



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y
COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TRABAJO DE TITULACIÓN PARA LA OBTENCIÓN DEL
TÍTULO DE:**

**TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TEMA: “IMPLEMENTACIÓN DE UN CONTROL DE
SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE
ARDUINO-MATLAB”**

AUTOR: ARIAS CASILLAS EDISON LEONARDO

DIRECTOR: ING. CAMPAÑA RIOFRIO EMMA DOLORES

LATACUNGA

2016



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y
COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TRABAJO DE TITULACIÓN PARA LA OBTENCIÓN DEL
TÍTULO DE:**

**TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TEMA: “IMPLEMENTACIÓN DE UN CONTROL DE
SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE
ARDUINO-MATLAB”**

AUTOR: ARIAS CASILLAS EDISON LEONARDO

DIRECTOR: ING. CAMPAÑA RIOFRIO EMMA DOLORES

LATACUNGA

2016



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

CERTIFICACIÓN

Certifico que el presente Trabajo de Graduación, titulado “IMPLEMENTACIÓN DE UN CONTROL DE SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE ARDUINO-MATLAB” fue realizado en su totalidad por el señor ARIAS CASILLAS EDISON LEONARDO, como requerimiento parcial para la obtención del título de TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA.

Latacunga, 26 de agosto del 2016

**SRA. ING. EMMA CAMPAÑA
DIRECTOR DEL TRABAJO DE GRADUACIÓN**



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

AUTORÍA DE RESPONSABILIDAD

Yo, ARIAS CASILLAS EDISON LEONARDO, con cedula de identidad N° 050317343 declaro que este trabajo de grado denominado “IMPLEMENTACIÓN DE UN CONTROL DE SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE ARDUINO-MATLAB”, ha sido desarrollado en base a una investigación científica exhaustiva, respetando derechos intelectuales de terceros conforme las citas constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente, este trabajo es de mi autoría, por lo que en virtud de esta declaración, me responsabilizó del contenido, veracidad y alcance científico de la investigación mencionada.

Latacunga, 26 de agosto del 2016

ARIAS CASILLAS EDISON LEONARDO

C.C. 050317340-3



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

AUTORIZACIÓN

Yo, ARIAS CASILLAS EDISON LEONARDO, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “IMPLEMENTACIÓN DE UN CONTROL DE SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE ARDUINO-MATLAB”, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 26 de agosto del 2016

ARIAS CASILLAS EDISON LEONARDO

C.C. 050317340-3



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

DEDICATORIA

Este trabajo va dedicado con mucho cariño a mis padres Leonardo y Beatriz, de manera muy especial para mis pastores Juan y Silvia quienes son un pilar fundamental en mi vida y un incentivo para formar mi carácter para así seguir adelante a lo largo de la carrera y de una manera muy especial está dirigido a la persona que siempre estuvo a mi lado, para ti Angelita.

Edison Arias



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

AGRADECIMIENTO

Mi agradecimiento a Dios que a pesar de no contar con la economía necesaria me ha apoyado y bendecido permitiendo la culminación de mi carrera académica.

Mi agradecimiento profundo a mis ñañas Paola, Adela y Tamyra, por su apoyo y comprensión, animándome día a día a cumplir mis metas, a mis amigas, hermanos y familiares que pesar de los obstáculos confiaron en mí.

Edison Arias

ÍNDICE DE CONTENIDOS

CARATULA	
CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	ix
RESUMEN	xi
ABSTRACT	xii
CAPITULO I	13
PLANTEAMIENTO DEL PROBLEMA	13
1.1 ANTECEDENTES	13
1.2. PLANTEAMIENTO DEL PROBLEMA	14
1.3. JUSTIFICACIÓN E IMPORTANCIA	14
1.4. OBJETIVOS	15
1.5. ALCANCE	15
CAPÍTULO II	17
MARCO TEÓRICO	17
2.1 ARDUINO	17
2.2. TIPOS DE ARDUINO	17
2.3. ARDUINO UNO	27
2.4 MATLAB	30
2.5 SERVO MOTORES HITEC HS-311	34
2.6 LOS MOTORES PASO A PASO	35
2.7 APLICACIÓN EN LA ROBÓTICA	38

CAPÍTULO III	41
DESARROLLO DEL TEMA	41
3.1. PRELIMINARES	41
3.2. PUESTO EN MARCHA EL ARDUINO.....	41
3.3 DESCARGA DEL SOFTWARE PROPIO DE ARDUINO	41
3.4. INSTALACIÓN DE PAQUETE ENLACE ARDUINO-MATLAB	45
3.5. CONFIGURACIÓN DEL ARDUINO EN LA PC	46
3.6. IDENTIFICAR EL PUERTO “COM” CONECTADO A LA PLACA ARDUINO.....	46
3.7. ENLACE ARDUINO-MATLAB.....	48
3.8. CONFIGURACIÓN PARA EL ENLACE EN MATLAB	49
3.9. PROGRAMA PARA EL CONTROL DE LOS SERVOS Y EL MOTOR PASO A PASO.....	51
3.10. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL ENLACE MATLAB ARDUINO.....	55
 CAPÍTULO IV	 59
CONCLUSIONES Y RECOMENDACIONES	59
4.1. CONCLUSIONES.....	59
4.2. RECOMENDACIONES	60
 GLOSARIO DE TÉRMINOS	 61
ANEXOS	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 1 Elementos del Arduino Uno.....	28
Tabla 2 De Caracteres Especiales	32
Tabla 3 Herramientas y Paquetes	33
Tabla 4 Variables y Operadores de Matlab.....	34
Tabla 5 Evolución del Robot	39
Tabla 6 Funcionamiento del Servomotor y motor Paso a Paso	58

ÍNDICE DE FIGURAS

Figura 1 Arduino Zero	18
Figura 2 Arduino YUN	19
Figura 3 Arduino Leonardo.....	19
Figura 4 Arduino DUE.....	20
Figura 5 Arduino MEGA	21
Figura 6 Arduino ETHERNET	22
Figura 7 Arduino Fio.....	22
Figura 8 Arduino Nano	23
Figura 9 ArduinoLilyPad	24
Figura 10 Arduino Pro	24
Figura 11 Arduino Pro Mini.....	25
Figura 12 Arduino Mega ADK	25
Figura 13 Arduino Esplora.....	26
Figura 14 Arduino Robot	27
Figura 15 Arduino Uno	27
Figura 16 Elementos del Arduino Uno	28
Figura 17 Servomotor	35
Figura 18 Grados de Libertad	40
Figura 19 Descarga del Software controlador Arduino	42
Figura 20 Inicio de descarga arduino 1.6.8.....	42
Figura 21 Pantalla de licencia de arduino	43
Figura 22 Pantalla de opciones de instalación	43
Figura 23 Carpeta instalar	44
Figura 24 Pantalla fin de instalación.....	44
Figura 25 Software controlador Arduino	45
Figura 26 Descarga del Software controlador Arduino	45
Figura 27 Descomprimir Archivo	46
Figura 28 Administración de equipos	46
Figura 29 Descarga del Software controlador Arduino	47
Figura 30 Pantalla de Puerto Serial.....	47

Figura 31 Pantalla de Puerto Serial	48
Figura 32 Pantalla de Puerto Serial	49
Figura 33 Pantalla de Puerto Serial	49
Figura 34 Pantalla de Puerto Serial	50
Figura 35 Conexiones Servomotores	55
Figura 36 Conexiones Motor Paso a Paso.....	56
Figura 37 Conexión de elementos.....	56
Figura 38 Programa de enlace.....	57
Figura 39 Programa de control.....	57

RESUMEN

El objetivo a alcanzar con el presente trabajo de graduación para la obtención del Título de Tecnólogo en Electrónica mención Instrumentación & Aviónica, es el diseño y la implementación de un control de servomotores y motores paso a paso controlados mediante Arduino-Matlab, que será de gran importancia en el laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías. Se realizó un control y toda la programación haciendo uso de la comunicación serial con el software Matlab con los tres servomotores y un motor paso de 5v DC. Los mismos que reciben los datos y son energizados a través de la plataforma sencilla y económica Arduino Uno, siendo de fácil acceso y de sencilla comunicación, utilizada frecuentemente en la Simulación de Automatización de Procesos y en la Robótica. Para realizar el enlace Arduino-Matlab es necesario de los siguientes pasos fundamentales, primero se procede a descargar el paquete para el enlace desarrollado por Mathworks, siendo de código abierto, cualquier usuario puede descargarlo desde la Web; y, como segundo paso realizar la instalación del software propio del Arduino, encontrado en la versión actual en su página oficial. Alcanzando así como siguiente actividad el reconocimiento de las entradas digitales a utilizar y realizar las respectivas conexiones de los servomotores y el motor paso a paso, para posterior a ello elaborar la programación del enlace Arduino-Matlab. Como paso final del proyecto investigativo se realiza las respectivas pruebas de funcionamiento para analizar el comportamiento del Arduino y así corregir posibles errores en la programación.

PALABRAS CLAVE:

- **ARDUINO**
- **ENLACE**
- **PROGRAMACION**
- **MATLAB**
- **SERVOMOTOR**

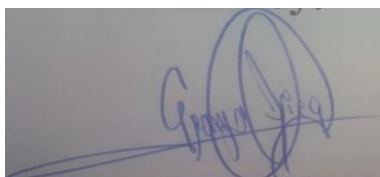
ABSTRACT

The objective to achieve with this research job to get the Electronics Technician mention Avionics and instrumentation degree, is the design and implementation of machines and servomotor control step by step controlled through Arduino – Matlab, it will be very important in the Virtual Instrumentation laboratory at Unidad de Gestión de Tecnologías. A control was performed and all programming using serial communication with Matlab software with the three servomotors and a stepper motor 5v DC. They are receiving the same data and they are energized through Arduino Uno it is an economic and simple platform it is easy to access and simple communication, it is often used in the Simulation of Process Automation and at Robotics. In order to make the Arduino-Matlab link some fundamental steps are required, first download the package for the link developed by Mathworks, it is open source, any user can download it from the Web; as a second step install the Arduino software in its actual version on its official website. Thus reaching the recognition of the digital inputs to perform the appropriate connections of the servomotors and stepper motor step by step, after that completes the programming of the Arduino-Matlab link. Finally, the operating tests are performed in order to analyze the Arduino behavior and correct out programming errors.

KEYWORDS:

- ARDUINO
- LINK
- PROGRAMMING
- MATLAB
- SERVOMOTOR

Checked by:



Lic. Diego Granja P.

Jefe Sec. Dpto. Lenguas UGT.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

TEMA: “IMPLEMENTACIÓN DE UN CONTROL DE SERVOMOTORES Y MOTORES PASO A PASO MEDIANTE ARDUINO-MATLAB”.

1.1 ANTECEDENTES

La Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas-ESPE es un campo educativo de formación tecnológica superior, que cuenta con las carreras de Mecánica Aeronáutica mención Aviones, Mecánica Aeronáutica mención Motores, Electrónica mención Instrumentación y Aviónica, Ciencias de la Seguridad mención Aérea y Terrestre, Logística y Transporte, situado en la provincia de Cotopaxi, cantón Latacunga; en la calle Xavier Espinoza y Av. Amazonas enfocada a la formación de mano de obra calificada en el área aeronáutica e industrial; para la realización de proyectos debería contar con laboratorios que le permitan poner en práctica la teoría acorde a la necesidades del ámbito laboral en que se va a desempeñar.

La carrera de Electrónica Mención Instrumentación y Aviónica, busca la actualización de sus laboratorios como es el caso de Instrumentación Virtual, a través de la implementación de nuevos módulos que tengan dispositivos de tecnología de punta para ampliar el conocimiento en la adquisición de señales analógicas y digitales.

Es necesario recalcar que existe una aplicación con Arduino-Matlab en la tesis de los señores: Moyolema Chaglla Diego Paúl y Pereira Salazar Roberto Patricio con el tema “DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL PARA EL RECORRIDO DE TRAYECTORIAS DEFINIDAS POR UN COMPUTADOR PARA EL LABORATORIO DE ROBÓTICA INDUSTRIAL DEL DECEM” en dicho proyecto se utilizaron sensores ultrasónicos HC-SR 04 para determinar distancias de 20 cm a 2 m, acelerómetro MMA7361L que permite determinar el inclinación de la superficie. (Moyolema Chaglla & Pereira Salazar, 2014),

Por ello se plantea implementar módulos con este tipo de tecnología debido a las bondades que presentan y sus múltiples aplicaciones, permitiendo a los estudiantes realicen prácticas con servomotores y motores paso a paso, controladas por el software Matlab, para así contar con otras herramientas que faciliten las simulaciones de procesos industriales en tiempo real.

1.2. PLANTEAMIENTO DEL PROBLEMA

El laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías de la Universidad de Fuerzas Armadas – ESPE, no poseían los medios y herramientas actuales necesarias para permitir a los estudiantes realicen simulaciones de procesos industriales en tiempo real.

La falta de conocimiento del control de señales a través de software y tarjetas utilizadas para adquirir datos, imposibilitaba el desarrollo de prácticas que aumente la experiencia en el avance tecnológico con Arduino y Matlab.

El desconocimiento de la programación y el diseño en la adquisición de datos puede provocar un déficit en la preparación como tecnólogos, debido que en la actualidad la mayoría de procesos industriales y equipos electrónicos son controlados por software, por lo tanto el estudiante debe estar preparado en este ámbito, lo que le permitirá tener confianza y seguridad al momento de ejercer su profesión.

1.3. JUSTIFICACIÓN E IMPORTANCIA

El control de la tarjetas Arduino Uno mediante Matlab, permite mejorar la enseñanza-aprendizaje a los estudiantes aumentando el conocimiento, cómo se realiza el control de procesos industriales, la adquisición y manipulación de señales Digitales y Analógicas.

Arduino Uno tiene diferentes aplicaciones permitiendo al estudiante dominar el proceso de adquisición de datos sean entradas digitales como el caso de un interruptor cerrado o abierto o a su vez entradas analógicas permitiendo conectar sensores de luz, temperatura o sonido.

Con las diferentes prácticas utilizando servomotores y motores paso a paso se busca obtener el conocimiento de cómo Arduino envía y controla los diferentes dispositivos electrónicos, similares a los que se utiliza en la industria, siendo preparados para su futuro desempeño en el ámbito profesional.

1.4. OBJETIVOS

1.4.1. General:

Implementar un sistema de control de Servomotores y Motores Paso a Paso a través Arduino-Matlab para el laboratorio de Instrumentación Virtual.

1.4.2. Específicos:

- Establecer las características y funcionamiento de la tarjeta Arduino Uno a través de la revisión bibliográfica para establecer la compatibilidad con Matlab.
- Instalar el Software Matlab y Arduino adquiridos en la Web para realizar el enlace y envío de señales al Servomotor y motor Paso a Paso.
- Realizar las prácticas con los Servomotores y el motor paso a paso para comprobar su correcto funcionamiento.

1.5. ALCANCE

El control de los Servomotores y Motores paso a paso a través Matlab-Arduino, se pretende optimizar la enseñanza y aprendizaje de los estudiantes de la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas - ESPE y a todo el personal que hace uso del laboratorio de Instrumentación Virtual el cual permite la familiarización de los diferentes tipos de Arduino en el momento de la adquisición y envío de datos a cualquier elemento o dispositivo a controlar.

CAPÍTULO II

MARCO TEÓRICO

2.1 ARDUINO

Es una pequeña placa de microcontrolador con un puerto USB, para conectar al ordenador y diversos zócalos de conexión, que se pueden conectar mediante el cableado a todo tipo de componentes electrónicos externos, como motores, relés, sensores de luz, diodos de láser, altavoces, micrófonos, entre otros. Se puede alimentar mediante la conexión USB del ordenador o con una pila de 9 V.

El software para programar Arduino es fácil de usar e igualmente está disponible libremente para equipos Windows, Mac y Linux sin costo alguno. Siendo Arduino una de las primeras plataformas microcontroladoras de Open Source (código abierto) en el mundo, era de esperar que se desarrollaran varias versiones de ésta. Muchos principiantes dan sus primeros pasos con el Arduino UNO, pero a medida que se avanza en el dominio de la programación y la electrónica, los requerimientos de procesamiento y número de pines de entrada y salida aumentan de manera que el UNO no es suficiente para nuestros proyectos. (Monk, 2012)

2.2. TIPOS DE ARDUINO

A continuación se detalla los diferentes modelos oficiales de Arduino:

2.2.1. Arduino Zero

En aspecto es similar a Arduino UNO, pero esta placa esconde sorpresas con respecto a la plataforma UNO. En vez del microcontrolador AtmelATmega basado en arquitectura AVR de 8 bits, el Zero contiene un potente ATMEL Cortex-M0 SAMD21 de 32 bits. Con 256 KB de memoria flash, 32 KB de SRAM y una EEPROM de más de 16KB por emulación. El voltaje en el que opera es de 3.3v/5v (7mA) y contiene 14 pines E/S digitales, de los cuales 12 son PWM y UART. En el

terreno analógico se dispone de 6 entradas para un canal ADC de 12 bits y una salida analógica para DAC de 10 bits, dos conectores tipo Micro USB el primero se lo utiliza como debug y el otro para programación. En definitiva, esta placa va destinada para los que Arduino UNO se les quede corto y necesitan algo más de potencia de procesamiento. (Gonzales, 2014)



Figura 1 Arduino Zero

Fuente: (Gonzales, 2014)

2.2.2. ArduinoYun

Se basa en el microcontrolador ATmega32u4 y en un chip AtherosAR9331 (que controla el host USB, el puerto para micro-SD y la red Ethernet/WiFi), ambos comunicados mediante un puente. El procesador Atheros soporta la distribución Linux basadas en OpenWrt llamada OpenWrt-Yun. Se trata de una placa similar a Arduino UNO pero con capacidades nativas para conexión Ethernet, WiFi, USB y micro-SD sin necesidad de agregar o comprar shields aparte. Contiene 20 pines digitales, 7 pueden ser usados en modo PWM y 12 como analógicos. El microcontrolador ATmega32u4 de 16Mhz, trabaja a 5v.y contiene una memoria de solo 32KB (4KB reservados al bootloader), SRAM de solo 2,5KB y 1KB de EEPROM. También cabe destacar que el Arduino Yún cuenta con 3 botones de reset, uno para el microcontrolador, uno para el microprocesado y otro para el Wifi. (González, 2014)



Figura 2 Arduino YUN

Fuente: (González, 2014)

2.2.3. Arduino Leonardo

Es una placa basada en un microcontrolador ATmega32u4 de bajo consumo y que trabaja a 16Mhz. La memoria flash tiene una capacidad de 32KB (4KB para el bootloader) y 2.5KB de SRAM. La EEPROM es de 1KB, también muy similar a Arduino UNO en cuanto a capacidades de almacenamiento. A nivel electrónico y de voltajes es igual al UNO. Pero este microcontrolador puede manejar 20 pines digitales (7 de ellos pueden ser manejados como PWM) y 12 pines analógicos. Como se puede apreciar contiene los mismos pines que Yun, solo que prescinde de las funcionalidades de red. El volumen ocupado por Leonardo es inferior al de UNO, puesto que carece de las inserciones de los pines y en su lugar posee perforaciones con pads de conexión en la propia placa. Además las dimensiones del conector USB de la placa es mucho menor, debido a que cuenta con una mini-USB para ahorrar espacio. Por eso es idóneo para proyectos en los que se requiera ahorrar algo de espacio, pero todo al mismo precio que UNO. (Arduino, 2016)



Figura 3 Arduino Leonardo

Fuente: (Arduino, 2016)

2.2.4. Arduino Due

Es una placa con un microcontrolador AtmelSAM3X8EARM Cortex-M3 de 32 bits. Este chip que trabaja a 84 Mhz (3,3v) aporta una potencia de cálculo bastante superior a los anteriores microcontroladores. Por eso es idóneo para todos aquellos que necesiten de un proyecto con alta capacidad de procesamiento. Al tener un core a 32 bits permite realizar operaciones con datos de 4 bytes en un solo ciclo de reloj. Además, la memoria SRAM es de 96KB, superior al resto de placas vistas anteriormente e incorpora un controlador DMA para acceso directo a memoria que intensifica el acceso a memoria que puede hacer la CPU.

Para el almacenamiento se dispone de 512KB de flash, una cantidad muy grande de memoria para cualquier código de programación. En cuanto a soporte de voltajes en intensidades es idéntica a UNO, solo que el amperaje de los pines se extiende hasta los 130-800mA (para 3v y 5v respectivamente). El sistema dispone de 54 pines de E/S digitales, 12 de ellos pueden ser usados como PWM. También tiene 12 analógicos, 4 UARTs (serie, frente a los dos de UNO), capacidades de conexión USB OTG, dos conexiones DAC (conversión digital a analógico), 2 TWI, un power jack, SPI y JTAG.



Figura 4 Arduino DUE

Fuente: (Arduino, 2016)

2.2.5. Arduino Mega

Su nombre proviene del microcontrolador que lo maneja, un ATmega2560. Este chip trabaja a 16Mhz y con un voltaje de 5v. Sus capacidades son superiores al ATmega320 del Arduino UNO, aunque no tan superiores como las soluciones

basadas en ARM. Este microcontrolador de 8 bits trabaja conjuntamente con una SRAM de 8KB, 4KB de EEPROM y 256KB de flash (8KB para el bootloader). Como puedes apreciar, las facultades de esta placa se asemejan al Due, pero basadas en arquitectura AVR en vez de ARM. En cuanto a características electrónicas es bastante similar a los anteriores, sobre todo al UNO. Pero como se puede apreciar a simple vista, el número de pines es parecido al Arduino Due: 54 pines digitales (15 de ellos PWM) y 16 pines analógicos. Esta placa es idónea para quien necesita más pines y potencia de la que aporta UNO, pero el rendimiento necesario no hace necesario acudir a los ARM-based.



Figura 5 Arduino MEGA

Fuente: (Arduino, 2016)

2.2.6. Arduino Ethernet

Esta placa es bastante similar a la UNO, incluso en el aspecto, pero tiene capacidades de red. Su microcontrolador es un ATmega328 que trabaja a 16Mhz (5v). Va acompañado de 2KB de SRAM, 1KB de EEPROM y 32KB de flash. El resto de características electrónicas son como las de UNO, solo que añade capacidad para conexión Ethernet gracias a un controlador W5100TCP/IP y facilita conectar una tarjetas de memoria microSD. Los pines disponibles son 14 digitales (4 PWM) y 6 analógicos. Lo que hay que tener en cuenta que Arduino reserva los pines 10-13 para ser usado para SPI, el 4 para la tarjeta SD y el 2 para el interruptor W5100.



Figura 6 Arduino ETHERNET

Fuente: (Arduino, 2016)

2.2.7. Arduino Fio

Es una placa Arduino reducida a la mínima expresión, su tamaño es especialmente considerado para proyectos móviles inalámbricos o para ser insertados en espacios reducidos. Funciona con un microcontrolador ATmega328P, una versión similar a la del Ethernet pero que trabaja a una frecuencia inferior de 8Mhz. Al ser tan reducida carece de ciertas comodidades, por ejemplo, para subir los sketches hay que usar un cable FTDI o una placa adicional adaptadora Sparkfun. Igualmente, las tensiones con las que se trabaja se ven mermadas hasta los 3.3-5-12v máximo. 14 pines digitales (6 PWM) y 8 pines analógicos serán los únicos disponibles en esta placa. Tampoco ayuda sus 2KB de SRAM, 32KB de flash y 1KB de EEPROM, todo esto limitará mucho el tamaño de los sketches y del circuito del proyecto.

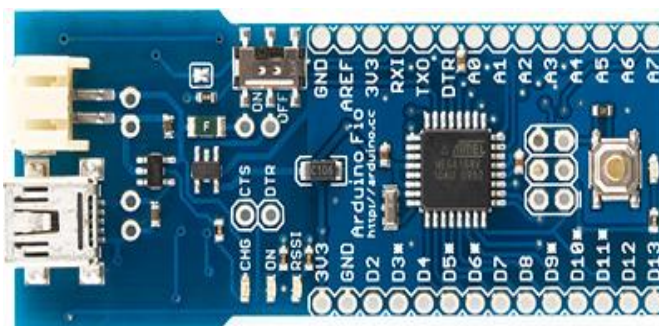


Figura 7 Arduino Fio

Fuente: (Arduino, 2016)

2.2.8. Arduino Nano

Empezaron incorporados un ATmega328 como el de otras placas vistas anteriormente, pero tras la revisión 2.x se sustituyó por un ATmega168 a 16Mhz. Sus dimensiones son aún más reducidas que las de Fio, de tan solo 18,5×43.2mm. Su reducido tamaño no le quita la posibilidad de ser una placa completa, pero sí que necesita de un cable mini-USB y no posee conector de alimentación externa. Esta versión fue diseñada y producida por la compañía Gravitech, especialmente pensado para aplicaciones de reducido costo y donde el tamaño importe. A nivel eléctrico se comporta como un UNO, con 14 pines digitales (6 PWM) y 8 analógicos. Pero sus capacidades han ido a menos con las nuevas revisiones en pos de un menor consumo. Por ello se ha pasado de 32 a 16KB de flash (2 reservados al cargador de arranque), de 2 a 1KB de SRAM y de 1KB a 512 bytes de EEPROM.



Figura 8 Arduino Nano

Fuente: (Arduino, 2016)

2.2.9. Arduino LilyPad

Está diseñado para ser integrado en prendas y textiles, se puede coser a la tela e incluir de manera similar fuentes de alimentación, sensores y actuadores con el hilo conductor. Fue desarrollado por Leah y SparkFun Electronics para ser empleado con los mismo fines que otros Arduino's solo que con ciertas limitaciones a cambio de sus capacidades de integración y su base flexible. Se basa en dos versiones de microcontrolador diferentes, ambas de bajo consumo denominadas Atmega168V y ATmega328V, esta segunda más potente. Ambos trabajan a 8Mhz, pero la primera

trabaja a solo 2,7v y a segunda a 5,5v. Dispone de 14 pines digitales (6 PWM) y 6 analógicos a lo largo de su perímetro. Además integra 16KB de memoria flash para el código del programa, 1KB de SRAM y 512 bytes de EEPROM.

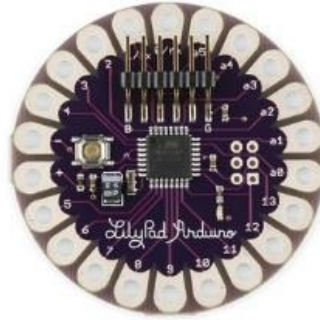


Figura 9 ArduinoLilyPad

Fuente: (Arduino, 2016)

2.2.10. Arduino Pro

Contiene un microcontrolador ATmega 168 o Atmega 328, con versiones de 3.3V y 8Mhz o 5V para 16Mhz. Contiene 14 pines de E/S digitales (6 de ellos son PWM) y 6 pines analógicos. Esta placa diseñada y construida por SparkFUN Electronics integra entre 32KB y 16KB de flash según el microcontrolador en el que se base (2KB reservados para el gestor de arranque). Lo que sí es común para todos los modelos es el KB de SRAM y los 512 bytes de EEPROM. A pesar de su nombre, la versión Pro no es una de las más potentes. Pero se ha concebido para usuarios avanzados que necesitan flexibilidad y precios bajos.

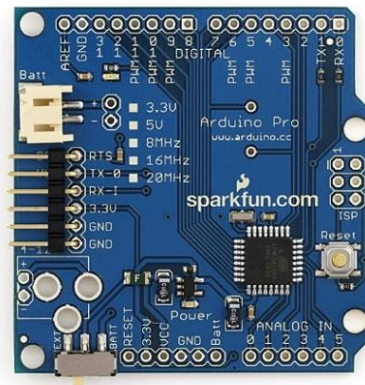


Figura 10 Arduino Pro

Fuente: (Arduino, 2016)

2.2.11. Arduino Pro Mini

Es la hermana pequeña de la versión Pro. Además de la flexibilidad y precios bajos hay que añadirle su reducido tamaño. Para reducir coste y tamaño dispone de los componentes mínimos (sin conector USB Integrado ni conectores de pin). El resto de características son muy similares a la versión Pro.



Figura 11 Arduino Pro Mini

Fuente: (Arduino, 2016)

2.2.12. Arduino Mega ADK

Placa basada en un ATmega 2560 como la versión Mega vista anteriormente, pero su principal ventaja es que dispone de una interfaz preparada para ser conectada mediante USB a dispositivos móviles basados en Android, gracias a su ICMAX 3421e. Esto permite contar con todas las capacidades de una placa Arduino Mega (igual en cuanto al resto de características) más las posibilidades de desarrollo conjuntas con una plataforma Android.

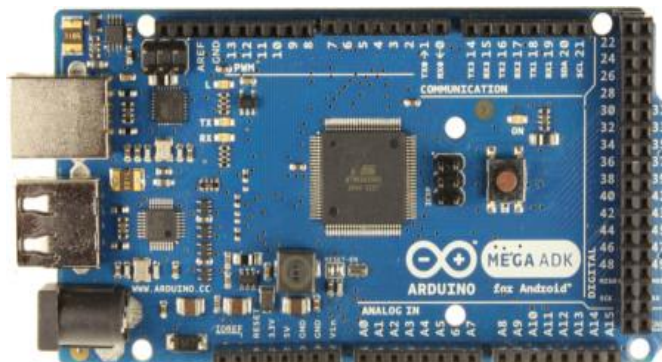


Figura 12 Arduino Mega ADK

Fuente: (Arduino, 2016)

2.2.13. Arduino Esplora

Se diferencia del resto de placas, a parte de su reducido tamaño y por su forma, en que dispone de una serie de sensores a bordo. Es bueno para las personas que comiencen a dar sus primeros pasos en el mundo de la electrónica y están aprendiendo, de ahí su nombre. Incluye unos sensores (acelerómetro, temperatura, luz), zumbador, botones, joystick, micrófono y un socket para conectar una pantalla a color TFT LCD. Funciona con un microcontrolador ATmega32u4 que trabaja a 16MHz y 5V, con SRAM de 2,5KB y 1KB de EEPROM. La flash llega a los 32KB con 4KB reservados al bootloader.

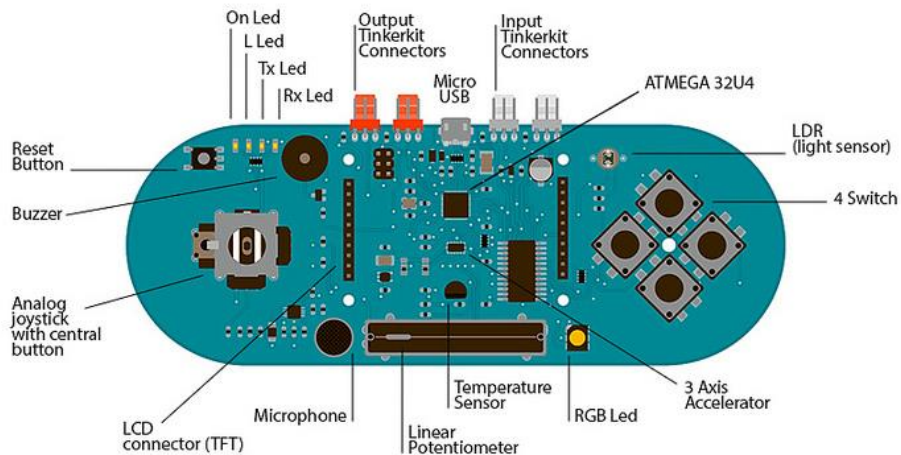


Figura 13 Arduino Esplora
Fuente: (Arduino, 2016)

2.2.14. Arduino Robot

Es el primer Arduino oficial sobre ruedas, que tiene dos procesadores, uno en cada uno de sus dos tableros. La Junta de motor controla los motores, y la de Control lee sensores y decide cómo operar. Cada una de las tablas es una programable completa placa Arduino utilizando el IDE de Arduino. Ambas juntas de motor y de control son basados en el microcontrolador ATmega32u4.

El robot tiene muchas de sus pines asignadas a los sensores y actuadores a bordo. Programación de los robots similar al proceso con el Arduino Leonardo. Ambos procesadores se han incorporado en la comunicación USB, eliminando la

necesidad de un procesador secundario. Esto permite que el robot que aparece a un ordenador conectado como virtual (CDC) de puerto serie/COM.



Figura 14 Arduino Robot

Fuente: (Arduino, 2016)

2.3. ARDUINO UNO

Es el último modelo diseñado y distribuido por la comunidad Arduino. La placa tiene un tamaño de 75x53mm. Su unidad de procesamiento consiste en un microcontrolador ATmega328. Puede ser alimentada mediante USB o alimentación externa y contiene pines tanto analógicos como digitales.



Figura 15 Arduino Uno

Fuente: (Arduino, 2016)

2.3.1 Especificaciones del Arduino Uno

Tabla 1

Elementos del Arduino Uno

N°	Especificaciones	Rango
1	Microcontrolador	Atmega 328
2	Voltaje de Operación	5 V
3	Voltaje de Entrada	7–12 V
4	Voltaje de Salida	6-20 V
5	I/O Digitales	14 (6PWM)
6	Entradas Análogas	6
7	Memoria Flash	23 KB
8	SRAM	2 KB
9	EEPROM	1KB
10	Velocidad de Reloj	16 Mhz

Fuente: (USA ONLY Arduino , 2016)

2.3.2. Elementos del Arduino

A continuación se muestra en la ilustración donde están ubicados los elementos más importantes que componen la placa Arduino Uno, que son descritos de arriba y de izquierda a derecha:

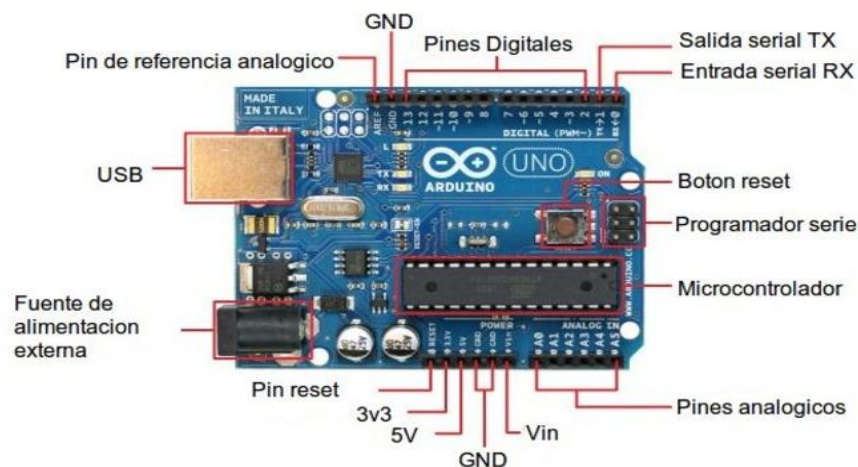


Figura 16 Elementos del Arduino Uno

Fuente: (Arduino, 2016)

- **Referencia para pines analógicos (AREF).**- Tensión de referencia para entradas analógicas. Se utiliza con la función “analog Refence”.
- **Pines de tierra (GND).**- Masa del circuito para pines, es decir es la tensión de refencia de 0V.
- **Pines digitales de entrada y salida.**- Hay 14 pines digitales que pueden utilizarse como entrada o salida con las funciones pinMode, digitalWrite, y digitalRead. Operan a 5 voltios.

Cada pin proporciona o recibe como máximo 40mA y disponen de una resistencia pull-up (desconectada por efecto) de 20-50kOhmios. Ciertos pines son reservados para determinados usos:

- **Serie:** 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie. Están directamente conectados a los pines serie del microcontrolador. Utilizando estos pines podremos conectarnos con otras placas.
 - **Interruptores externos:** 2 y 3. Estos pines pueden ser configurados para activar interruptores.
 - **PWM:** 3, 5, 6, 10 y 11. Proporcionan una salida de 8 bits en modo PWM.
 - **SPI:** 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI.
 - **LED:** 13. Este pin está conectado con un led de la placa. Cuando se le asigne un valor HIGH se encenderá, en cambio sí lo dejamos en LOW estará apagado.
- **Conector USB.**- Existen varios tipos de conectores USB, en concreto esta placa utiliza el tipo B hembra y deberá conectarse a un conector tipo A hembra a nuestra PC (por ejemplo a un ordenador o al cargador de un móvil). La placa se puede alimentarse con la tensión de 5V que le proporciona el bus serie USB.
Al cargar un programa a la placa desde el software de Arduino se inyectara el código del ordenador por este bus.

2.4 MATLAB

2.4.1. Historia

Fue creado por el matemático y programador de computadoras Cleve Moler en 1984, surgiendo la primera versión con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas en dicho lenguaje.

El lenguaje de programación M fue creado en 1970 para proporcionar un sencillo acceso al software de matrices LINPACK y EISPACK sin tener que usar Fortran. En 2004, se estimaba que MATLAB era empleado por más de un millón de personas en ámbitos académicos y empresariales.

2.4.2 Definición

Matlab es una de las muchas sofisticadas herramientas de computación disponibles en el comercio, para resolver problemas matemáticos especialmente es superior al desarrollar cálculos que involucren matrices. El nombre MATLAB es una abreviatura de MATrixLABoratory, "laboratorio de matrices" es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M) y servicio de especie. Está disponible para las plataformas Unix, Windows, Mac OS X s, y GNU/Linux.

2.4.3 Operadores Aritméticos

Matlab tiene dos tipos diferentes de operadores aritméticos:

- **Operadores Matriciales:** Gobernados por las reglas de álgebra lineal
- **Operadores Vectoriales:** Se llevan a cabo elemento a elemento.

El punto precediendo al signo del operador distingue a las operaciones vectoriales de las operaciones matriciales. Los pares ".+" y "-." no se utilizan puesto que las operaciones de suma y resta vectorial y matricial coinciden.

2.4.4 Documentación de Matlab

La documentación de MATLAB incluye ayuda en línea y manuales impresos:

- **MATLAB USER'S GUIDE:** Contiene aspectos específicos de la plataforma acerca de su uso y un tutorial con las funcionalidades básicas de Matlab.
- **MATLAB REFERENCE GUIDE:** Compendio alfabético de todos los comandos de Matlab.
- **EXTERNAL INTERFACE GUIDE:** Describe los interfaces externos de Matlab, incluyendo la importación y exportación de datos, librerías FORTRAN y C para linkado dinámico, lectura y escritura de ficheros y llamada a Matlab como motor de cálculo.
- **INSTALLATION GUIDE:** Describe cómo instalar Matlab.
- **ONLINE HELP:** La ayuda en línea del programa y las demos proporcionan información acerca de los comandos de Matlab y muestran algunas de sus características.
- **AYUDA EN INTERNET:** En internet se puede encontrar una gran variedad de sitios con información abundante sobre MATLAB. Desde el text de la propia casa MATHWORKS, hasta FAQs (Preguntas más frecuentes), tutoriales, ayudas y foros.

2.4.5 Categoría de funciones

MATLAB proporciona 20 categorías de funciones. Algunas de las funciones están implementadas en el intérprete, mientras que otras están escritas en forma de fichero .m los mismos que están organizados en 20 directorios. El comando help muestra la siguiente tabla de categorías:

- **Color:** Funciones de control de color y modelo de iluminación.
- **datafun:** Análisis de datos y transformada de Fourier.
- **demos:** Demostraciones y ejemplos.
- **elfun:** Funciones matemáticas elementales.
- **elmat:** Matrices elementales y manipulación de matrices.
- **funfun:** Funciones de funciones (métodos numéricos no lineales).
- **general:** Comandos de propósito general.

- **graphics:** Funciones gráficas de propósito general.
- **iofun:** Funciones de E/S de ficheros de bajo nivel.
- **lang:** Construcciones de lenguaje y depuración.
- **matfun:** Funciones matriciales (álgebra lineal numérica).
- **ops:** Operadores y caracteres especiales.
- **plotxy:** Gráficos bidimensionales.
- **plotxyz:** Gráficos tridimensionales.
- **polyfun:** Funciones polinomiales y de interpolación.
- **sparfun:** Funciones de matrices dispersas (sparse).
- **specfun:** Funciones matemáticas especializadas.
- **specmat:** Matrices especializadas.
- **sounds:** Funciones de procesamiento de sonidos.
- **strfun:** Funciones de cadenas de caracteres.

2.4.6 Caracteres Especiales

Tabla 2

De Caracteres Especiales

Nº	Símbolo	Definición
1	[]	Se utiliza para componer vectores y matrices
2	()	Indica procedencia de expresiones aritméticas
3	=	Utilizado en sentencias de Asignación
4	‘	Transpuesta de la matriz
5	.	Punto Decimal. Indicador de decimales elemento a elemento
6	...	Indica que continuara en la línea de abajo
7	,	Separa índices de la matriz y argumentos de las funciones
8	;	Finaliza la línea, después de una expresión la separa de otra
9	%	Denota un comentario, indica el fin lógico de una línea
10	!	Indica que el resto de la línea es un comando operativo
11	:	Crea vectores, como índice de matrices y para iteraciones

Fuente: (The MathWorks, Inc., 2016)

2.4.7 Características básicas del Matlab

- Cálculos intensivos desde un punto de vista numérico.
- Gráficos y visualización avanzada.
- Lenguaje de alto nivel basado en vectores, arrays y matrices.
- Colección muy útil de funciones de aplicación.

2.4.8 Cajas de Herramientas y Paquetes

Tabla 3

Herramientas y Paquetes

Nº	MATLAB	SIMULINK
Cajas de herramientas		
1	Matemáticas y Optimización	Modelado de punto fijo
2	Estadísticas y Análisis de datos	Modelado basado en eventos
3	Diseño de sistemas de control y análisis	Modelado físico
4	Procesado de señal y comunicaciones	Gráficos de simulación
5	Procesado de imagen	Grafica de simulación
6	Pruebas y medidas	Procesado de señal y comunicaciones
7	Biología computacional	Generación de códigos
8	Modelado y análisis financiero	Prototipos de control rápido y SW/HW HIL
9	Desarrollo de aplicaciones	Tarjetas integradas
10	Informe y conexiones a base de datos	Verificación, validación y comprobación

Fuente: (The MathWorks, Inc., 2016)

2.4.9 Variables y Operadores.- Los operadores básicos que usa Matlab son:

Tabla 4

Variables y Operadores de Matlab

N°	Símbolo	Variable y operador
1	+	Suma
2	-	Resta
3	*	Multiplicación
4	/	División
5	<	Menor que
6	< =	Menor o igual
7	= =	Igual
8	>	Mayor que
9	> =	Mayor o igual
10	~ =	No igual

Fuente: (The MathWorks,Inc., 2016)

2.4.10 Beneficios de Matlab

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas
- Redes neuronales y otros

2.5 SERVO MOTORES HITEC HS-311

Es un aparato destinado a vencer grandes resistencias, mediante la amplificación de fuerzas aplicadas, relativamente más pequeñas, que consta de dos bombines hidráulicas. El servomotor fue diseñado como un actuador mecánico, en cualquier sistema de servomecanismo que tiene por objeto llevar al sistema a una

lectura cero, es decir la señal de error se redujo a cero, y existen dos tipos de servomotores, de corriente directa y de corriente alterna.

Sus características fundamentales para DC y AC se basaron en lo siguiente: la primera radicó en que el par de salida del motor sea aproximadamente proporcional a su voltaje de control aplicado y la segunda se basa en el hecho de que la dirección del par éste determinada por la polaridad instantánea del voltaje de control. El servomotor de DC campo controlado es aquel donde el par que produjo este motor es cero cuando el amplificador de error de DC no le suministra excitación de campo, y si se invierte la polaridad del campo, se invierte la dirección del motor.



Figura 17 Servomotor

Fuente: (González Victor, 2002-2003)

2.6 LOS MOTORES PASO A PASO

Los motores paso a paso se pueden ver como motores eléctricos sin escobillas. Es típico que todos los bobinados del motor sean parte del estator, y el rotor puede ser un imán permanente o en el caso de los motores de reluctancia variable, un cilindro sólido con un mecanizado en forma de dientes (similar a un engranaje), construido con un material magnéticamente "blando" (como el hierro dulce).

La conmutación se maneja de manera externa con un controlador electrónico, típicamente los motores y sus controladores se diseñan para que el motor se pueda mantener en una posición fija y también para que se lo pueda girar en un sentido y en el otro. La mayoría de los motores paso a paso conocidos se pueden avanzar a frecuencias de audio, lo que permite girar a gran velocidad. Con un controlador apropiado se puede arrancar y detener en un instante con posiciones controladas.

2.6.1 Comportamiento propio de los motores paso a paso:

Los motores paso a paso tienen un comportamiento diferente al de los motores de corriente continua. En primer lugar, no giran libremente por sí mismos, como lo indica su nombre avanzan girando por pequeños pasos. También difieren de los motores de CC en la relación entre velocidad y torque (un parámetro que también es llamado "par motor" y "par de giro"). Los motores de CC no son buenos para ofrecer un buen torque a baja velocidad sin la ayuda de un mecanismo de reducción, los motores paso a paso en cambio trabajan de manera opuesta: su mayor capacidad de torque se produce a baja velocidad.

Los motores paso a paso tienen una característica adicional: el torque de detención (que se puede ver mencionado también como "par de detención", e incluso par/torque "de mantenimiento"), que no existe en los motores de CC. El torque de detención hace que un motor paso a paso se mantenga firmemente en su posición cuando no está girando. Esta característica es muy útil cuando el motor deja de moverse y, mientras está detenido, la fuerza de carga permanece aplicada a su eje. Se elimina así la necesidad de un mecanismo de freno.

Si bien es cierto que los motores paso a paso funcionan controlados por un pulso eléctrico de avance, el mismo no se realiza aplicando directamente. Estos motores tienen varios bobinados, para producir el avance de ese paso, deben ser alimentados en una adecuada secuencia. Si se invierte el orden de esta secuencia, se logra que el motor gire en sentido opuesto. Si los pulsos de alimentación no se proveen en el orden correcto, el motor no se moverá apropiadamente, puede ser que zumbe y no se mueva, o puede ser que gire, pero de una manera brusca e irregular.

Esto significa que hacer girar un motor paso a paso no es tan simple como hacerlo con un motor de corriente continua, al que se le entrega una corriente y listo. Se requiere un circuito de control, que será el responsable de convertir las señales de avance de un paso y sentido de giro en la necesaria secuencia de energización de los bobinados.

2.6.2 Características comunes de los motores paso a paso:

Un motor paso a paso se define por estos parámetros básicos:

- **Voltaje.-** Los motores paso a paso tienen una tensión eléctrica de trabajo. Este valor viene impreso en su carcasa o por lo menos se especifica en su hoja de datos. Algunas veces puede ser necesario aplicar un voltaje superior para lograr que un determinado motor cumpla con el torque deseado, pero esto producirá un calentamiento excesivo y/o acortará la vida útil del motor.
- **Resistencia eléctrica.-** Otra característica de un motor paso a paso es la resistencia de los bobinados. Esta resistencia determinará la corriente que consumirá el motor, y su valor afecta la curva de torque del motor y su velocidad máxima de operación.
- **Grados por paso.-** Generalmente, este es el factor más importante al elegir un motor paso a paso para un uso determinado. Este factor define la cantidad de grados que rotará el eje para cada paso completo. Una operación de medio-paso o semi-paso (half step) del motor duplicará la cantidad de pasos por revolución al reducir la cantidad de grados por paso. Cuando el valor de grados por paso no está indicado en el motor, es posible contar a mano la cantidad de pasos por vuelta, haciendo girar el motor y sintiendo por el tacto cada "diente" magnético.

Los grados por paso se calculan dividiendo 360 (una vuelta completa) por la cantidad de pasos que se contaron. Las cantidades más comunes de grados por paso son: $0,72^\circ$, $1,8^\circ$, $3,6^\circ$, $7,5^\circ$, 15° y hasta 90° . A este valor de grados por paso usualmente se le llama la resolución del motor.

En el caso de que un motor no indique los grados por paso en su carcasa, pero sí la cantidad de pasos por revolución, al dividir 360 por ese valor se obtiene la cantidad de grados por paso.

Un motor de 200 pasos por vuelta, por ejemplo, tendrá una resolución de $1,8^\circ$ por paso.

2.7 APLICACIÓN EN LA ROBÓTICA

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas. La Organización Internacional de Estándares (ISO), incluye la necesidad de tener varios grados de libertad y define al robot industrial como:

- **Robot:** Manipulador multifuncional reprogramable con varios grados de libertad, capaz de manipular materia, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas.

Asociación Francesa de Normalización (AFNOR) define al robot de una forma más completa, tomando en cuenta la definición de manipulador, posterior a ello el concepto de robot:

- **Manipulador:** mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivos lógicos.
- **Robot:** manipulador automático servocontrolado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas.

Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno.

Tabla 5

Evolución del Robot

Generación	Nombre	Tipo de Control	Grado de movilidad	Usos más frecuentes
1(1982)	Pick & Place	Fines de carrera, aprendizaje	Ninguno	Manipulación de Servicio de maquinas
2(1984)	Servo	Servocontrol, trayectoria continua, progreso condicional	Desplazamiento por vía	Soldadura, Pintura
3(1989)	Ensamblado	Servos de precisión, visión, tacto, progreso. Off-line	AGV Guiado por vía	Ensamblado, Desbarbado
4(2000)	Móvil	Sensor inteligente	Patas Ruedas	Construcción Mantenimiento
5(2010)	Especial	Controlados con técnicas de IA	Andante Saltarín	Uso militar Uso espacial

Fuente: (Barrientos, Peñin, Balaguer, & Aracil)

2.7.1 Morfología del robot

Un robot está formado por los siguientes elementos:

- Estructura mecánica,
- Transmisiones,
- Sistemas de accionamiento,
- Sistema sensorial,
- Sistema de control,
- Elementos terminales

2.7.2 Estructura Mecánica

Mecánicamente, un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. La constitución física de la mayor parte de robot industrial guardan cierta similitud con la anatomía del brazo humano, por lo que se usan términos como: brazo, codo y muñeca. El movimiento de cada articulación puede ser:

- Desplazamiento,
- Giro
- Combinación de ambos.

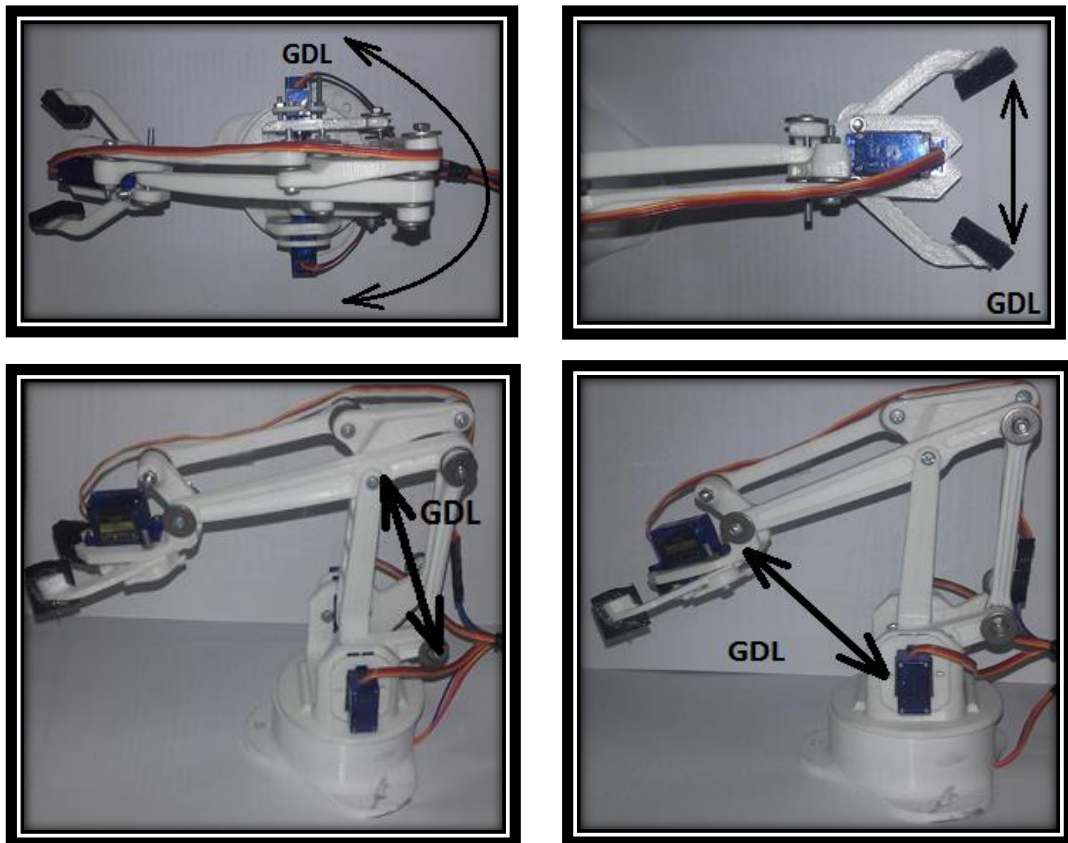


Figura 18 Grados de Libertad

Cada movimiento independiente que puede realizar cada articulación con respecto a la anterior, se denomina “grado de libertad” (GDL). El número de grado de libertad suele coincidir con el número de articulaciones las mismas que son únicamente de rotación y prismática.

CAPÍTULO III

DESARROLLO DEL TEMA

3.1. PRELIMINARES

En el presente capítulo se detalla paso a paso como se realizó el enlace entre la placa Arduino-Matlab para controlar al Servomotor y el Motor paso a paso. Para el enlace se puede utilizar las diferentes versiones de Arduino como: Leonardo, MegaADK, entre otros y en el presente proyecto será necesario Arduino Uno. Se crea un programa dentro de Matlab haciendo uso del GUI, el cual, al pulsar algún botón se podrá observar la acción real por parte de la placa, al realizar giros en el Servomotor y en el motor Paso a Paso.

3.2. PUESTO EN MARCHA EL ARDUINO

Cuando se compra una placa Arduino Uno o Duemilanove, por lo general suelen venir con un programa Blink de muestra preinstalado que hará parpadear el pequeño LED que trae integrado. Para que funcione el Arduino es necesario proporcionarle la alimentación. La manera más fácil de alimentarla es conectarla al puerto USB del ordenador de conexión tipo A y tipo B. Si todo funciona bien, el LED de la placa debe parpadear una vez cada dos segundos. La razón por la cual las nuevas placas Arduino tiene este programa Blink (parpadeo) es que cuando está instalado, sirve para verificar que la placa funcione en una forma correcta.

3.3 DESCARGA DEL SOFTWARE PROPIO DE ARDUINO

Se debe descargar la última versión de Arduino, la misma que se encuentra en su página oficial (www.arduino.cc/cc.es), según el tipo de ordenador disponible, sea para Windows o Mac, para el presente proyecto se adquirió para la plataforma Windows. Este programa se ejecuta en el ordenador, permite crear programas y

descargarlos en la placa. Diríjase a la opción Download y seleccione la descarga para Windows installer, como se muestra en la Figura.



Figura 19 Descarga del Software controlador Arduino
Fuente: (Arduino, 2016)

- Seleccione la opción Guardar archivo del dialogo, esto iniciará la descarga del instalador que contiene el software de arduino como se muestra en la figura.

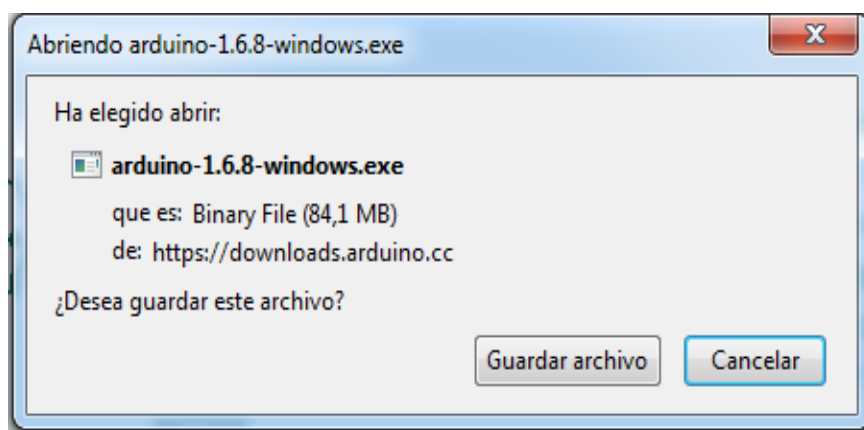


Figura 20 Inicio de descarga arduino 1.6.8
Fuente: (Arduino, 2016)

- Como siguiente paso se muestra la pantalla de Licencias, seleccione la opción I Agree como se puede mirar en la figura.

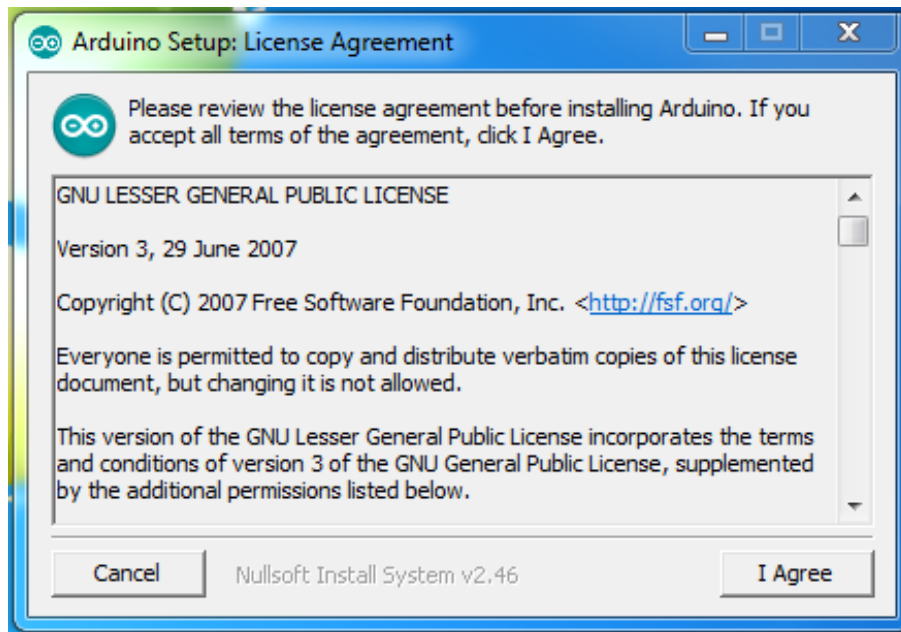


Figura 21 Pantalla de licencia de arduino

Fuente: (Arduino, 2016)

- A continuación se puede apreciar la pantalla Opciones de Instalación, en la cual se debe seleccionar la opción Next, tal como se observa en la figura.

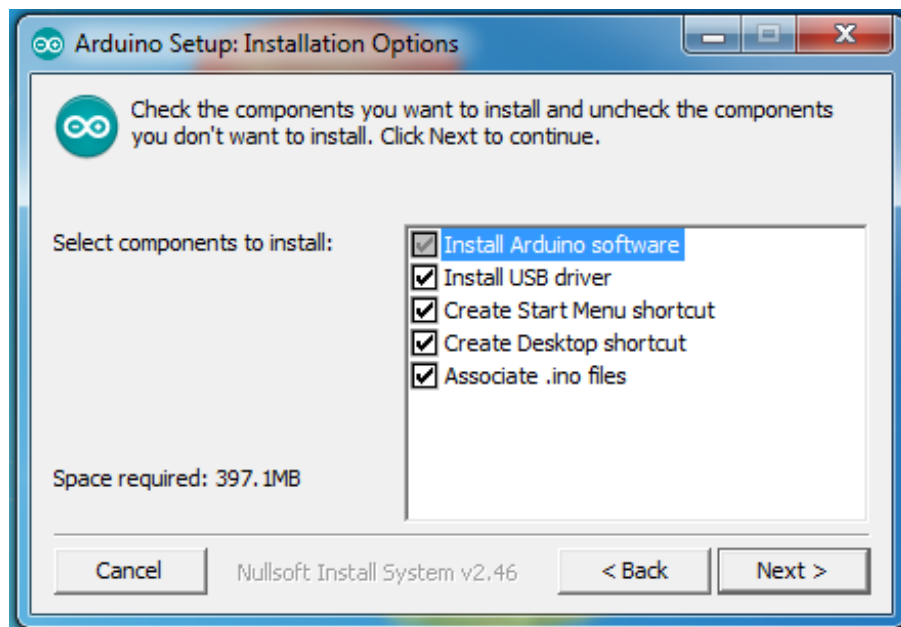


Figura 22 Pantalla de opciones de instalación

Fuente: (Arduino, 2016)

- Siguiendo el proceso de instalación se indica la siguiente pantalla, a donde va dirigida la carpeta Arduino, realice un clic en la opción Install, como se puede mirar en la figura.

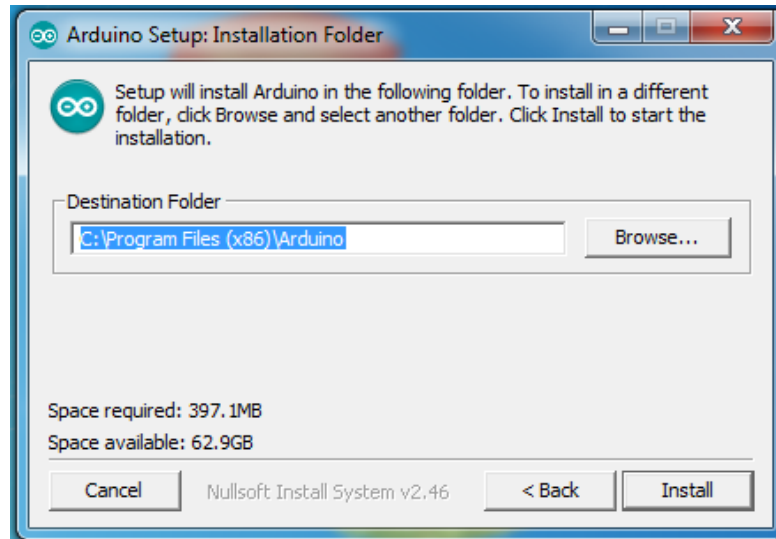


Figura 23 Carpeta instalar

Fuente: (Arduino, 2016)

- Una vez completada la instalación, como paso final escoja y de Clic en la última opción Close.

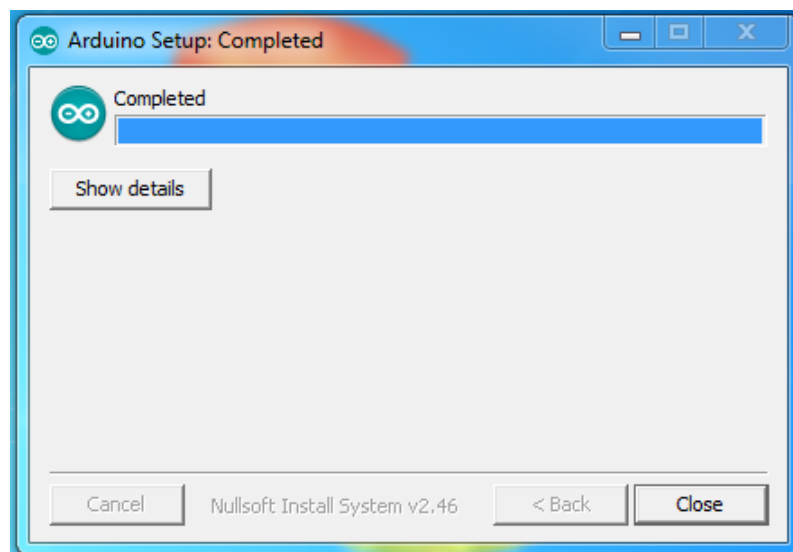


Figura 24 Pantalla fin de instalación

Fuente: (Arduino, 2016)

- Terminada la instalación del software propio de Arduino abra el sketch_mar31a l Arduino 1.0.6. programa el cual permite realizar los diferentes proyectos para descargarlos en la placa Arduino.

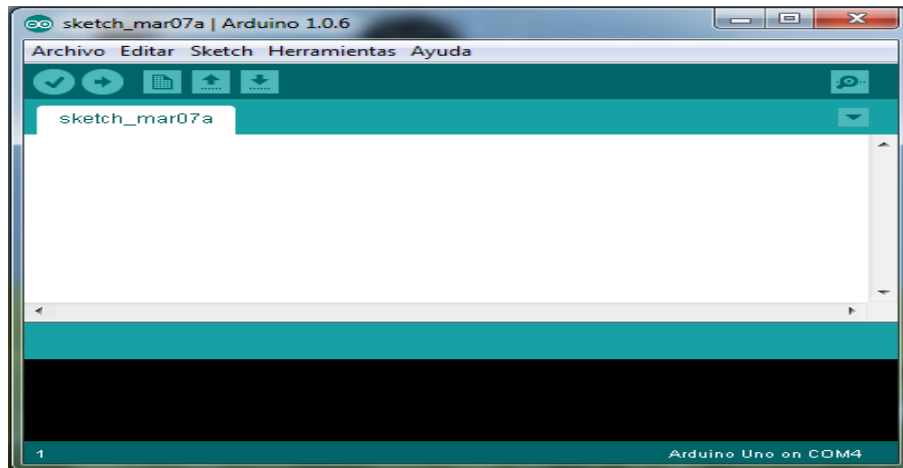


Figura 25 Software controlador Arduino
Fuente: (Arduino, 2016)

3.4. INSTALACIÓN DE PAQUETE ENLACE ARDUINO-MATLAB

Para realizar el enlace Arduino–Matlab es necesario descargar un paquete de datos proporcionado por Mathwoks en su página oficial, o a su vez realizar un click directamente en esta dirección: <http://www.mathworks.com/matlabcentral/fileexchange/32374-matlab-support-package-for-arduino-aka-arduinoio-package?Download=true>, seleccione Download Zip y, esto iniciara la descarga del archivo Zip.

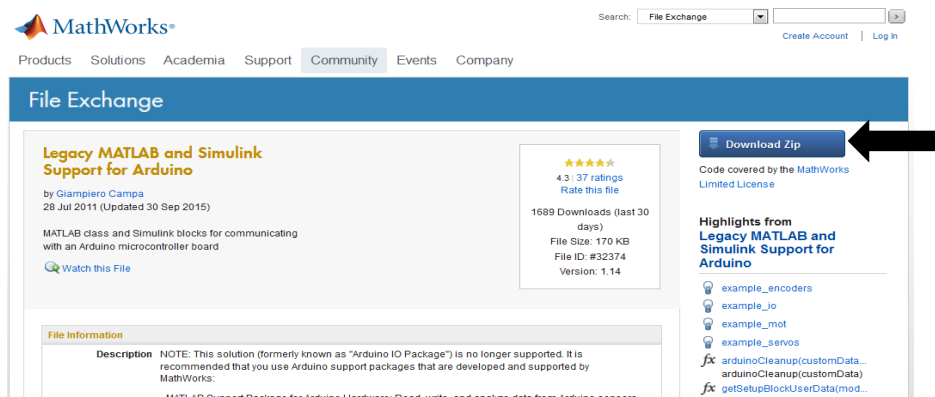


Figura 26 Descarga del Software controlador Arduino
Fuente: (The MathWorks,Inc., 2016)

- El archivo zip se debe descomprimir y guardar en el disco local C, para mayor seguridad.

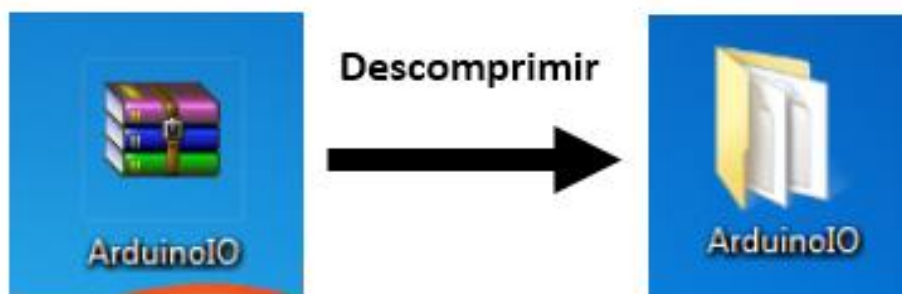


Figura 27 Descomprimir Archivo

3.5. CONFIGURACIÓN DEL ARDUINO EN LA PC

Conectar el cable USB de la placa arduino al ordenador para iniciar la comunicación, programación y enviar mensajes. Observar el encendido de los led's, así verificar su correcto reconocimiento. Una vez instalado Matlab en nuestra PC y descargado los archivos antes mencionados, se procede a crear el nuevo SKETCH.

3.6. IDENTIFICAR EL PUERTO “COM” CONECTADO A LA PLACA ARDUINO

- Inicio > Equipo > “click derecho” > Administrados > Administrador de dispositivos > Puertos > Arduino uno. En el lado derecho de “ARDUINO UNO” sale COMX, siendo “X” un nuevo a escoger de “0 y 6” usualmente.

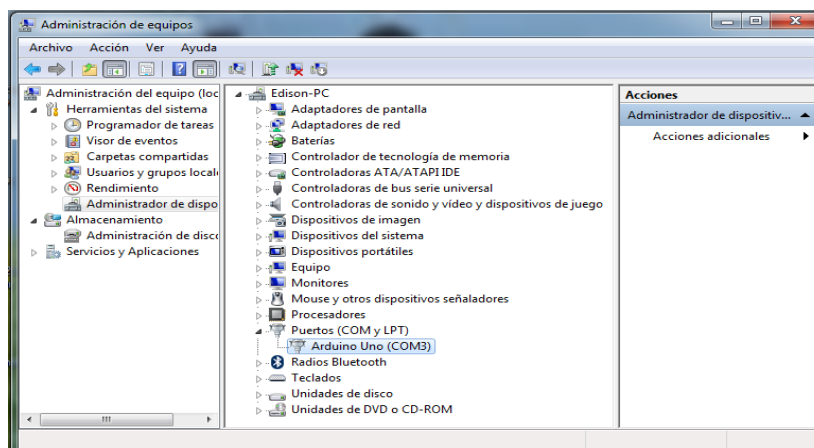


Figura 28 Administración de equipos

- Configurar en el software el tipo de Arduino a utilizar, en el presente proyecto se utilizara Arduino uno, como se muestra en la figura: Herramienta > Tarjeta > elegir la “placa Arduino Uno”

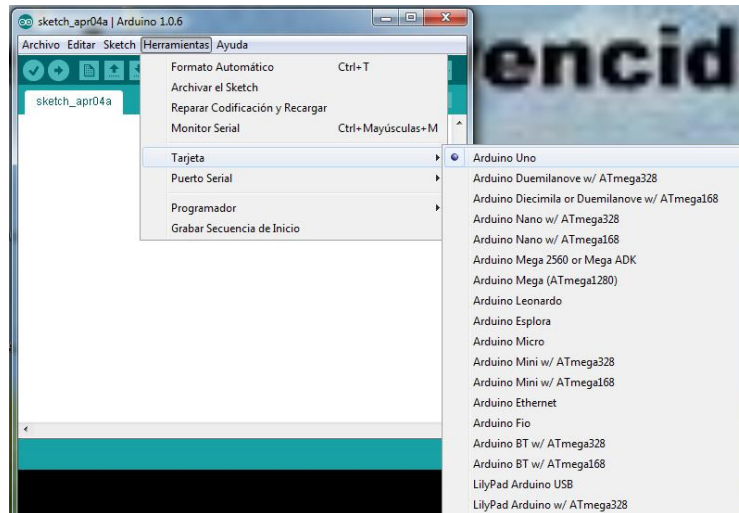


Figura 29 Descarga del Software controlador Arduino
Fuente: (Arduino, 2016)

- Seleccione el puerto serial a utilizar, diríjase a: Herramienta > Puerto Serial > y seleccionar COM3.

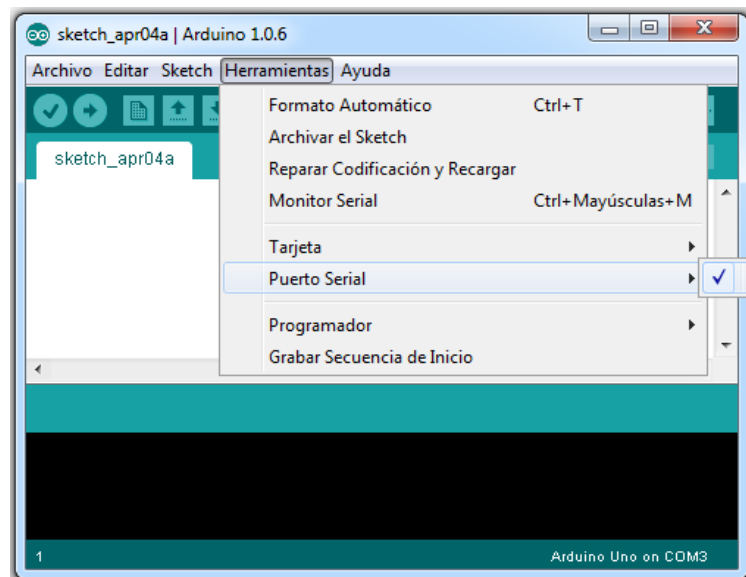


Figura 30 Pantalla de Puerto Serial
Fuente: (Arduino, 2016)

3.7. ENLACE ARDUINO-MATLAB

Como primer paso para el enlace introducir un sketch en la placa Arduino, el mismo que permitirá realizar la conexión entre el ARDUINO y MATLAB, con ello se consigue que la placa pueda entender las órdenes enviadas desde MATLAB:

Dentro del programa Arduino:

Archivo > Abrir (buscamos dentro de los archivos descargados en la dirección): Arduino > pde > adiosrv.pde

- Teniendo el adiosrv abierto le damos a “Cargar” y espere hasta que diga “Carga Terminada”.

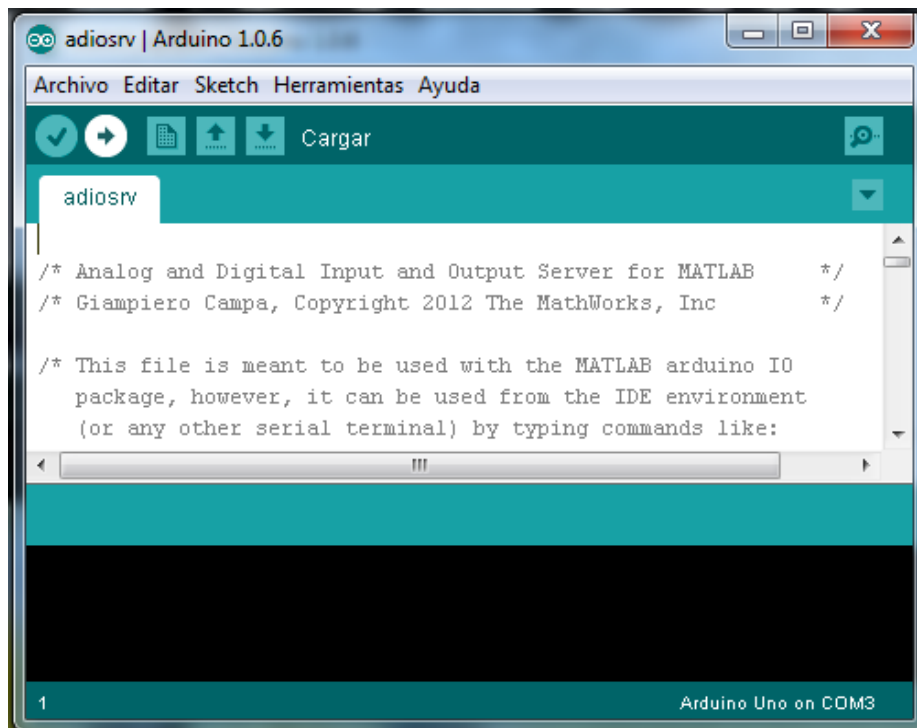


Figura 31 Pantalla de Puerto Serial

Fuente: (Arduino, 2016)

- Cerrar el “ARDUINO” para desocupar el Puerto, así poder establecer la conexión a través del Matlab.

3.8. CONFIGURACIÓN PARA EL ENLACE EN MATLAB

Una vez abierto el software Matlab proceder a agregar el paquete proporcionado por Mathwoks para realizar la comunicación con la tarjeta Arduino, realice los pasos descritos a continuación:

- Matlab > (click derecho) > Ejecutar como administrador.

El directorio “Current Folder” lo ubicamos dentro de la carpeta “ArduionIO”.

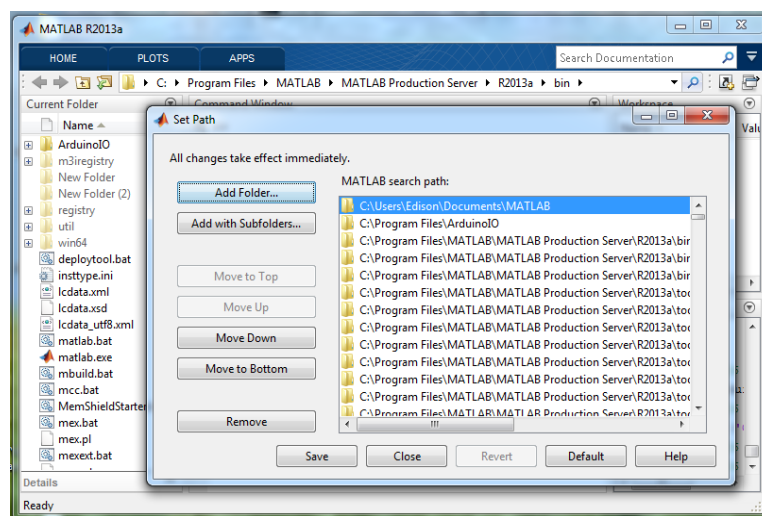


Figura 32 Pantalla de Puerto Serial

Fuente: (Arduino, 2016)

En Matlab digite los siguientes comandos:

- `install_arduino` (sin comillas), como se muestra en la figura.

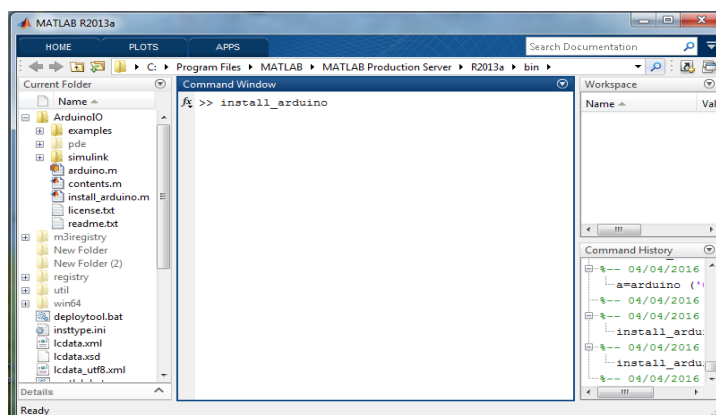


Figura 33 Pantalla de Puerto Serial

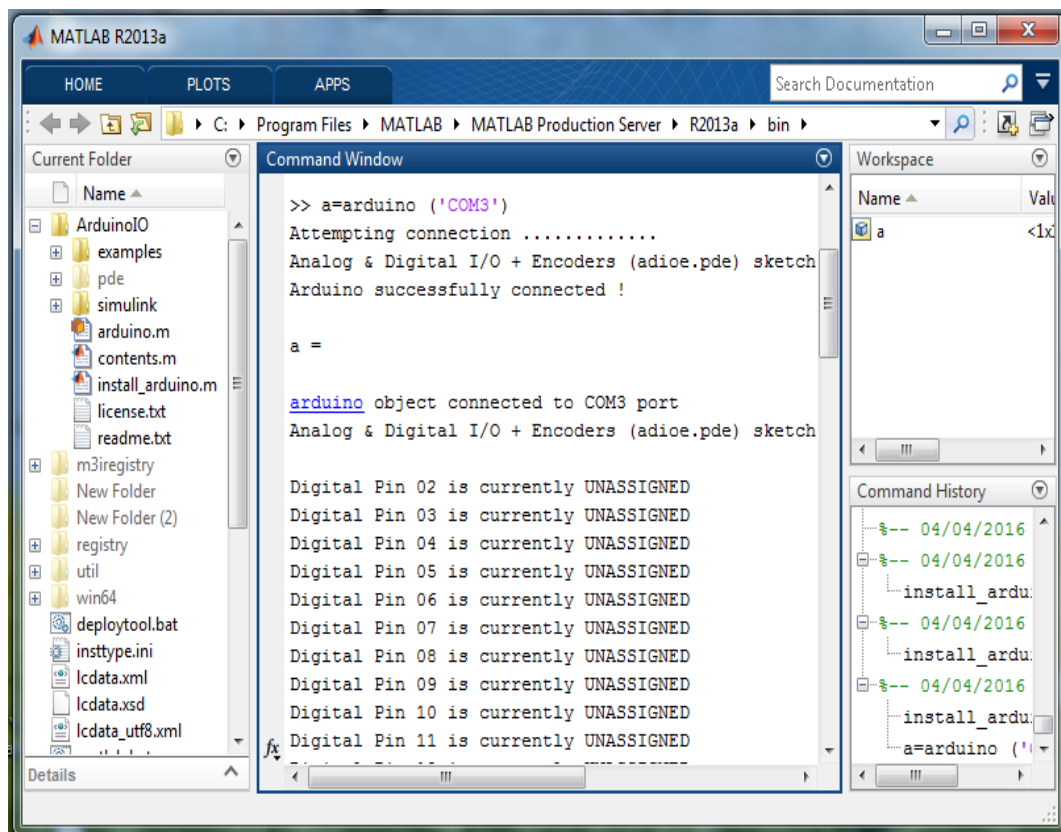
Fuente: (Arduino, 2016)

Nota: Si sale en pantalla “error”, existió una equivocación en los paso de la configuración del paquete para el enlace, se escribió mal “install_arduino” o ya se ha instalado anteriormente.

- Como siguiente paso configurar el puerto a utilizar, digitar: `a=arduino ('COM3')`

Dónde: 3 es el número hallado en el administrador de la PC.

Justo en ese momento aparecerán algunas instrucciones que serán de guía para el usuario al momento de la programación.



The screenshot shows the MATLAB R2013a interface. The Command Window displays the following text:

```
>> a=arduino ('COM3')
Attempting connection .....
Analog & Digital I/O + Encoders (adiao.pde) sketch
Arduino successfully connected !

a =

arduino object connected to COM3 port
Analog & Digital I/O + Encoders (adiao.pde) sketch

Digital Pin 02 is currently UNASSIGNED
Digital Pin 03 is currently UNASSIGNED
Digital Pin 04 is currently UNASSIGNED
Digital Pin 05 is currently UNASSIGNED
Digital Pin 06 is currently UNASSIGNED
Digital Pin 07 is currently UNASSIGNED
Digital Pin 08 is currently UNASSIGNED
Digital Pin 09 is currently UNASSIGNED
Digital Pin 10 is currently UNASSIGNED
Digital Pin 11 is currently UNASSIGNED
```

The Workspace window shows a variable 'a' with a value of '<1x...'. The Command History window shows the command 'a=arduino ('COM3')' being executed on 04/04/2016.

Figura 34 Pantalla de Puerto Serial
Fuente: (Arduino, 2016)

Tras haber ejecutado el procedimiento, ahora se tiene el enlace o conexión entre Matlab y Arduino

3.9. PROGRAMA PARA EL CONTROL DE LOS SERVOS Y EL MOTOR PASO A PASO

```
String text_in="";
String ar[5];
int begin_idx=0,idx;

#include <Servo.h>

Servo hombro;
Servo brazo;
Servo mano;

////////////////////////////////////VOID SETUP
void setup(){

  Serial.begin(9600);

  Serial.println("DEMO ROBOT");

  hombro.attach(4);
  brazo.attach(7);
  mano.attach(8);

  hombro.write(0);
  brazo.write(0);
  mano.write(0);

  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);

  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  digitalWrite(13, LOW);

}

////////////////////////////////////VOID LOOP
void loop(){
  while(1){
    rx_serial();
  }
}
```

```

void rx_serial(){
  if (Serial.available()){
    char c = Serial.read();
    switch(c){
      case '@':
        text_in="";
        begin_idx=0;

        break;

      case '<':
        if(text_in!=""){

          Serial.print("T: ");
          Serial.println(text_in);

          for(uint8_t w=0;w<5;w++){
            idx = text_in.indexOf("&",begin_idx+1);
            ar[w] = text_in.substring(begin_idx+1,idx);
            Serial.println(ar[w]);
            begin_idx = idx;
          }

          text_in = "";

          if(ar[1]=="PROCESO1"){

            sec1();

          }
          if(ar[1]=="PROCESO2"){
            sec2();
          }

          if(ar[1]=="PROCESO3"){
            sec3();
          }
        }

        break;
      default:
        text_in+=c;
        break;
    }
  }
}

void sec1(){
  // base.write(90); hombro.write(0); brazo.write(0); mano.write(0);
  // delay(1500);
}

```

```

for(int w=0;w<90;w++){
    mano.write(w);
    delay(10);
}
delay(2000);

for(int w=0;w<90;w++){
    hombro.write(w); brazo.write(0);
    delay(10);
}
delay(1000);

for(int w=90;w>0;w--){
    mano.write(w);
    delay(10);
}
mano.write(0);
delay(1000);

for(int w=90;w>0;w--){
    hombro.write(w);
    delay(10);
}
delay(1000);

}

void sec2(){

for(int w=0;w<120;w++){
    digitalWrite(10, HIGH); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10, LOW ); digitalWrite(11, HIGH); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, HIGH); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, HIGH);
    delay(5);
}
digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);

for(int w=0;w<90;w++){
    hombro.write(w); brazo.write(0);
    delay(10);
}

```

```

for(int w=0;w<90;w++){
    mano.write(w);
    delay(10);
}
delay(1000);

for(int w=90;w>0;w--){
    hombro.write(w);
    delay(10);
}
delay(1000);

for(int w=0;w<120;w++){
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, HIGH);
    delay(5);
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12,HIGH); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10,LOW ); digitalWrite(11, HIGH); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10, HIGH); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
}
digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);

for(int w=90;w>0;w--){
    mano.write(w);
    delay(10);
}
mano.write(0);
delay(1000);
}

void sec3(){

for(int w=0;w<120;w++){
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, HIGH);
    delay(5);
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12,HIGH); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10,LOW ); digitalWrite(11, HIGH); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10, HIGH); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);
    delay(5);
}
digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);

for(int w=0;w<90;w++){
    hombro.write(w); brazo.write(0);
    delay(10);
}

for(int w=0;w<90;w++){
    mano.write(w);
    delay(10);
}
delay(1000);

for(int w=90;w>0;w--){
    hombro.write(w);
    delay(10);
}
delay(1000);

```



```

for(int w=0;w<120;w++){
  digitalWrite(10, HIGH); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);
  delay(5);
  digitalWrite(10,LOW ); digitalWrite(11, HIGH); digitalWrite(12, LOW); digitalWrite(13, LOW);
  delay(5);
  digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12,HIGH); digitalWrite(13, LOW);
  delay(5);
  digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, HIGH);
  delay(5);
}
digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, LOW);

for(int w=90;w>0;w--){
  mano.write(w);
  delay(10);
}
mano.write(0);
delay(1000);
}

```

3.10. DESCRIPCIÓN DEL FUNCIONAMIENTO DEL ENLACE MATLAB ARDUINO

- **Conexión del cableado de los servomotores y motor paso a paso.-** La conexión del servomotor al Arduino se lo realizara como se lo indica a continuación en la siguiente tabla:

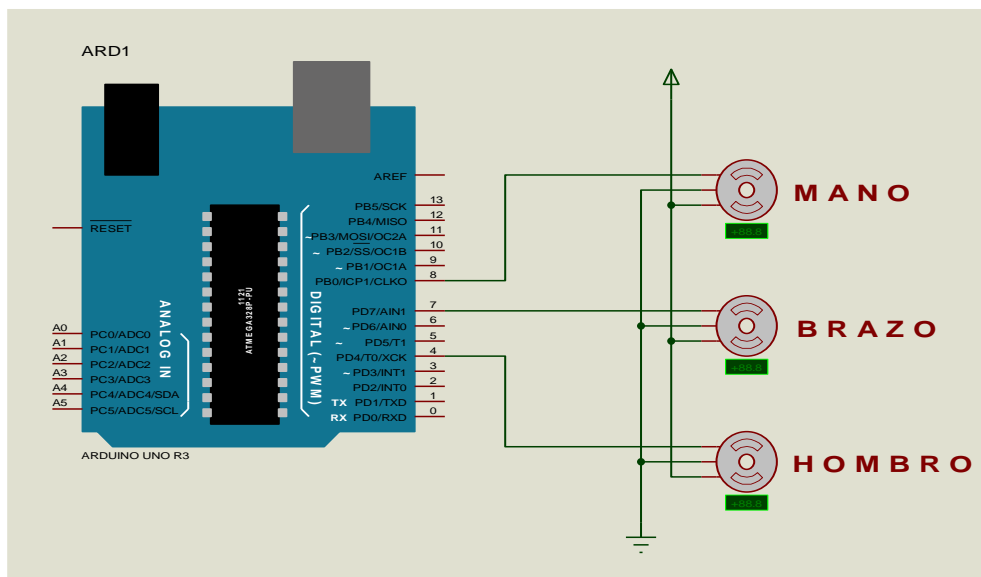


Figura 35 Conexiones Servomotores

Para la conexión del motor Paso a Paso es necesaria un elemento adicional el ULN2803, para realizar los giros según el requerimiento del usuario, la conexión se realiza como se indica en la siguiente tabla:

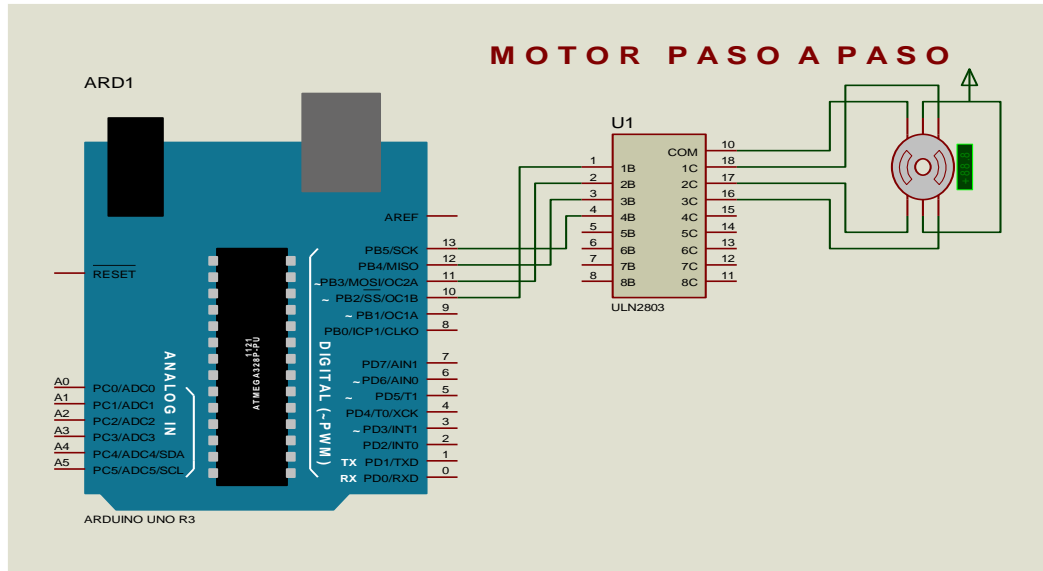


Figura 36 Conexiones Motor Paso a Paso

- Diagrama final de conexión de los elementos Utilizados

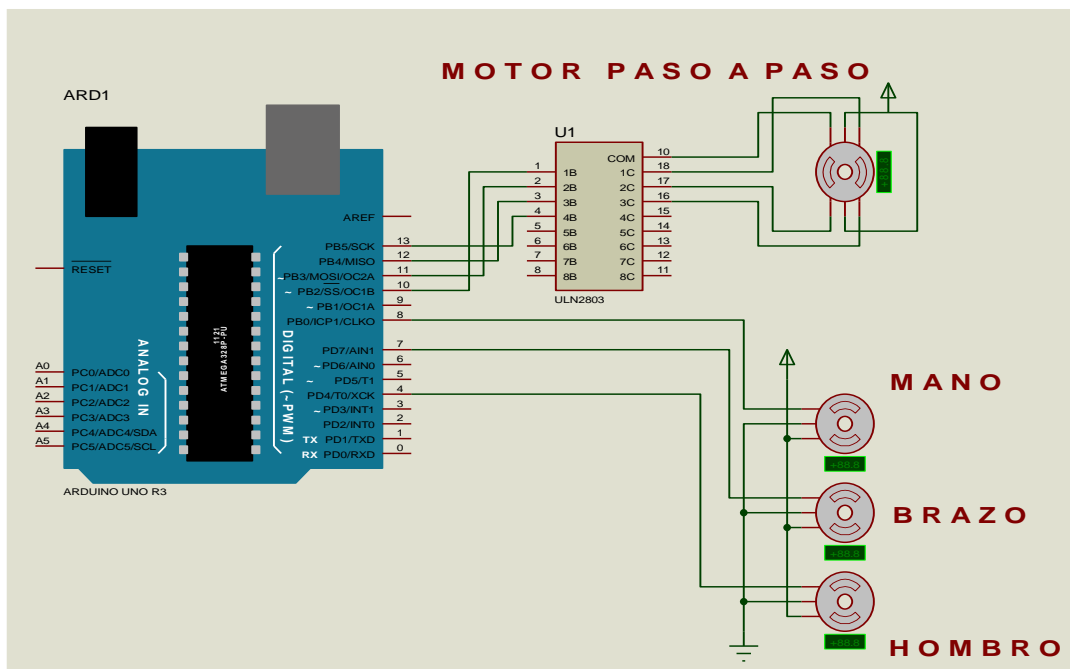


Figura 37 Conexión de elementos

Una vez realizadas las conexiones, proceder a abrir el Matlab con el programa del enlace, cargado en el Arduino.

Para su control sencillo utilizaremos una Interface gráfica. “GUI”.

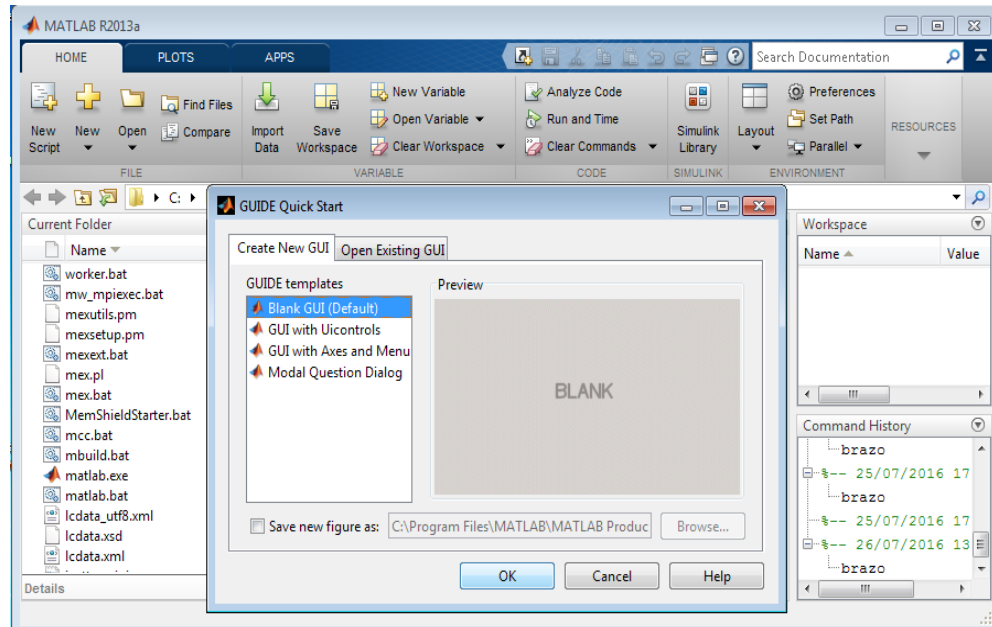


Figura 38 Programa de enlace

Utilizar cuatro “pushbutton” dentro de un “uipanel 1”, que controlará el brazo robótico según los deseos del usuario, como se muestra en la figura

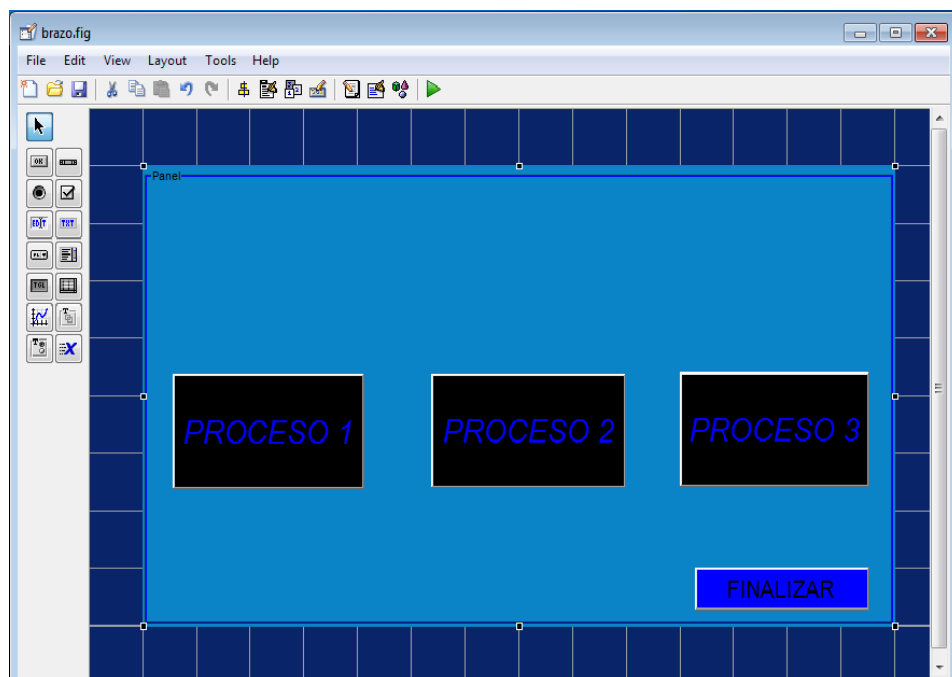


Figura 39 Programa de control

Al pulsar cada pushbutton se obtendrá los siguientes resultados como se muestra en la tabla a continuación.

Tabla 6

Funcionamiento del Servomotor y motor Paso a Paso

Nombre	Elemento a pulsar	Elemento a controlar	Actividad
PROCESO 1	Pushbutt 1	mano-brazo	Recoge objeto
PROCESO 2	Pushbutt 2	motor de paso- hombro-brazo-mano	Deja objeto en lado derecho
PROCESO 3	Pushbutt 3	motor de paso- hombro-brazo-mano	Deja objeto en lado izquierdo
FINALIZAR	Pushbutt 4	Final de secuencia	

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Una vez establecida las características físicas, estructura interna, componentes electrónicos y el funcionamiento de la tarjeta Arduino Uno, se alcanzó a establecer la compatibilidad entre Matlab y Arduino, siendo necesario la descarga desde la Web de un paquete propio para el enlace, el mismo que es proporcionado por Mathworks.
- Instalado los Softwares de sencilla programación, Matlab y Arduino, los mismos que son proporcionados por los fabricantes a través de la Web y accesibles al público en general sin algún costo, por tal motivo se les denomina de Código Abierto, se estableció el enlace procediendo al envío de las señales desde Matlab a los Servomotor y motor Paso a Paso.
- Realizadas las prácticas con el Servomotor, motor Paso a Paso, y sus respectivas guías de laboratorio, se comprobó su correcto funcionamiento, su aplicación en la industria y su sencilla manipulación.

4.2. RECOMENDACIONES

- El estudiante tendrá la opción de manipular diferentes elementos electrónicos que ampliara sus conocimientos en el control Automático de Procesos, a través del enlace Arduino-Matlab.
- Se recomienda utilizar la placa Arduino de cualquier versión que existe en el mercado, debido a que es muy útil para medios de aprendizaje por que cuenta con un código abierto facilitando así su accesibilidad, y se puede descargar desde la Web su programa, información y sus diferentes accesorios.
- El enlace se los realiza mediante paquetes de datos del software proporcionados por los administradores de Arduino.
- Para un mejor aprendizaje se recomienda variar los grados de libertad que forman los servomotores en el brazo robótico, así alcanzamos un mayor conocimiento en todos lo referente a prácticas de Instrumentación virtual.

GLOSARIO DE TÉRMINOS

CONTROL DE PROCESOS. Teoría básica de control automático, acciones de control, parámetros de sintonización en un controlador comercial, tipos de controlador, controladores neumáticos e hidráulicos, control de nivel de líquido, temperatura con control PID.

CONTROLADOR. Dispositivo con una salida que varía para regular una variable de control de una manera específica. Un controlador manual varía su salida automáticamente en respuesta a una entrada directa o indirecta de un proceso variable. Un controlador manual es una estación manual de carga y su salida no depende de una medida de un proceso variable pero puede variarse solamente por medio de un procedimiento manual.

El Set Point o punto de referencia puede ser establecido manualmente, automáticamente o programado. Su valor se expresa en las mismas unidades que la variable controlada.

DESARROLLO TECNOLÓGICO. Es el avance que ha tenido la tecnología a través de los años, de hecho se cree que evoluciono de manera acelerada en la revolución industrial con el inicio de la máquina de vapor.

DIGITAL. Término aplicado a una señal o dispositivo que usa dígitos.

FUNCIÓN. Propósito que debe cumplir un dispositivo de control.

INTERFAZ. Conexión e interacción entre hardware, software y el usuario.

INDUSTRIA. Es el conjunto de procesos y actividades que tienen como finalidad transformar las materias primas en productos elaborados, de forma masiva. Existen diferentes tipos de industrias, según sean los productos que fabrican.

INSTRUMENTACIÓN. Colección de instrumentos o sus aplicaciones con el fin de observar mediciones, control, o cualquier combinación de estos.

INSTRUMENTACIÓN INDUSTRIAL. Es el grupo de elementos que sirven para medir, controlar y registrar variables de un proceso con el fin de optimizar los recursos utilizados en este.

PROCESO. Es cualquier operación o secuencia de operaciones que involucren un cambio de energía, estado, composición, dimensión, u otras propiedades que pueden referirse a un dato.

PROYECTO. Es el conjunto de actividades coordinadas e interrelacionadas que buscan cumplir con ciertos objetivos específicos, este generalmente debe ser alcanzado en un periodo de tiempo previamente definido y respetando un presupuesto.

SISTEMAS DE CONTROL. El sistema de control de un sistema automatizado permite ejecutar el programa y lograr que el proceso realice su función definida.

SOFTWARE. Es conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.

SKETCH. Programas hechos con el software Arduino.

TIPOS DE DATOS. Se utilizan para determinar el tamaño de un elemento de datos y cómo deben interpretarse los datos.