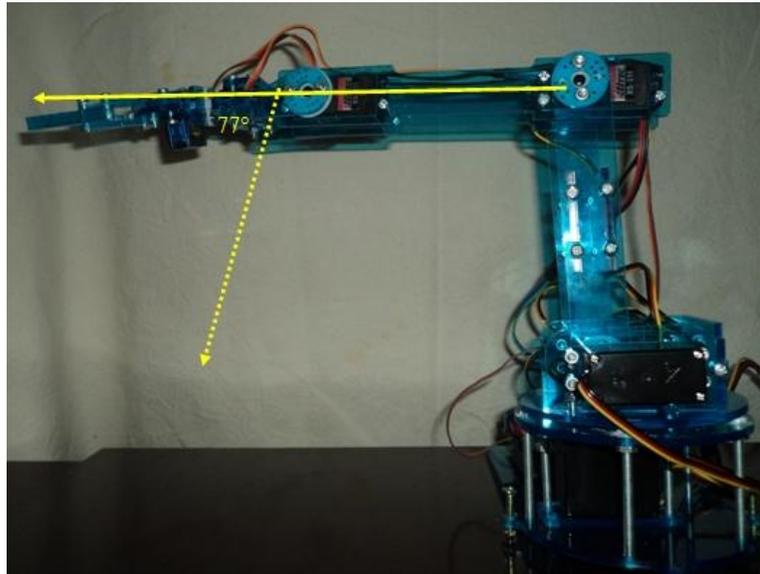


Figura 3.13. Movimiento 4 Muñeca  
Fuente: Autor

**Movimiento 5 (Mano):** Como se observa en la figura 3.13 la mano realiza un movimiento rotacional de aproximadamente  $20^\circ$  entre su rango de operación para lograr un mejor agarre del objeto por parte de la pinza.

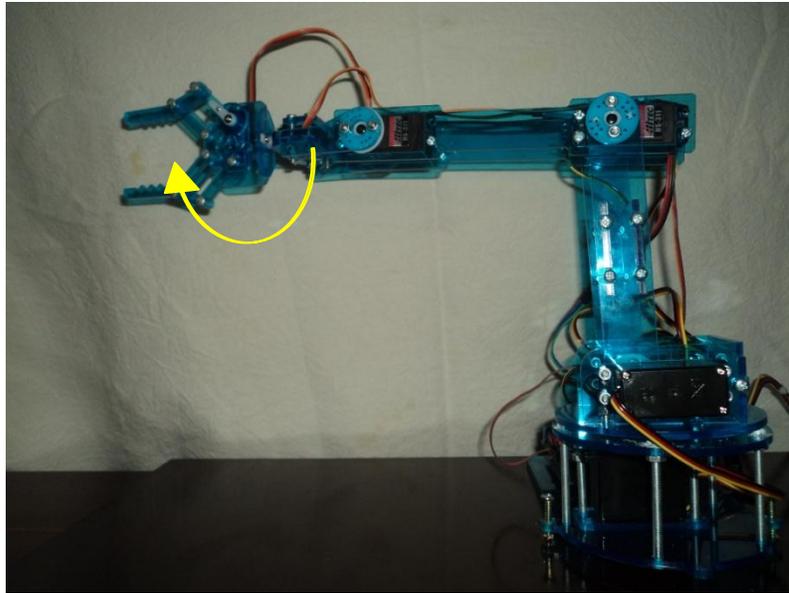


Figura 3.14. Movimiento 5 mano  
Fuente: Autor

**Movimiento 6 (Pinza):** El movimiento se lo realiza mediante el sistema de engranes haciendo que la pinza se abra y se cierre de forma horizontal y el servomotor de dicha parte se mueve entre los  $35^\circ$  y  $89^\circ$  aproximadamente.

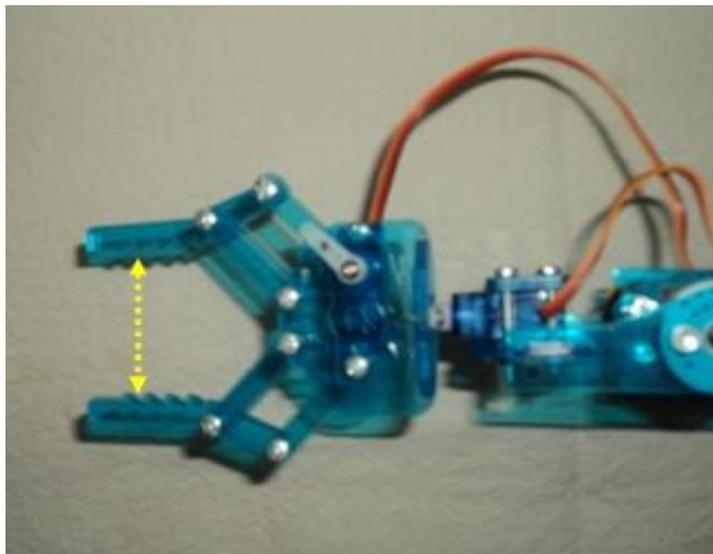


Figura 3.15. Movimiento 6 pinza  
Fuente: Autor

### **3.5 Validación del funcionamiento del sistema implementado.**

Para la validación del funcionamiento del brazo robótico se tomaron en cuenta tres tipos de pruebas que son:

Prueba 1: Comportamiento del brazo robótico tanto en el modo manual como automático al transportar cubos de madera de diferente masa.

Prueba 2: Comportamiento del brazo robótico tanto en el modo manual como automático al transportar cubos de madera de masa comprendida entre 0 a 175g.

Prueba 3: Tiempo de transporte de cubos de madera de masa entre 75 y 175 gramos de dimensiones (3x3x3) cm, (4x4x4) cm, (4.5x4.5x4.5) cm, (5x5x5) en el modo automático.

#### **3.5.1 Prueba 1**

La prueba consiste en transportar cubos de madera de diferente masa para saber el comportamiento del brazo robótico tanto en el modo manual como en el modo automático y poder determinar cuál es el peso máximo que puede transportar el brazo de un lugar a otro, cuyos resultados se muestran en la siguiente tabla:

<b>Número de cubos Transportados</b>	<b>Masa del cubo (g)</b>	<b>Comportamiento Modo Manual</b>	<b>Comportamiento Modo Automático</b>
0 - 5	10	Correcto	Correcto
6 - 10	25	Correcto	Correcto
11 - 15	50	Correcto	Correcto
16 - 20	75	Correcto	Correcto
21 - 25	100	Correcto	Correcto
26 - 30	125	Correcto	Correcto
31 - 35	150	Correcto	Correcto
36 - 40	175	Correcto	Correcto
41 - 45	200	Incorrecto	Incorrecto
46 - 50	250	Incorrecto	Incorrecto

Tabla 3.2. Cubos de madera de distinta masa transportados por el brazo robótico

Fuente: Autor

### 3.5.2 Prueba 2

La prueba consiste en transportar cubos de madera de diferentes dimensiones y de masa 175 gramos para saber el comportamiento del brazo robótico tanto en el modo manual como en el modo automático y poder determinar el tamaño del cubo que se adecúa a la pinza para un mejor agarre al ser trasladado, cuyos resultados se muestran en la tabla 3.3:

<b>Número de cubos Transportados</b>	<b>Dimensiones del cubo</b> (Largo x Ancho x Alto) cm	<b>Comportamiento Modo Manual</b>	<b>Comportamiento Modo Automático</b>
0 - 5	3x3x3	Correcto	Correcto
6 - 10	3x3x3	Correcto	Correcto
11 - 15	4x4x4	Correcto	Correcto
16 - 20	4x4x4	Correcto	Correcto
21 - 25	4.5x4.5x4.5	Correcto	Correcto
26 - 30	4.5x4.5x4.5	Correcto	Correcto
31 - 35	5x5x5	Correcto	Correcto
36 - 40	5x5x5	Correcto	Correcto
41 - 45	6x6x6	Incorrecto	Incorrecto
46 - 50	6x6x6	Incorrecto	Incorrecto

Tabla 3.3. Cubos de madera de distinto tamaño transportados por el brazo robótico  
Fuente: Autor

### 3.5.3 Prueba 3

La prueba consiste en transportar cubos de madera de masa entre 75 y 175 gramos de dimensiones (3x3x3) cm, (4x4x4) cm, (4.5x4.5x4.5) cm, (5x5x5) cm indistintamente para saber el tiempo que se demora el brazo robótico en realizar el proceso en el modo automático cuyos resultados se muestran en la tabla 3.4:

<b>Cubos transportados</b>	<b>Tiempo promedio de Transporte Modo Automático (segundos)</b>
10	5.92
20	5.46
30	5.88
40	6.30
50	6.03

Tabla 3.4. Tiempo de transporte de los cubos por el brazo robótico  
Fuente: Autor

### 3.6 Análisis de resultados

#### 3.6.1 Análisis de resultados de la prueba 1

Mediante la figura 3.16 se determina el peso máximo que puede tener el cubo a ser transportado por el brazo robótico; el color azul indica el número de cubos que se transportaron, el color vino indica la masa del los cubos especificado en gramos, el color verde indica el comportamiento del brazo robótico en el modo automático al momento de transportar los cubos con un calificativo de 100 en la escala gráfica y el color morado para el modo manual con un calificativo de 100 en la escala gráfica.

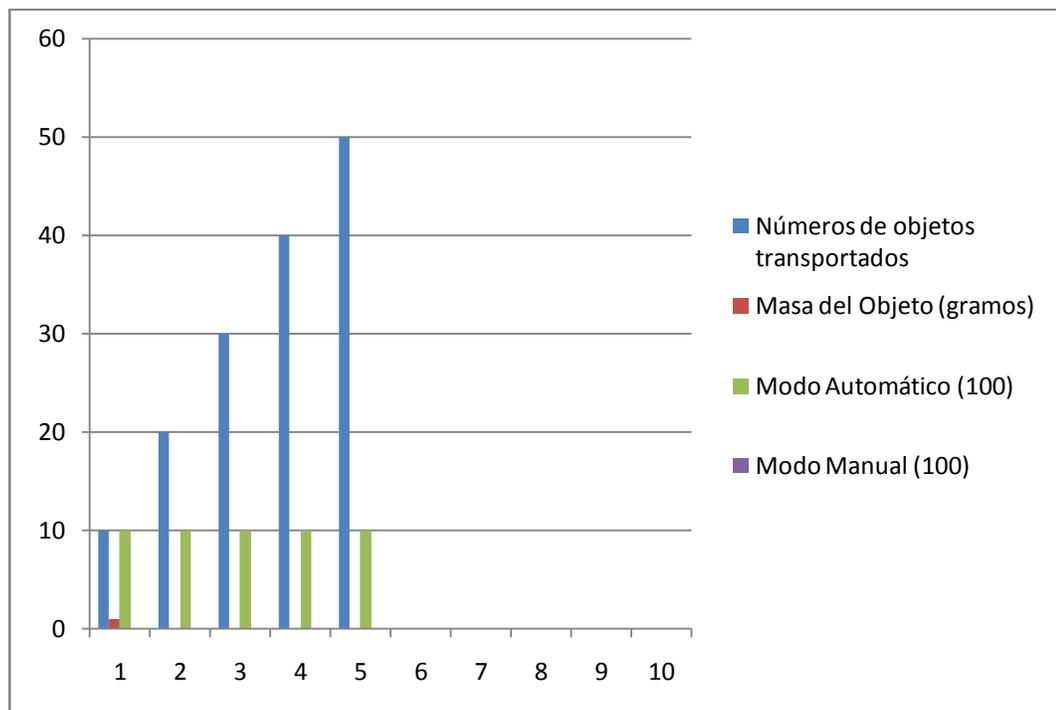


Figura 3.16. Gráfica estadística para determinar el peso máximo a ser transportado por el brazo robótico

Fuente: Autor

Se puede concluir que el peso máximo del cubo a ser transportado es de 175 g aproximadamente, para que el brazo robótico pueda trabajar normalmente y su rendimiento no se vea afectado por exceder dicho peso.

### 3.6.2 Análisis de resultados de la prueba 2

Mediante la figura 3.17 se determina el tamaño del cubo a ser transportado por el brazo robótico; el color azul indica el número de cubos que se transportaron, el color verde indica el comportamiento del brazo robótico en el modo automático al momento de transportar los cubos con un calificativo de 10 en la escala gráfica y el color morado para el modo manual con un calificativo de 10 en la escala gráfica.

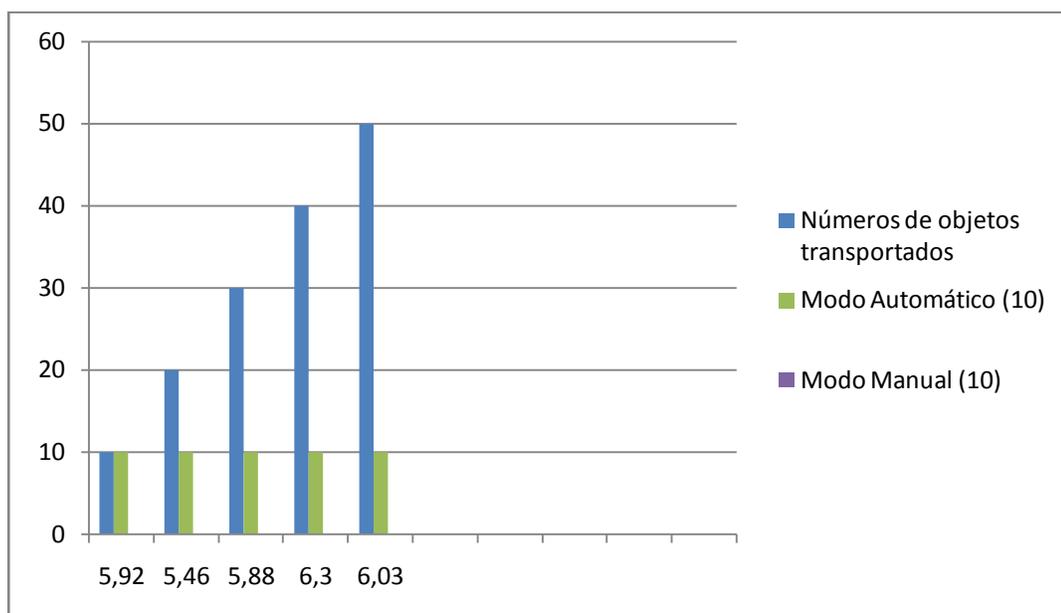


Figura 3.17. Gráfica estadística para determinar el tamaño de los cubos a ser transportados por el brazo robótico

Fuente: Autor

Se puede concluir que el brazo robótico trabaja normalmente cuando el cubo tiene dimensiones no mayores a 5 cm y su peso no sobrepase los 175 g.

### 3.6.3 Análisis de resultados de la prueba 3

Mediante la figura 3.18 se determina el tiempo promedio que se demora el brazo automático en transportar el cubo; el color azul indica el número de cubos que se transportaron, el color verde indica el comportamiento del brazo robótico en el modo automático, y el color vino muestra los tiempos promedios de duración del transporte de los cubos.

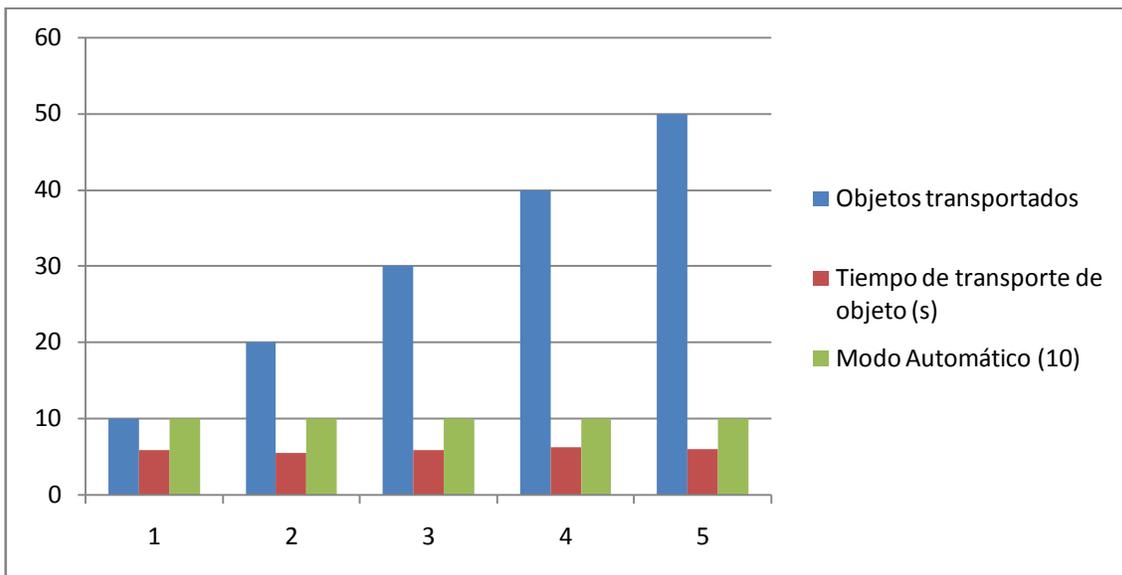


Figura 3.18. Gráfica estadística para determinar el tiempo de transporte de los cubos en el modo automático

Fuente: Autor

Se puede concluir que el brazo robótico trabaja normalmente en el proceso automático en transportando el cubo de dimensiones no mayores a 5 cm y su peso no sobrepase los 175 g en un tiempo promedio de de 5.918 segundos.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- Se implementó un brazo robótico automático con tecnología Arduino para el transporte de cubos de madera de dimensiones hasta máximo de 5 cm y con masa comprendida entre 0 y 175 g entre dos puntos fijos, mediante una interfaz gráfica elaborada en Labview para el control del mismo, y de esta manera fomentar nuevos conocimientos en los campos de la electrónica y robótica.
- Se debe utilizar una fuente de poder DC externa exclusiva para la alimentación de los servomotores, para que no existan conflictos con los voltajes y corrientes que transmite el computador al conectarse con la placa Arduino.
- Siempre se debe conectar primero la fuente externa al Arduino antes que el cable USB que se conecta a la computadora, debido a que la corriente de ésta puede ser muy variable y afectar el funcionamiento de los servomotores del manipulador.
- Es esencial la descarga de los software VI Package Manager e Instrumento de Arquitectura de Software Virtual (VISA) ya que es un estándar para la configuración, programación y resolución de problemas de sistemas de instrumentación tales como dispositivos electrónicos externos (Arduino) o sensores (servomotores) en el programa Labview.

## **Recomendaciones**

- Se podría cambiar el elemento terminal por una doble pinza para poder transportar objetos de mayor tamaño, mayor peso y diferente forma.
- Se podría mejorar el proceso automático determinando diferentes o nuevas posiciones en cada uno de los servomotores.
- Mantener los servomotores dentro del rango de posiciones respectivas para evitar el mal funcionamiento de los mismos.

## BIBLIOGRAFÍA

Abadía, J. A. (1997). *Energía y Computación*.

ANGULO USATEGUI, J. 2. (2000). Robótica práctica Tecnología y Aplicaciones. En J. ANGULO USATEGUI. Madrid: Paraninfo.

BARRIENTOS A., PEÑÍN L., BALAGUER C., RAFAEL ARACIL. (1997). Fundamentos de Robótica. En P. L. BARRIENTOS A., *Fundamentos de Robótica: Mc Graw Hill*.

Botscience. (2013). *Botscience*. Recuperado el Diciembre de 2014, de [http://botscience.net/store/index.php?route=product/product&product\\_id=83](http://botscience.net/store/index.php?route=product/product&product_id=83)

Bueno, A. (16 de Noviembre de 2009). Recuperado el 10 de Enero de 2015, de [http://www.portaleso.com/usuarios/Toni/web\\_robot\\_3/robot\\_indice.html#indice](http://www.portaleso.com/usuarios/Toni/web_robot_3/robot_indice.html#indice)

Canto, C. (25 de Febrero de 2012). Recuperado el 2014 de diciembre de 14, de [http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES\\_8051\\_PDF/2\\_INTROD.PDF](http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF)

labelec. (Marzo de 2009). *labelec*. Recuperado el 2015, de <http://www.info-ab.uclm.es/labelec/solar/electronica/elementos/servomotor.htm>

Mosquera Garcés, L. (2015). *Trabajo de Titulación*.

ni. (13 de Agosto de 2012). *National Instruments*. Recuperado el 20 de Diciembre de 2014, de <http://www.ni.com/white-paper/7564/en/>

OCEANO. (1995). *Diccionario Enciclopédico Oceano*. Madrid.

OLLERO B., A. (2001). *ROBÓTICA, Manipuladores y Robots Móviles*. Barcelona: Marcombo Boixareu Editores.

Rojas, D. (17 de Marzo de 2012). Obtenido de <http://es.scribd.com/doc/85749234/Interfaz-Hombre-Maquina-HMI#scribd>

Sánchez A. (2010). Tutorial Labview. En Sánchez A, *Tutorial Labview* (pág. 10). ITST.

Servodatabase. (2009). *Servodatabase.com*. Recuperado el Diciembre de 2014, de <http://www.servodatabase.com/servo/hitec/hs-311>

SPONG, M. (1989). Robot dynamics and control. En M. SPONG, *Robot dynamics and control*. USA: Ed. John Wiley.

Torrente, Ó. (2010). Arduino. En Ó. Torrente, *Arduino: Curso Práctico de Formación*. AlfaOmega.

*TutorialLabview.pdf*. (s.f.). Obtenido de *TutorialLabview.pdf*: <http://www.esi2.us.es/~asun/LCPC06/TutorialLabview.pdf>

webelectro. (16 de Enero de 2015). *webelectro*. Recuperado el 02 de Febrero de 2015, de <http://www.webelectro.cl/producto/arduino-sensor-shield-v5-0-expansor>

## **ANEXOS**

**ANEXO 1**

**ACTA DE ENTREGA RECEPCIÓN DEL BRAZO ROBÓTICO**

## ACTA DE ENTREGA RECEPCIÓN

En las instalaciones de la Universidad Israel, en la ciudad de Quito, a los 30 días del mes de Marzo del dos mil quince, se procede a la subscripción de la presente acta de entrega – recepción de equipos. Yo, Carlos Vásconez con cédula de ciudadanía 1725752164 hago entrega de los equipos de acuerdo al siguiente detalle:

- 1 brazo robótico elaborado en material acrílico compuesto por 7 servomotores
- 1 Placa Arduino Uno
- 1 Placa Arduino Sensor Shield V.5
- 1 Fuente de alimentación ATX de 600W
- 1 Cable USB
- 4 Cubos de madera de dimensiones pequeñas
- Manual Técnico y Manual de Usuario del brazo robótico
- Cd de instalación del software del brazo robótico.

Para constancia de lo actuado en fe de conformidad y aceptación, firman la presente acta las personas que han intervenido en esta diligencia.

### Entrega

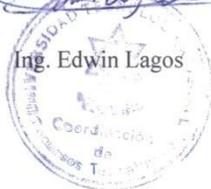


Carlos Vásconez  
C.C. 1725752164

### Recibe



Ing. Wilmer Albarracín

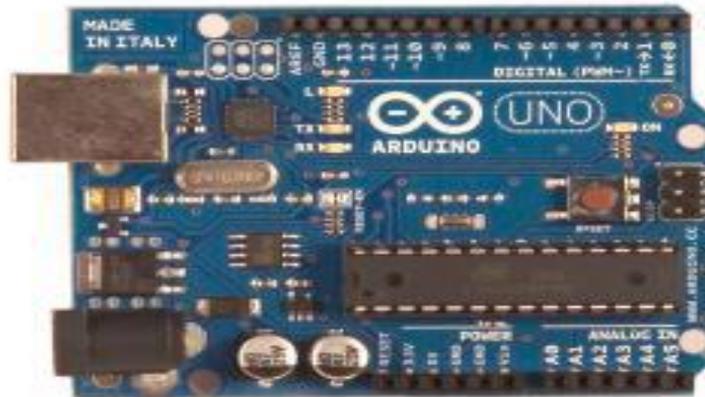
Ing. Edwin Lagos

  
Ing. María Luisa Carrera

## **ANEXO 2**

### **DATASHEET ARDUINO UNO Y ARDUINO SHIELD**

# Arduino UNO



## Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

## Index

Technical Specifications

Page 2

How to use Arduino  
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies  
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



# Technical Specification

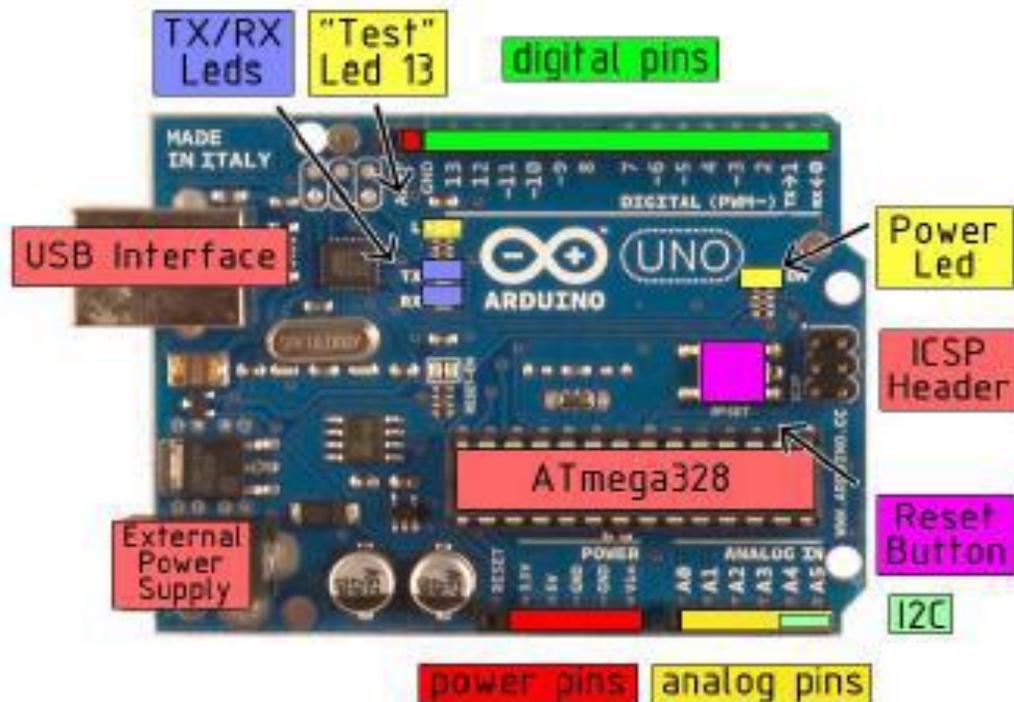


EAGLE files: [arduino@cam.ac.uk](mailto:arduino@cam.ac.uk) Schematic: [arduino-uno-uhemifile.pdf](http://arduino.cc/en/Hardware/arduino-uno)

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

## the board



radiospares

RADIONICS



## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



radiospares

RADIONICS



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an ".inf" file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



radiospares

RADIONICS



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 6 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



*radiospares*

**RADIONICS**



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

**Linux Install**

**Windows Install**

**Mac Install**

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
1 int ledPin = 13; // LED connected to digital pin 13
2
3 // The setup() method runs once, when the sketch starts
4
5 void setup() {
6   // initialize the digital pin as an output:
7   pinMode(ledPin, OUTPUT);
8 }
9
10 // the loop() method runs over and over again,
11 // as long as the Arduino has power.
12
13 void loop() {
14   digitalWrite(ledPin, HIGH); // set the LED on
15   delay(1000);                // wait for a second
16   digitalWrite(ledPin, LOW);  // set the LED off
17   delay(1000);                // wait for a second
18 }
19
```

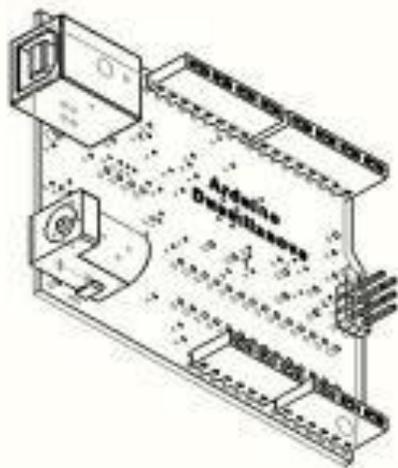
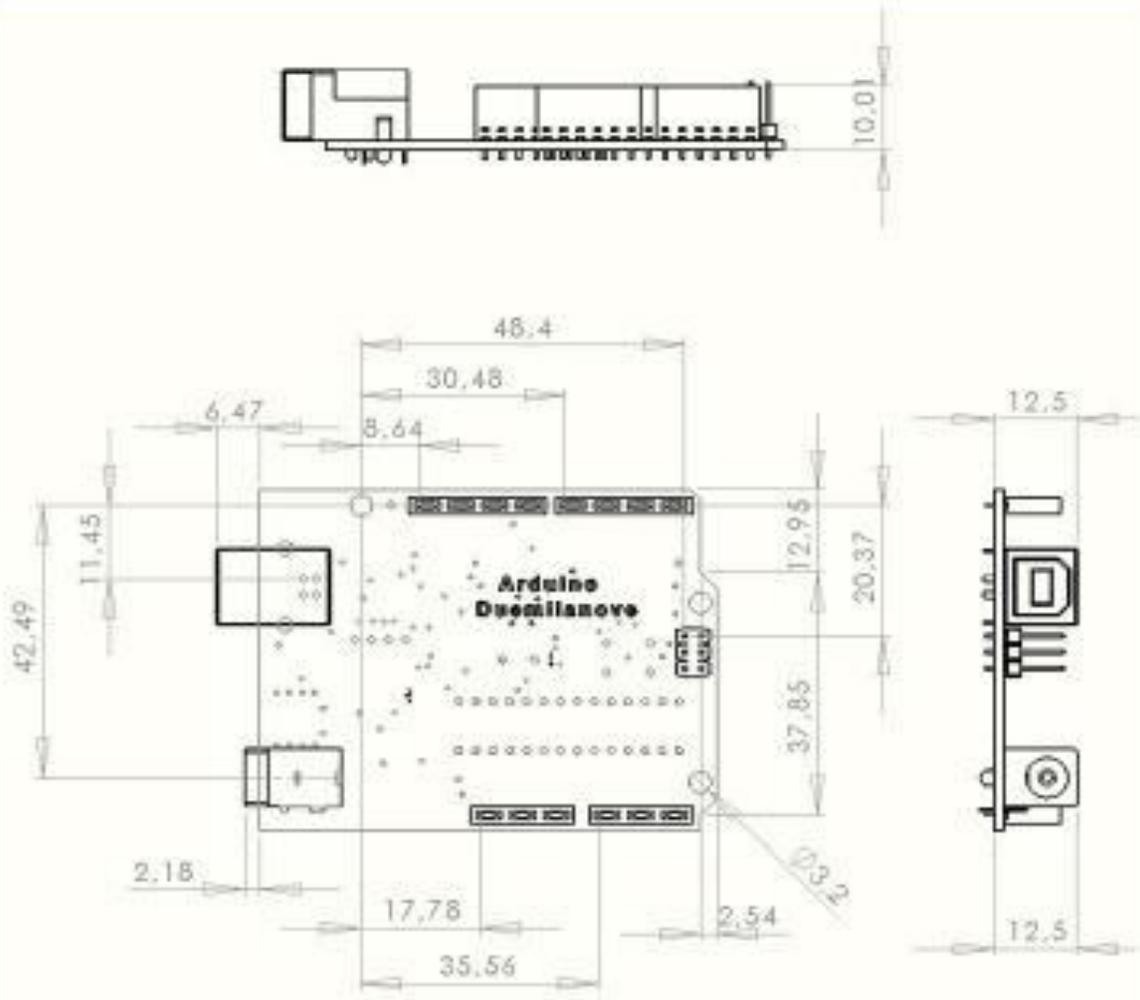


*radiospares*

**RADIONICS**



Dimensioned Drawing



*radiospares* **RADIONICS**



# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



## Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forests.



*radiospares*

**RADIONICS**



**ANEXO 3**

**DATASHEET DEL ARDUINO SHIELD**

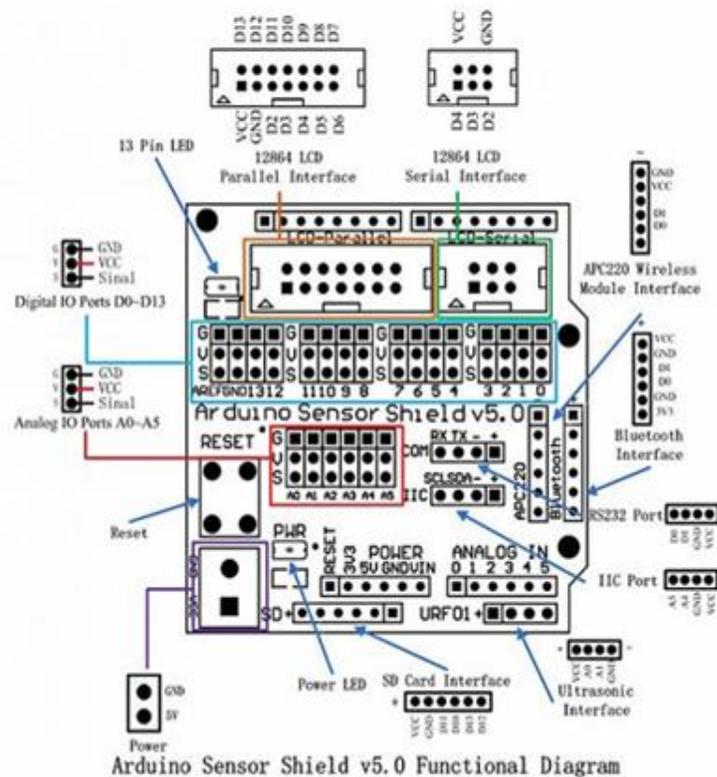
# Arduino I/O Expansion Shield

(SKU: DFR0014)

## Introduction

The Arduino I/O Expansion Shield provides an easy way to connect sensors, servos and RS485 device to Arduino board. It expands Arduino's Digital I/O and Analog Input Pins with Power and GND. It also provides separate PWM Pins which are compatible with standard servo connector. Another unique feature is that the I/O shield has a built-in RS485 converter which allows Arduino communicating with RS485 devices. The communication socket provides an extremely easy way to plug a wireless module such as APC220 RF module and DF-Bluetooth module. It has an individual power input for Servos. A servo power jumper allows user to select using external power or internal power to drive the Servos.

## Diagram



**ANEXO 4**

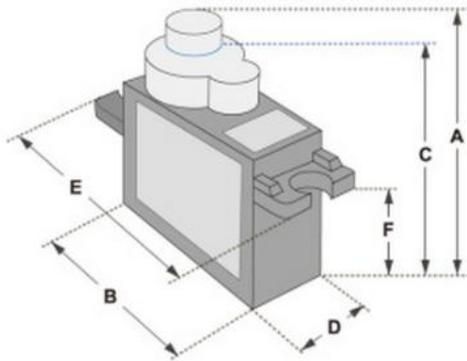
**DATASHEET SERVOMOTOR SG90**

# SERVO MOTOR SG90

# DATA SHEET

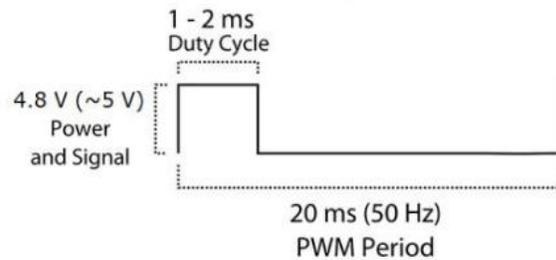
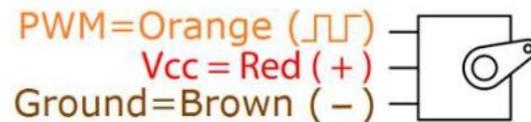


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



**ANEXO 5**

**DATASHEET SERVOMOTOR HITEC HS-311**

# HS - 311 Standard

## Detailed Specifications

**Control System:** +Pulse Width Control 1500usec Neutral

**Required Pulse:** 3-5 Volt Peak to Peak Square Wave

**Operating Voltage:** 4.8-6.0 Volts

**Operating Temperature Range:** -20 to +60 Degree C

**Operating Speed (4.8V):** 0.19sec/60° at no load

**Operating Speed (6.0V):** 0.15sec/60° at no load

**Stall Torque (4.8V):** 42 oz/in (3.0 kg/cm)

**Stall Torque (6.0V):** 49 oz/in (4.5 kg/cm)

**Current Drain (4.8V):** 7.4mA/idle, 160mA no load operating

**Current Drain (6.0V):** 7.7mA/idle, 180mA no load operating

**Dead Band Width:** 5usec

**Operating Angle:** 40° one side pulse traveling 400usec

**Direction:** Clockwise/Pulse Traveling 1500 to 1900usec

**Motor Type:** Cored Metal Brush

**Potentiometer Drive:** 4 Slider/Direct Drive

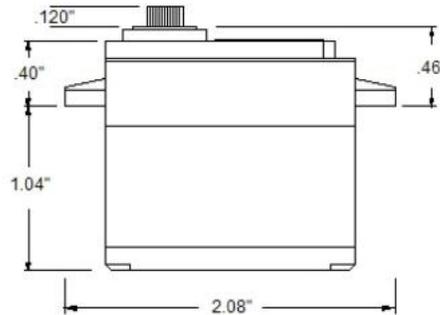
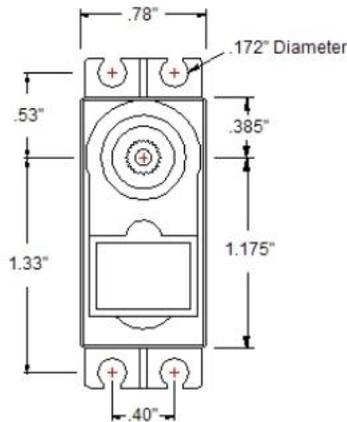
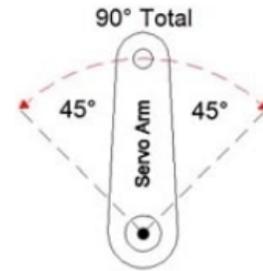
**Bearing Type:** Top/Resin Bushing

**Gear Type:** Nylon

**360 Modifiable:** [Yes](#)

**Connector Wire Length:** 11.81" (300mm)

**Weight:** 1.52oz (43g)



**ANEXO 6**

**MANUAL TÉCNICO DEL BRAZO ROBÓTICO**

## Descripción de bases de datos

### Especificaciones Técnicas

<b>Brazo robótico</b>		
Alimentación	V – Hz	110 – 50
Capacidad	W	200
Potencia Absorbida	W	40
Corriente Consumida	A	1
<b>Servomotor MS-310</b>		
Largo	mm	39.9
Ancho	mm	19.8
Alto	mm	36.3
Peso	g	43
Velocidad de giro 4.8 V	seg/60°	0.19
Velocidad de giro 6.0 V	seg/60°	0.15
Engranaje	Plástico	
<b>Servomotor SG90</b>		
Largo	mm	21.8
Ancho	mm	22.6
Alto	mm	11.4
Peso	g	9
Velocidad de giro	seg/60°	0.12
Voltaje de operación	V	5

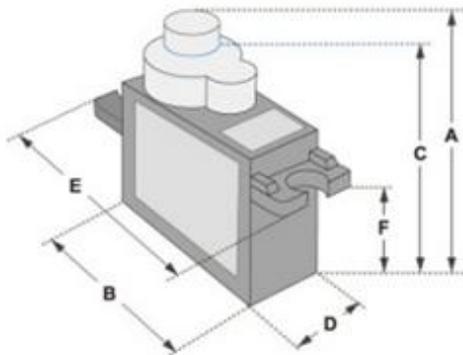
<b>Arduino UNO</b>		
Microcontrolador	ATmega238	
Voltaje de Operación	V	5
Voltaje de Entrada (Recomendada)	V	5 – 12
Voltaje de Entrada (Límites)	V	6 - 20
Pines E/S Digitales	14 (6 pines PWM)	
Pines de Entrada Analógica	6	
Corriente por Pin E/S	mA	40
Memoria Flash	32 KB (ATmega238)	
SRAM	KB	2
EEPROM	KB	1
Clock Speed	MHz	16
<b>Arduino Sensor Shield V.5</b>		
Controlador	Secuencia de interfaz analógica y digital Interfaz i2C	
Interfaz	Controladora de 32 servos. Comunicación del módulo Bluetooth. APC220 WIRELESS. rs232. Comunicación módulo de tarjeta SD. Comunicación APC220 inalámbrica módulo de RF. 12864 LCD serie y paralelo. Sensor de ultrasonidos.	

## Listado de Materiales

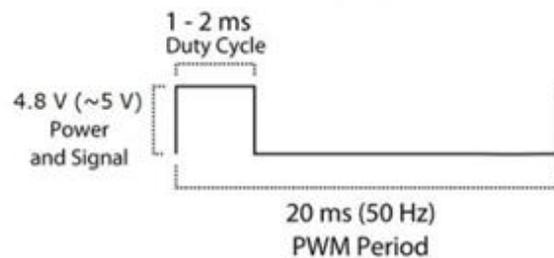
Elemento	Cantidad
Manipulador	1
Fuente de alimentación ATX	1
Cable USB	1
Microcontrolador Arduino UNO	1
Arduino Sensor Shield V.5	1
Servomotor SG90	2
Servomotor MS-311	5

## Diagramas Esquemáticos

### Servomotor SG90

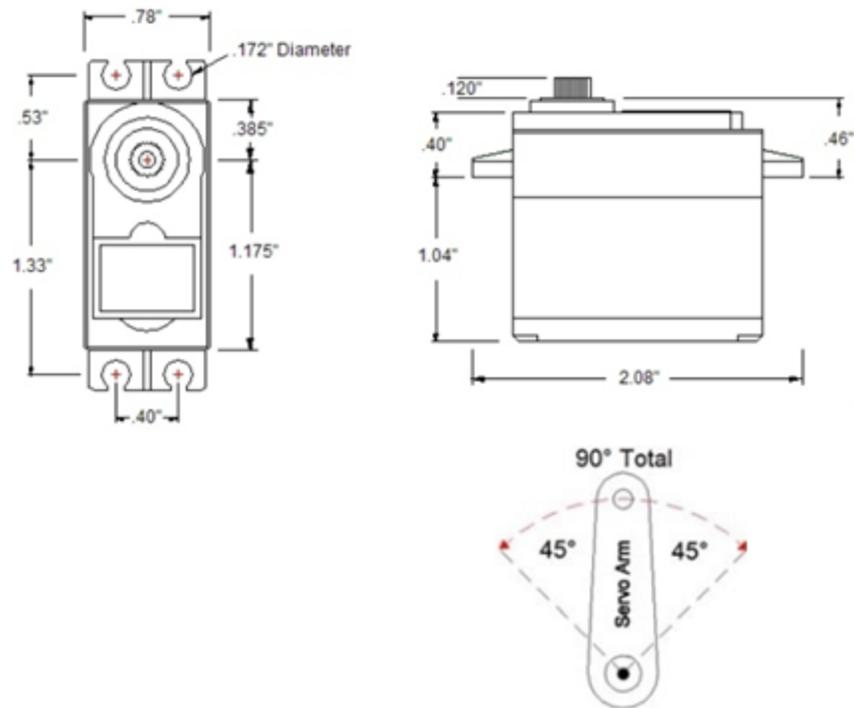


PWM=Orange (⏏)  
Vcc = Red (+)  
Ground=Brown (-)

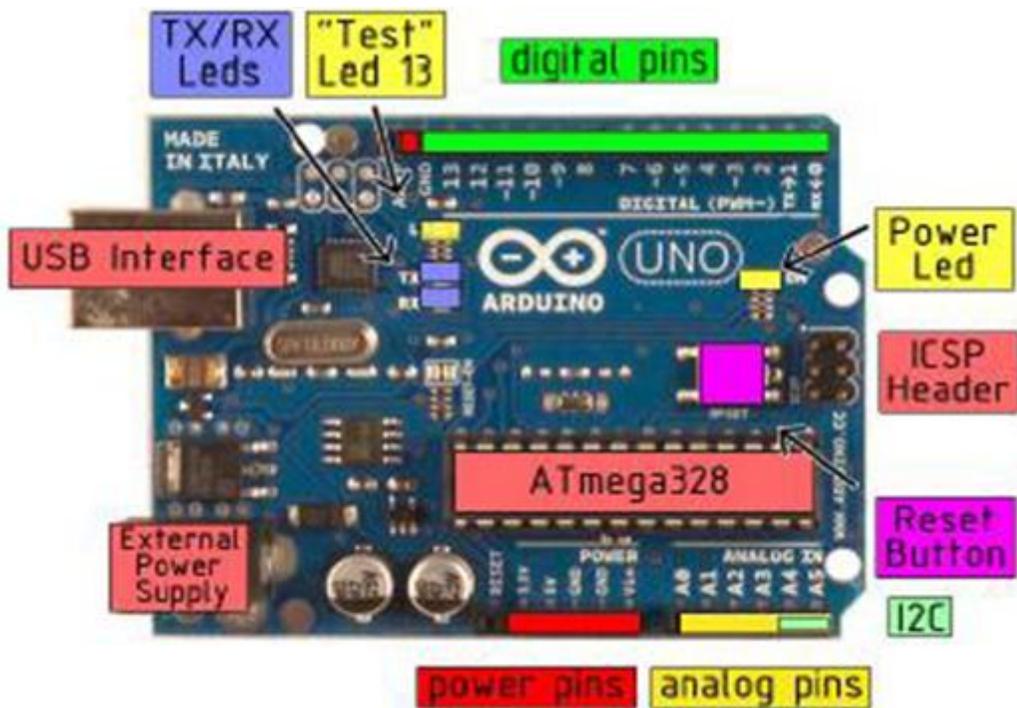


Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

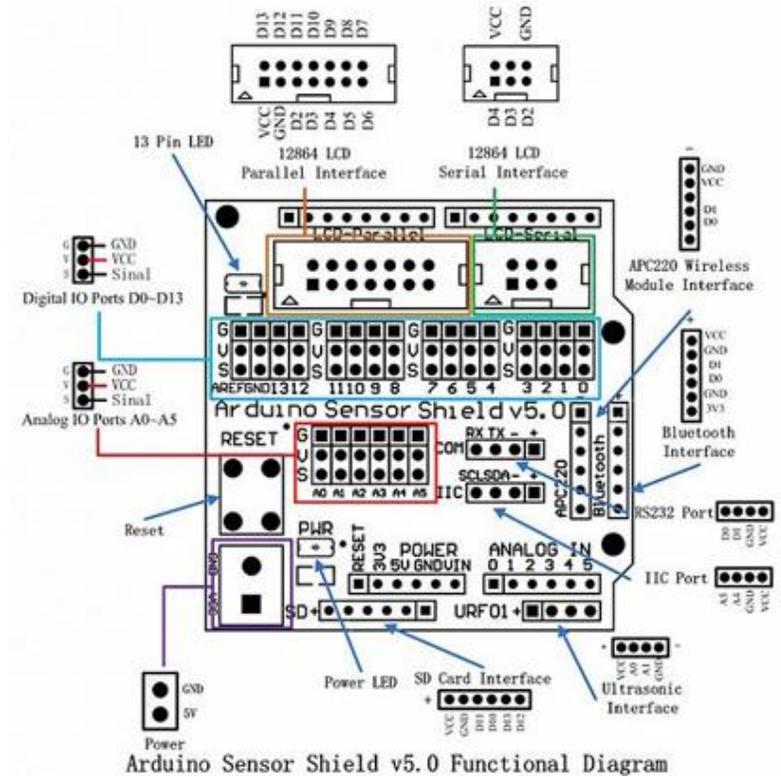
## Servomotor Ms-311



## Microcontrolador Arduino UNO



## Arduino Sensor Shield V.5



### Guía Rápida de Mantenimiento

- Revisión visual periódica del sistema
- Revisión de piezas flojas
- No someter al sistema a esfuerzos innecesarios
- Verificar que la fuente de alimentación se encienda.
- No utilizar líquidos inflamables ni corrosivos en la limpieza exterior del sistema

## **Guía Rápida de Solución de Problemas**

- En caso de mal funcionamiento del sistema pulsar el botón RESET de la placa Arduino
- Si el brazo robótico realiza esfuerzo innecesario detenga el sistema, y a continuación revise que las piezas no se encuentren flojas o hayan sufrido algún daño.
- En caso de contacto con el agua desconecte inmediatamente la alimentación del sistema
- En caso de detención del brazo robótico verifique que no exista ningún obstáculo que permita la movilidad del mismo.

**ANEXO 7**

**MANUAL DE USUARIO DEL BRAZO ROBÓTICO**

## I

### PRECAUCIONES DE SEGURIDAD

Para su seguridad y evitar lesiones personales y / o daños, tenga en cuenta las siguientes precauciones de seguridad:

Los tipos de símbolos que se indican a continuación se utilizan para clasificar y describir el tipo de instrucciones que deben tenerse en cuenta. (Los símbolos siguientes son ejemplos).



Este símbolo se utiliza para avisar a los usuarios que no puede realizarse un determinado procedimiento operativo.



Este símbolo se utiliza para avisar a los usuarios que debe realizarse un determinado procedimiento operativo para utilizar la unidad de forma segura.

#### Encendido del dispositivo



El voltaje de la fuente de alimentación externa para el sistema es de 5V. Conecte la fuente a una toma de CA de 110 V. Si se utiliza una fuente con una corriente y voltaje nominal sin especificar, el sistema podría emitir humo o sobrecalentarse.



Las conexiones a tierra están incorporadas dentro del sistema, la polaridad al momento de conectar la fuente a la placa Arduino debe ser correcta.



Para el encendido del sistema se debe conectar la fuente externa a la placa Arduino y después el cable USB al ordenador.

### Uso correcto



No tire del cable de alimentación ni del conector, no los doble, no coloque objetos encima de ellos y procure que no rocen con cables descubiertos. Si el cable de alimentación o el conector están dañados, podría provocar descargas eléctricas.



No tire de los cables de los servomotores, puede desconectar los mismos del Arduino Shield ocasionando descargas eléctricas.



Evite que el brazo robótico levante demasiado peso realizando esfuerzo innecesario.



Coloque los cubos de madera en la zona de transporte para que el brazo pueda recogerlo con su pinza.

### Desconexión del sistema



Oprima el botón Reset de la placa Arduino para apagar el sistema.



Desconecte el cable USB del ordenador antes de apagar la fuente externa.

## II

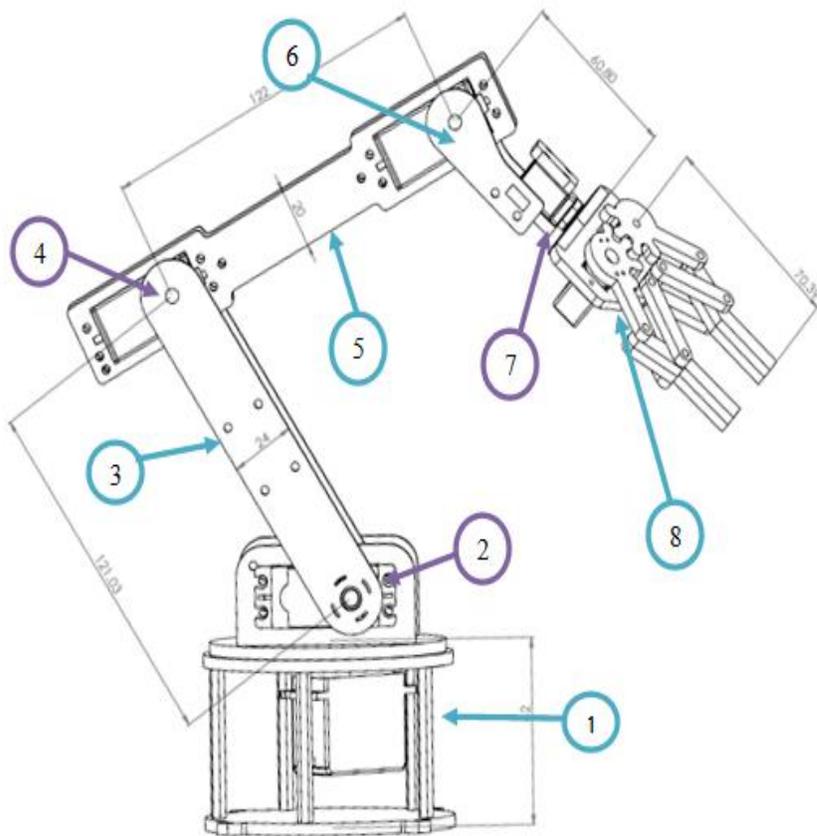
### MANUAL DEL HARDWARE

#### Introducción

##### Descripción de partes

Robot antropomórfico con 6 grados de libertad y un determinado rango de giro para cada articulación, de dimensiones reducidas y un peso que no supera los 3 kilogramos.

El brazo robótico lo conforman 5 partes principales: Base, antebrazo, brazo, muñeca y pinza

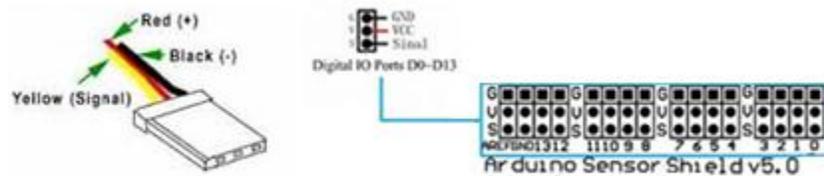


Partes del Brazo Robótico	
1	Base
2	Hombro
3	Antebrazo
4	Codo
5	Brazo
6	Muñeca
7	Mano
8	Pinza

Composición del brazo robótico

## Descripción del control del brazo robótico

El control del brazo robótico se basa en la placa Arduino Uno. La comunicación Ordenador - Arduino se realiza mediante un cable USB para la transmisión de instrucciones hacia los servomotores. Para facilitar la conexión de los servos se utiliza el Arduino Sensor Shield V.5, el cual contiene una trama de terminales de 3 pines (Señal, Voltaje, Tierra)



Terminales Servomotor – Arduino Shield

Son 7 servomotores ubicados en las diferentes partes del manipulador por lo que se asigna un número de referencia a cada uno de ellos para la conexión a las terminales del Arduino Shield para su respectivo control.

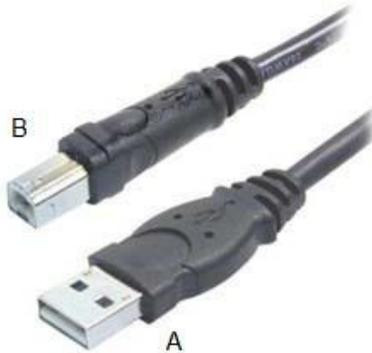
Parte del Manipulador	# Servomotor	Pines Arduino
Base	0	6
Hombro	1   2	7   8
Codo	3	9
Mano	4	10
Muñeca	5	11
Pinza	6	12

Nomenclatura Servomotor – Conexión Arduino Shield

## Accesorios

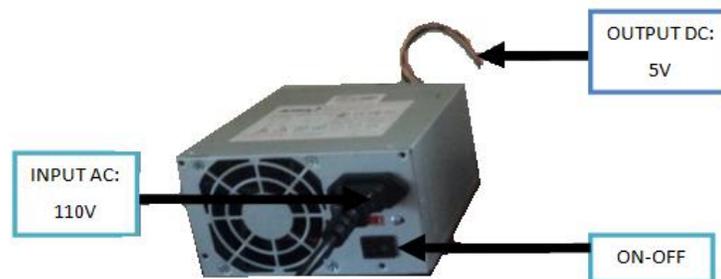
El cable USB que comunica Arduino con el ordenador, y la fuente externa para la alimentación exclusiva de los servomotores son accesorios propios del brazo robótico.

El Cable USB cuenta con un conector Tipo A para el enlace con el ordenador, mientras que el tipo B se conecta al dispositivo Arduino.



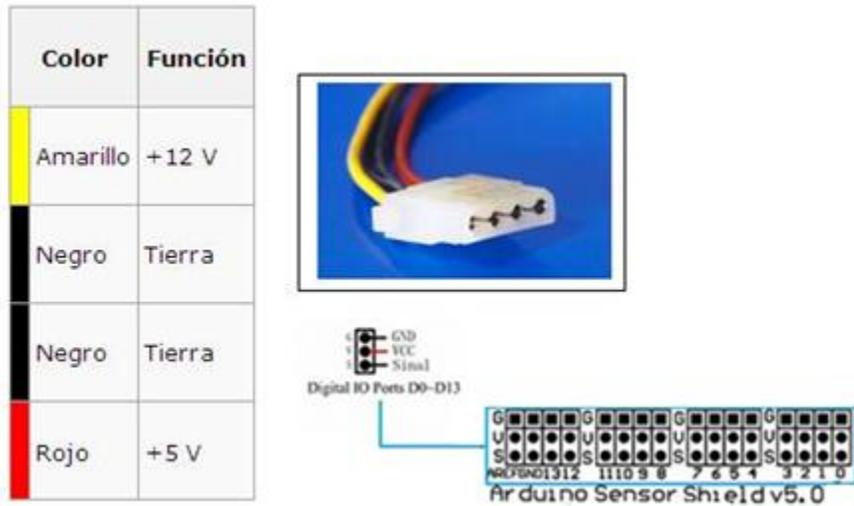
Cable USB para comunicación Arduino – Ordenador

La alimentación para los servomotores se da por una fuente externa ATX de 600W que cuenta con un voltaje de salida de 5V, ideal para el funcionamiento de los servos.



Fuente de alimentación Externa ATX

La salida de voltaje dc de la fuente tiene cuatro cables. Para la alimentación de los servomotores se debe conectar el cable rojo de +5V y el cable negro que es la tierra a la trama de terminales del Arduino Sensor Shield. El cable rojo se conecta al Pin Voltaje (V) y el cable negro al pin Ground (G).



Terminal Salida de Voltaje DC de la fuente – Conexión Arduino Shield

### III

## MANUAL DEL SOFTWARE

### **Introducción**

Se desarrolló una aplicación dentro del programa LabVIEW para que el usuario pueda controlar mediante una interfaz gráfica el brazo robótico de forma manual, y poder activar el modo automático cuando se lo desee. La aplicación se denomina “**ControlBrazoRobotico**” y es necesario instalarla previamente.

Para el control manual que podrá realizar el usuario a través de la interfaz gráfica se utilizan deslizadores para aumentar o disminuir la posición angular de los servomotores, ubicados en cada parte del brazo.

Para activar el modo automático se utiliza un switch deslizador, cuando está en la posición hacia arriba se encuentra en el modo manual, cuando el switch se encuentra en la posición hacia abajo el brazo robótico pasará al modo automático, realizando un movimiento secuencial de transporte para el cubo de madera desde un punto a otro.

### **Requisitos del Sistema para la instalación del software**

Sistema Operativo

- Windows 7 de 32 bits

### **Instalación del Software Control Brazo Robótico**

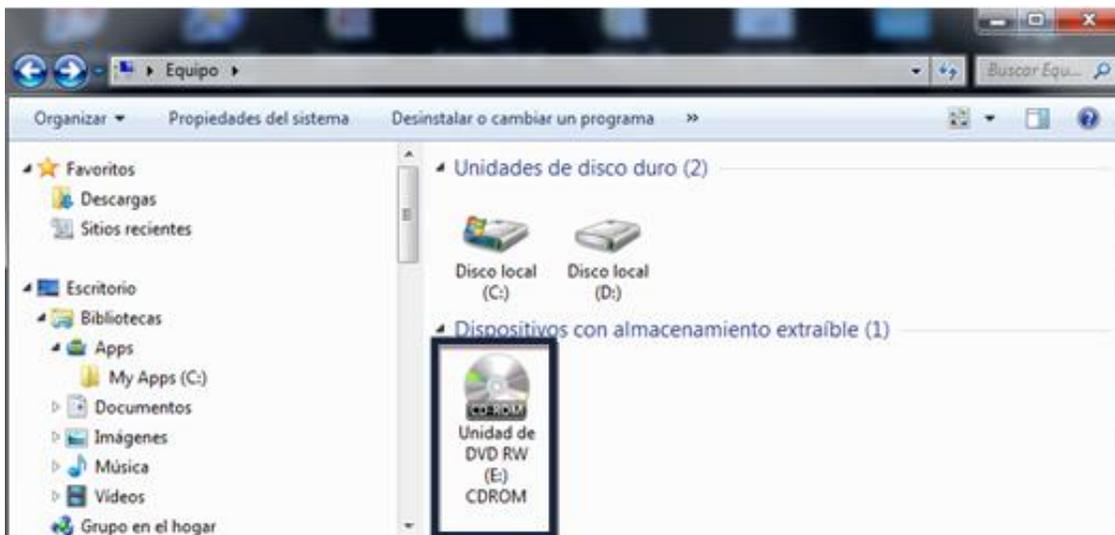
El sistema de instalación es automático pero tiene varios pasos a considerar

**Nota:** Para instalar el software en el Sistema Operativo Windows, se requiere permisos del Administrador de Windows.

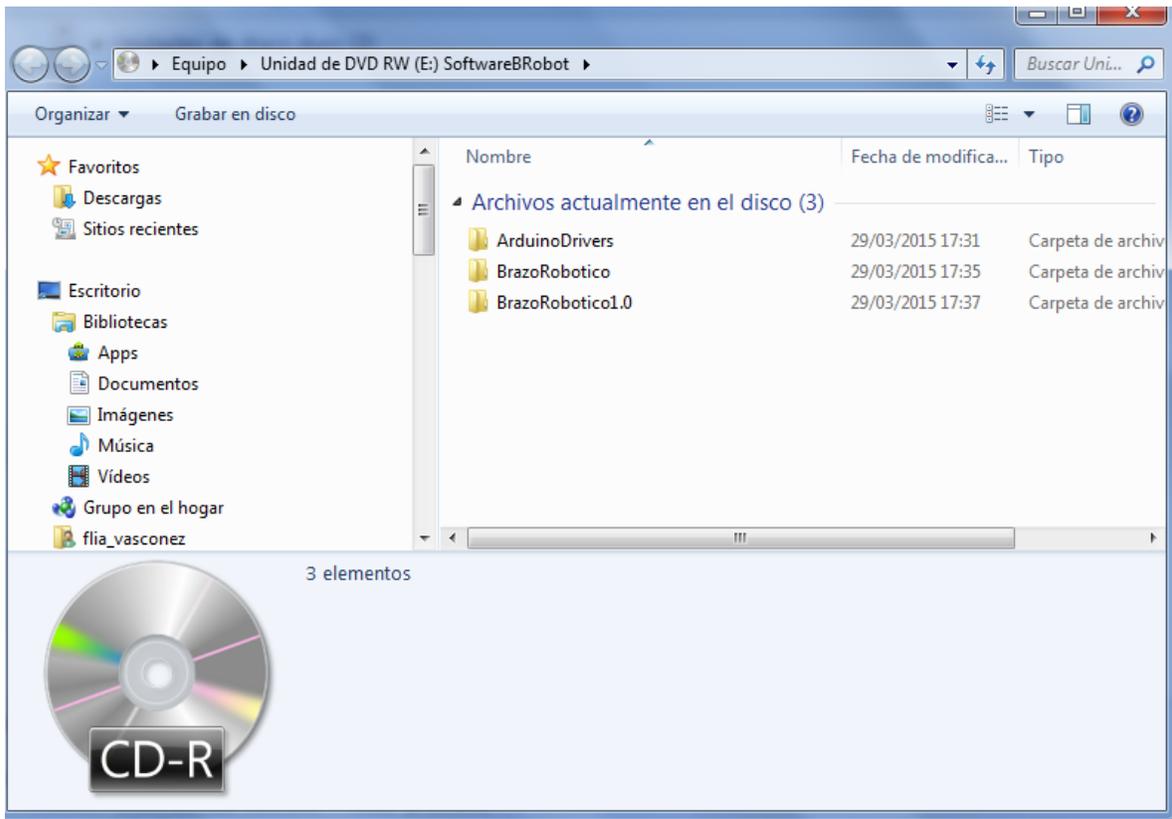
## Pasos de la Instalación

**Paso 1:** Insertar el Cd-R de Control Brazo Robótico en la unidad lectora del computador.

**Paso 2:** Si no existe reproducción automática que permita ver los archivos del Cd haga doble clic en el ícono de Unidad de DVD RW O CDROM (**Control Brazo Robótico**) que se encuentra en la ventana Equipo.

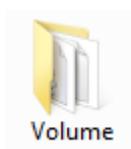


Se abrirá la siguiente ventana:

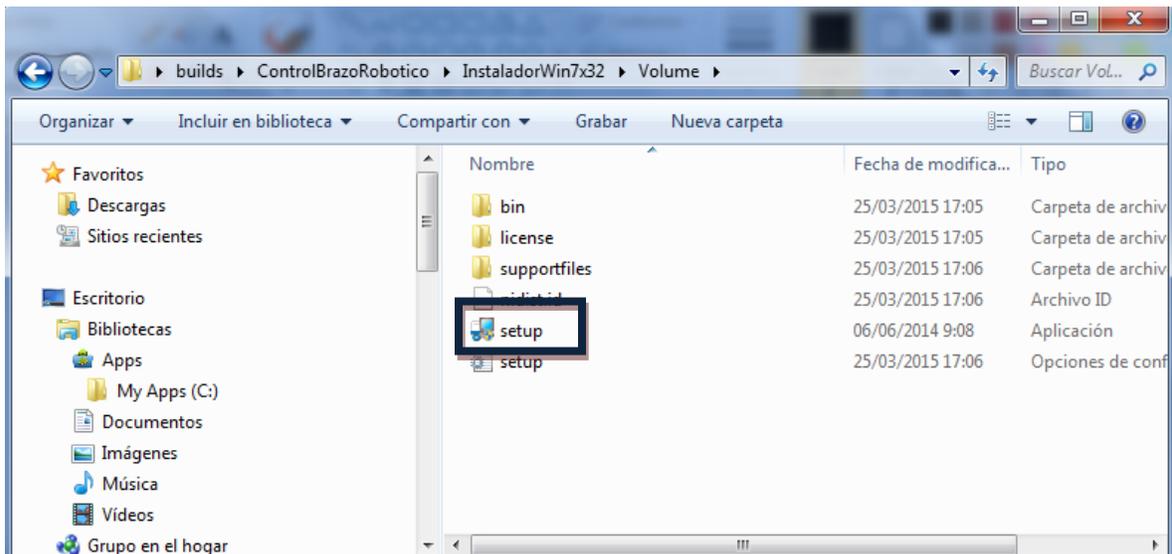


**Paso 3:** Hacer doble clic en la carpeta “BrazoRobotico1.0”.

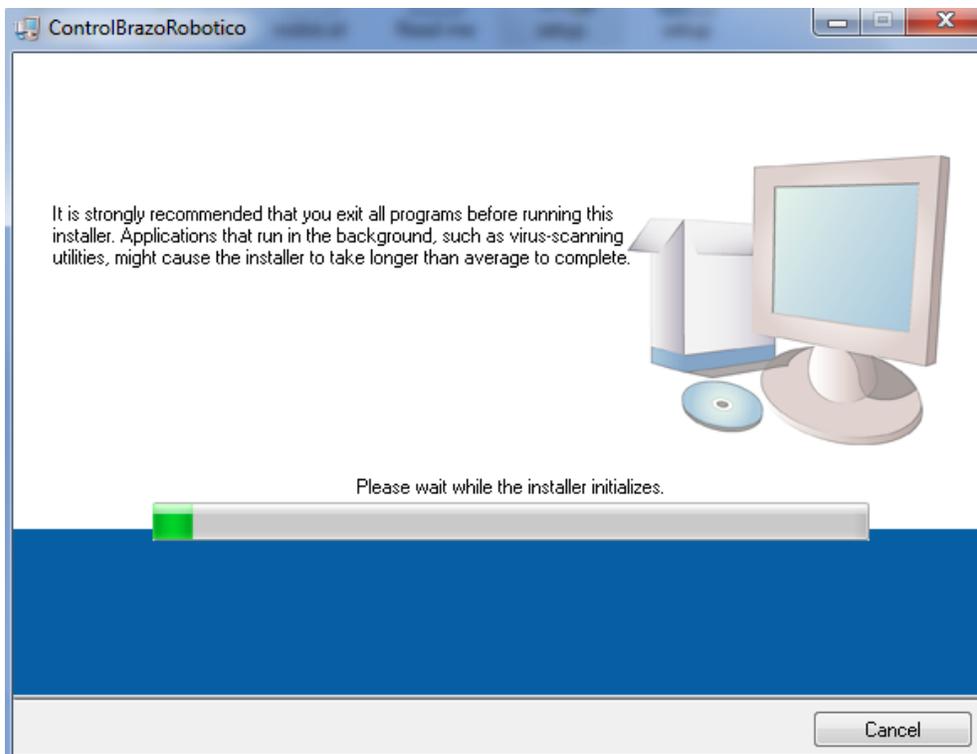
Se abrirá subcarpeta “volumen”, hacer doble clic en la misma. Se abrirá un ventana (Ver paso 4)



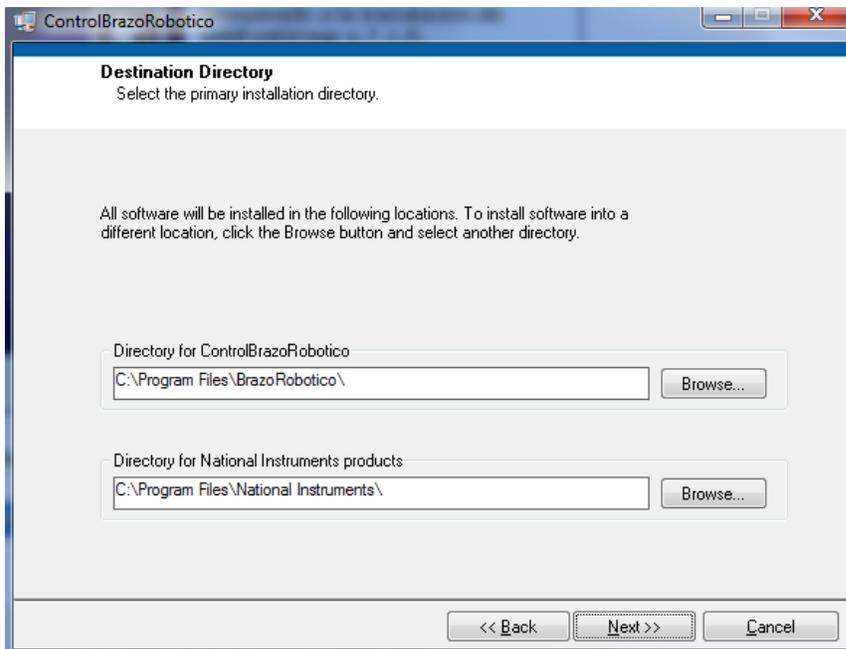
**Paso 4:** Hacer doble clic en el ícono del instalador **Setup**.



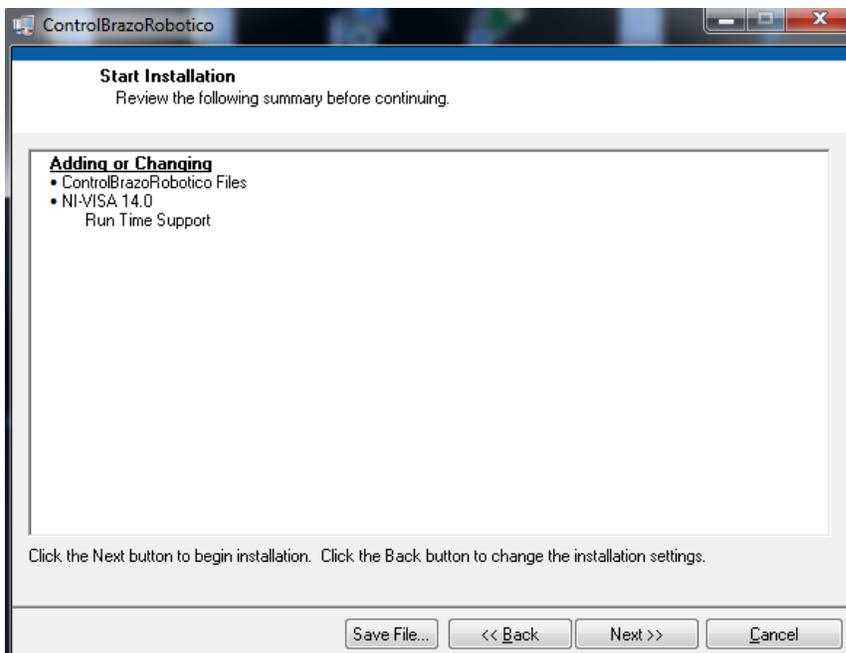
**Paso 5:** Se abrirá la siguiente ventana. Espere mientras se inicia el instalador del software Control Brazo Robótico.



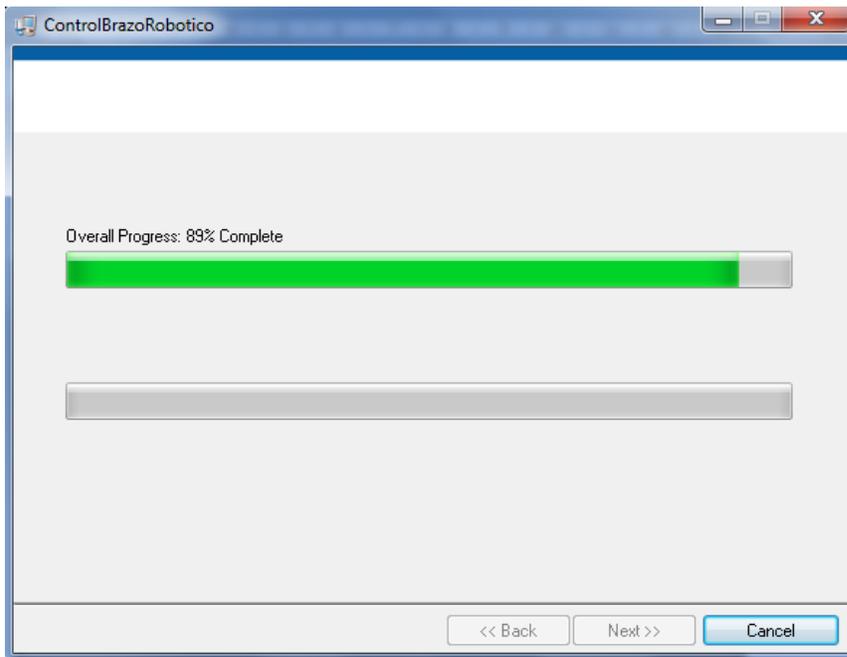
**Paso 6:** Seleccione la carpeta o destino donde se desea instalar la aplicación. Presione Next para continuar.



**Paso 7:** Se abrirá la siguiente ventana, donde se indica los archivos y componentes a instalar en el computador. Presione NEXT para iniciar la instalación

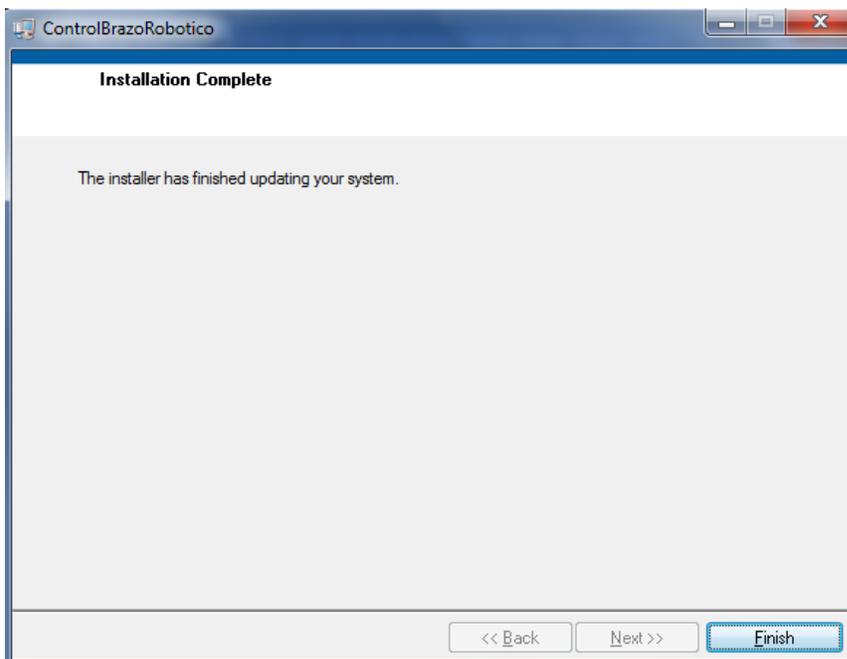


**Paso 8:** Se iniciará el proceso de instalación. Espere hasta que se complete la instalación.

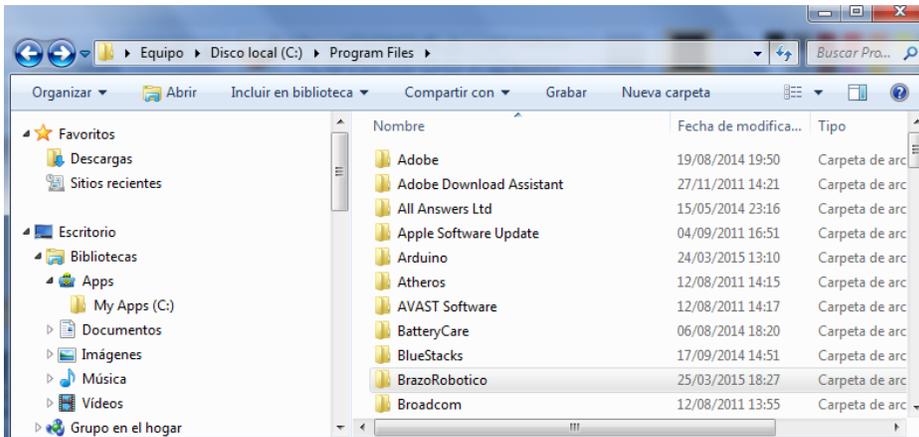


**Paso 9:** Si la instalación ha sido correcta, aparecerá automáticamente la siguiente ventana.

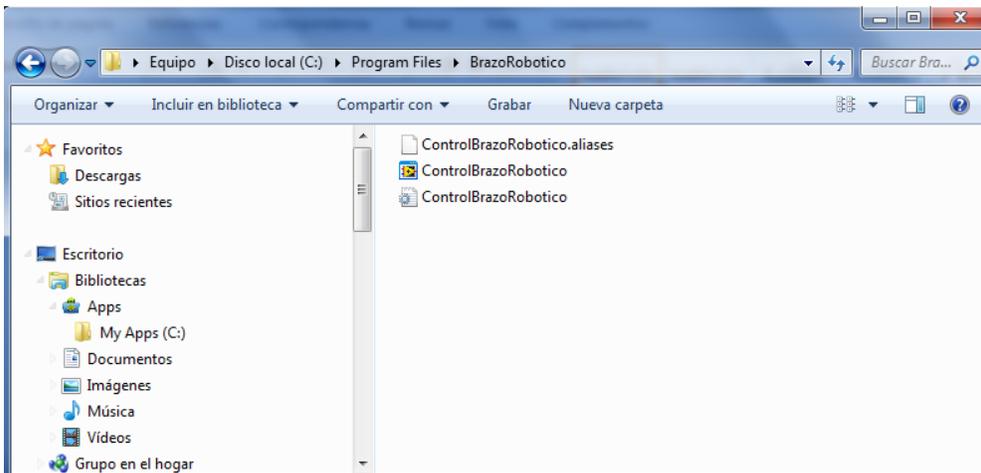
Pulse **FINISH** para finalizar el proceso de instalación.



**Paso 10:** Abrir la carpeta o destino donde se instaló la aplicación. Hacer doble clic en la carpeta “Brazo Robótico”.



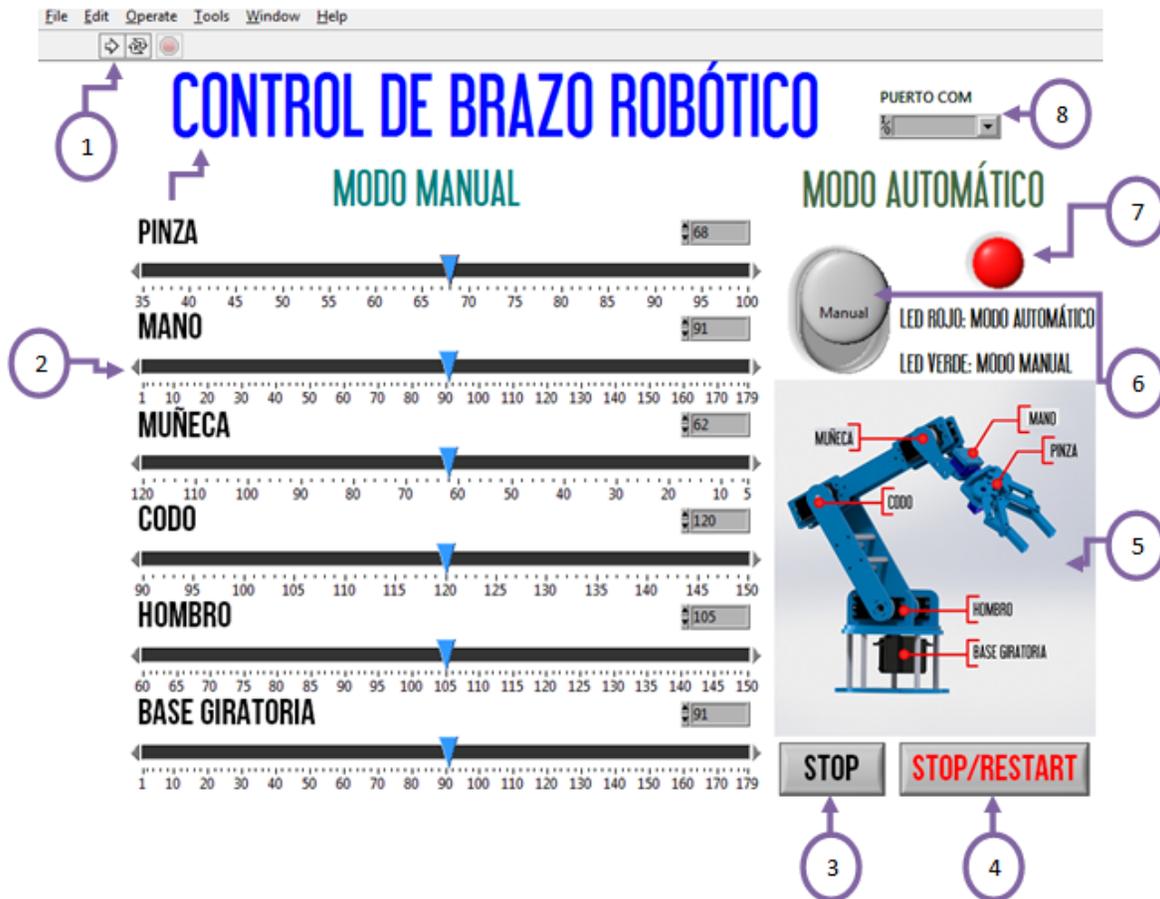
**Paso 11:** Se abrirá la siguiente ventana que contiene la aplicación Control Brazo Robótico. Hacer doble clic para ejecutar el programa.



## Aplicación control brazo robótico

Esta aplicación es una interfaz gráfica para que el usuario pueda controlar el brazo robótico de forma manual así como activar el modo automático de transporte de cubos de madera de un punto fijo a otro.

## Componentes de la Interfaz Gráfica



<b>Componentes utilizados para la interfaz gráfica</b>		<b>Descripción</b>
1	Opción "Run"	Ejecuta la aplicación.
2	Deslizadores	Permite que el usuario controle la posición angular del respectivo servomotor de forma manual.
3	Botón STOP	Detiene la aplicación. Para detener la aplicación es necesario que el brazo se encuentre en modo manual.
4	Botón STOP/RESTART	Detiene el proceso automático cambiando al modo manual luego de haberlo seleccionado. Reinicia el proceso automático si el botón SWITCH se encuentra en automático.
5	Imagen Brazo Robótico	Permite que el usuario conozca cual parte del brazo robótico es la que está controlando en el modo manual.
6	Botón SWITCH	Activa el modo Automático o el modo manual.
7	Led Indicador	Indica en qué modo se encuentra el brazo robótico
8	Puerto COM	El usuario debe seleccionar el puerto al cuál se encuentra conectado el dispositivo Arduino

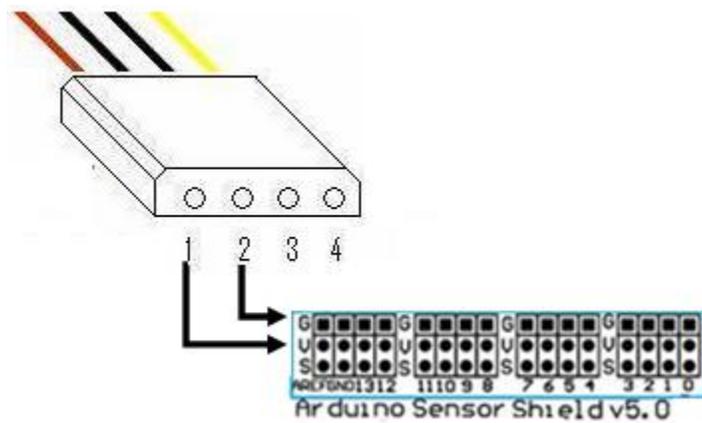
## IV

### INSTRUCCIONES DE USO DEL BRAZO ROBÓTICO

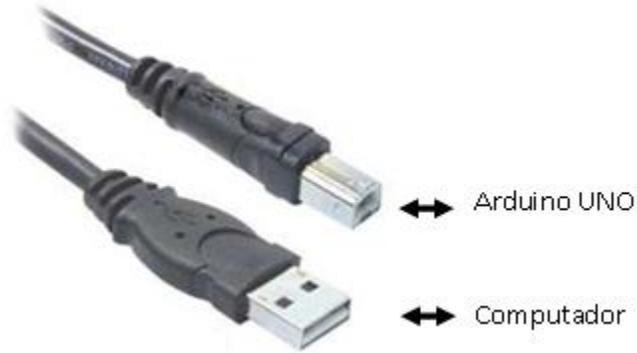
1. Verifique que la fuente de alimentación externa se encuentre apagada.



2. Conecte la fuente de alimentación externa de 5V dc (Cable rojo y negro) al Arduino Shield en los pines V (Voltaje) y G (Ground) de la trama donde se conectan los servomotores.



3. Encienda la fuente de alimentación externa (ON).
4. Conectar el cable USB de la Placa Arduino al Computador



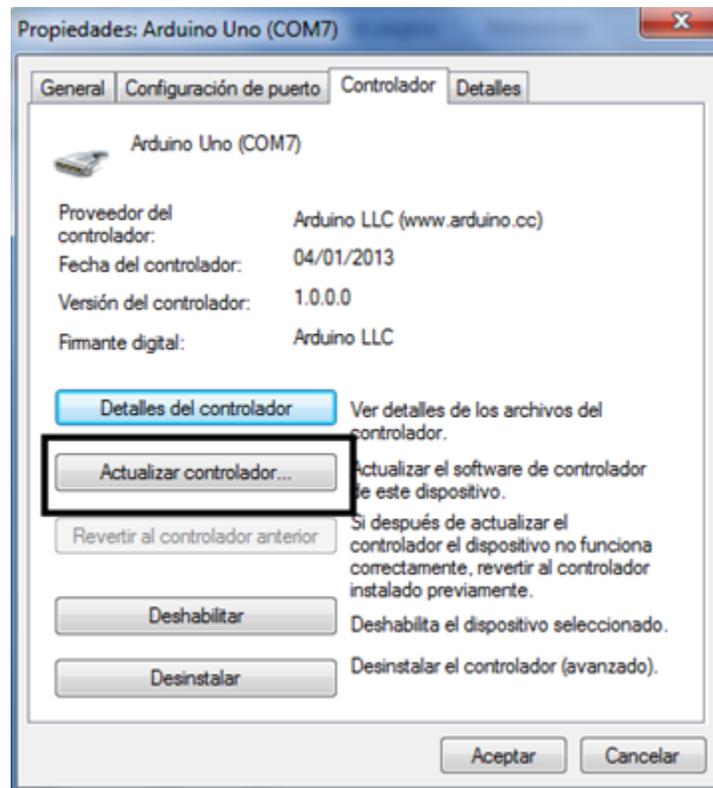
5. Verificar que el computador reconozca el dispositivo Arduino y en qué periférico (puerto COM) se encuentra conectado.

5.1 Para esto, Clic en Inicio, **Dispositivos e impresoras** o **Administrador de dispositivos**. Se abrirá una pantalla en la cual deberá aparecer el dispositivo Arduino especificando el periférico al cual se encuentra conectado.

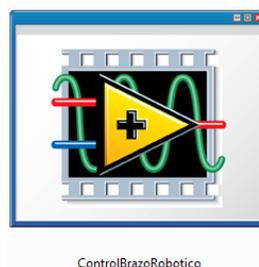


**Nota:** En caso de que el ordenador reconozca el dispositivo como desconocido, hacer clic derecho en el ícono Arduino Uno, elegir la opción propiedades. Seleccionar la pestaña Hardware, clic en la opción cambiar configuración. Seleccionar la Pestaña Controlador y hacer clic en la opción actualizar controlador. A continuación, seguir los pasos que el asistente de Windows indica. Se puede buscar el controlador automáticamente en el equipo o por internet; o buscar el controlador manualmente, en

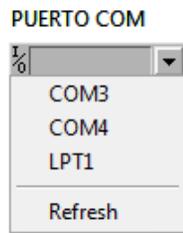
este caso existe una carpeta Arduino Drivers en el Cd de instalación del software proporcionado.



6. Hacer doble clic en la Aplicación “ControlBrazoRobotico” la cual debe estar previamente instalada en el computador. Se abrirá la interfaz gráfica que controla el brazo robótico.



7. Antes de inicializar la aplicación seleccione el puerto COM al cual se conectó el dispositivo Arduino. (Ver paso 5.1)



**Nota:** En caso de no encontrar el Puerto COM entre las opciones, seleccione REFRESH para actualizar las alternativas. Si el puerto COM no aparece aún revise el paso 5.1.

8. Ejecute la aplicación mediante la Opción RUN.



9. El brazo robótico siempre empezará activado en el modo MANUAL. Si se desea activar el modo Automático oprima el botón Switch hacia abajo.



10. Si se desea repetir el proceso automático, pulse el botón STOP/RESTART.



11. Para volver al modo manual oprima el botón Switch hacia arriba y a continuación oprima el botón STOP/RESTART.

12. Para detener la aplicación el brazo robótico debe encontrarse en el modo manual. A continuación oprima el botón STOP.



13. Para apagar completamente el sistema oprima directamente el botón STOP/RESET que se encuentra en la placa Arduino Shield.

14. Cierre la aplicación y desconecte el cable USB del ordenador.

15. Apague la fuente de alimentación externa.