

*Tiempo ni dinero, podrán impedir
llegar a la
Meta,
siempre y cuando no olvides el
camino...!*





ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA
UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

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 EDISON

DIRECTOR: ING. CAMPAÑA EMMA

LATACUNGA, 02 de septiembre del 2016

CAPITULO I

JUSTIFICACIÓN E IMPORTANCIA

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

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.

OBJETIVOS

General:

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

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.

ALCANCE

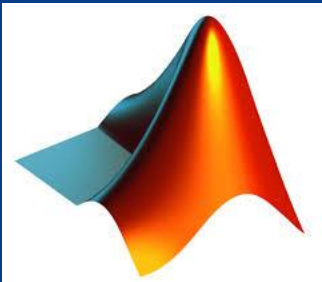
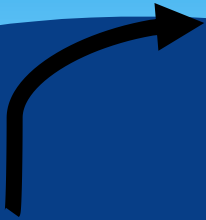
El control de los Servomotores y Motores paso a paso a través Arduino-Matlab, se pretende optimizar la enseñanza y aprendizaje de los estudiantes.

Y a todo el personal que hace uso del laboratorio de Instrumentación Virtual el cual permite la familiarización de los diferentes tipos de Arduino.

Adquisición y envío de datos a cualquier elemento o dispositivo a controlar.



CAPITULO III



The screenshot shows the Arduino website's download page. At the top left is the Arduino logo with "USA ONLY" below it. To its right is the Genuino logo with "OUTSIDE USA" below it. A search bar on the right says "Search the Arduino Website". A navigation menu includes "Home", "Buy", "Download", "Products", "Learning", "Forum", "Support", and "Blog". On the right side of the menu are "LOG IN" and "SIGN UP" links. Below the menu is a "DOWNLOAD" section with a language dropdown set to "ENGLISH". The main heading is "Download the Arduino Software". A large teal circle with the Arduino logo is on the left. To its right, the text reads: "ARDUINO 1.6.7. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions." On the right side of the page, there are links for "Windows Installer", "Windows ZIP file for non admin install", "Mac OS X 10.7 Lion or newer", "Linux 32 bits", and "Linux 64 bits". At the bottom right, there are links for "Release Notes", "Source Code", and "Checksums".

Software controlador de Arduino

```
BRAZO1 Arduino 1.6.9
Archivo Editar Programa Herramientas Ayuda

BRAZO1
String text_in="";
String ar[5];
int begin_idx=0,idx;

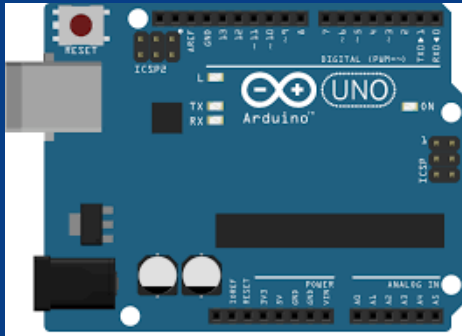
#include <Servo.h>

Servo hombro;
Servo brazo;
Servo mano;

//////////////////////////////////////VOID SETUP

Arduino/Genuino Uno en COM4
```

INSTALACIÓN DEL ARDUINO CON LA PC

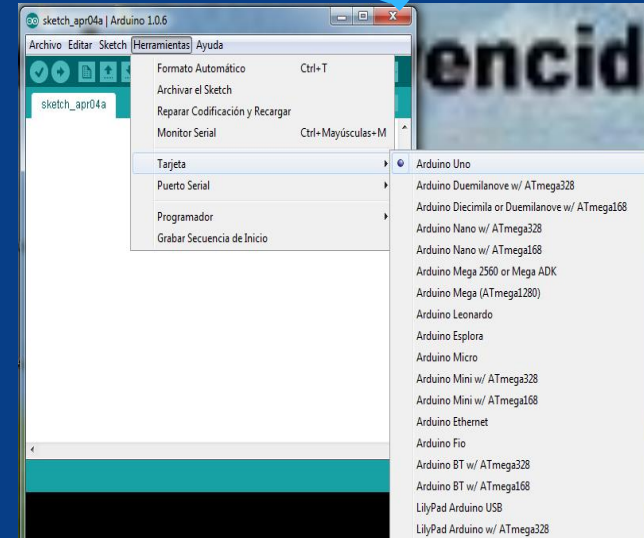
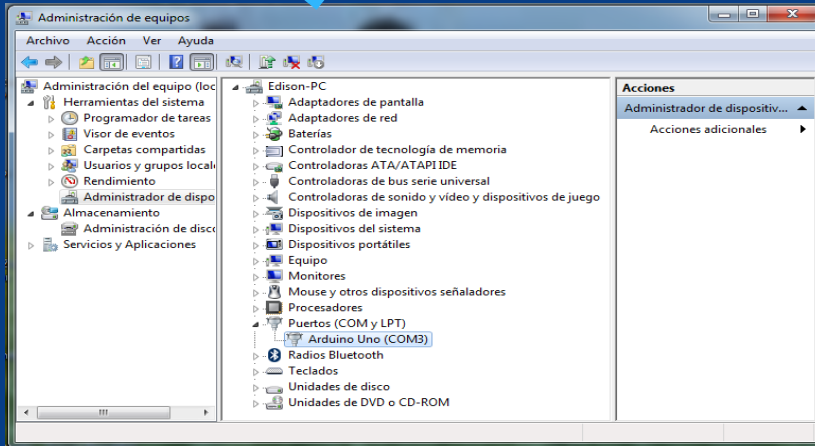
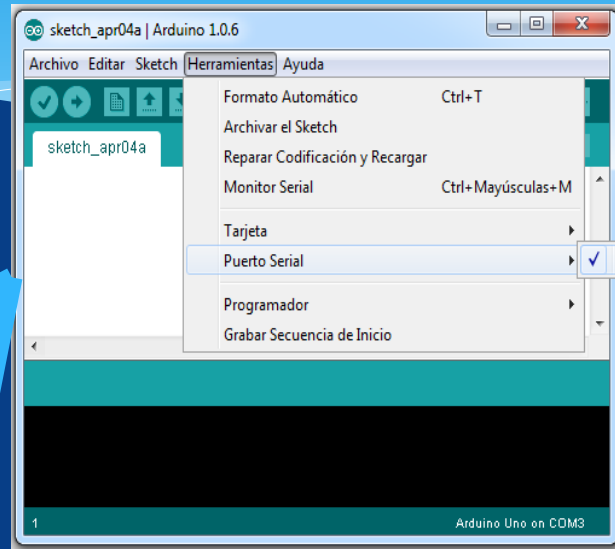


PAQUETE PARA EL ENLACE

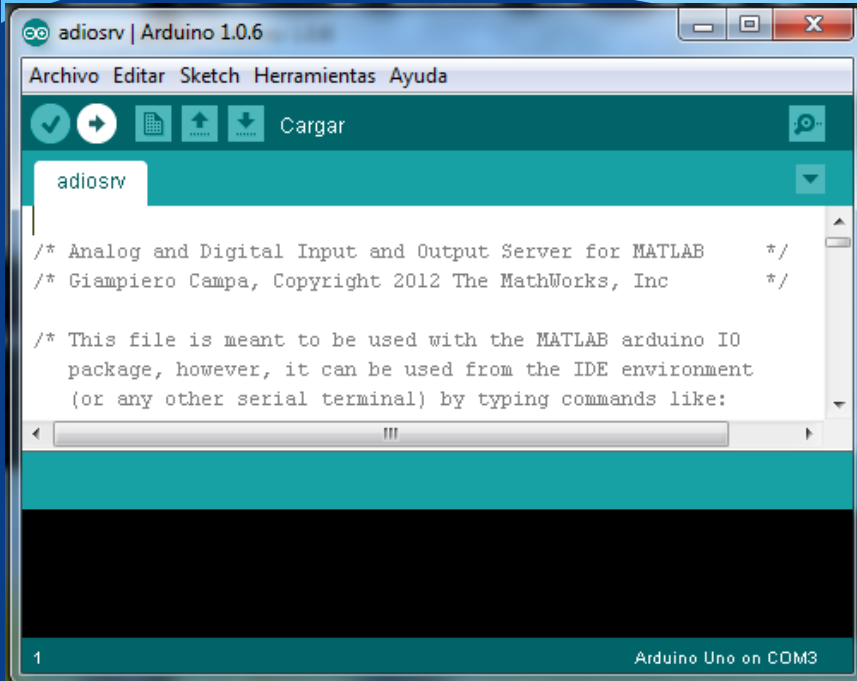
The screenshot shows the MathWorks File Exchange interface. At the top, there's a search bar and navigation links for Products, Solutions, Academia, Support, Community, Events, and Company. The main heading is 'File Exchange'. The featured item is 'Legacy MATLAB and Simulink Support for Arduino' by Giampiero Campa, updated on 30 Sep 2015. It has a 4.3 rating from 37 users and 1689 downloads. The file size is 170 KB, ID is #32374, and version is 1.14. A 'Download Zip' button is visible. The description notes that this solution is no longer supported and recommends using newer Arduino support packages. A list of highlights includes example files like 'example_encoders', 'example_io', 'example_mot', 'example_servos', and 'arduinoCleanup(customData...)'. A blue arrow points from this page down to the diagram below.



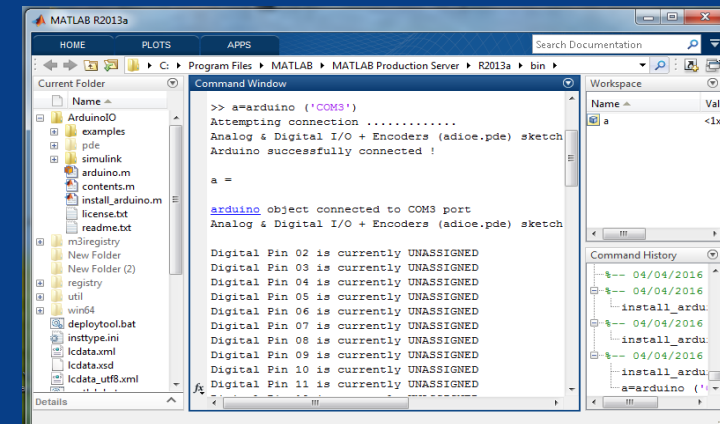
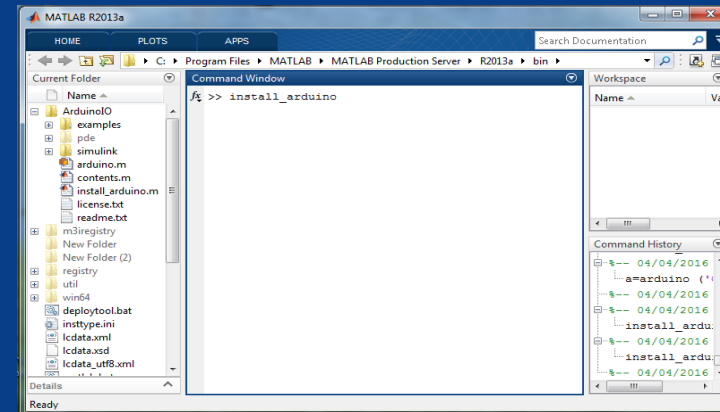
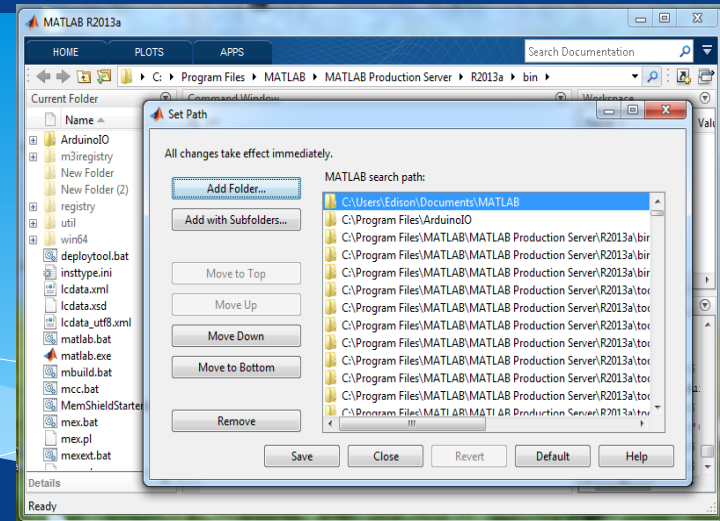
CONFIGURACIÓN-PLACA ARDUINO



Enlace Arduino - Matlab



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



Programa de Control en Arduino 1.0.6

```
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 secl(){
// 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<70;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);
```

```

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<70;w++){
    hombro.write(w); brazo.write(0);
    delay(10);
}

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

```

```

void sec3() {

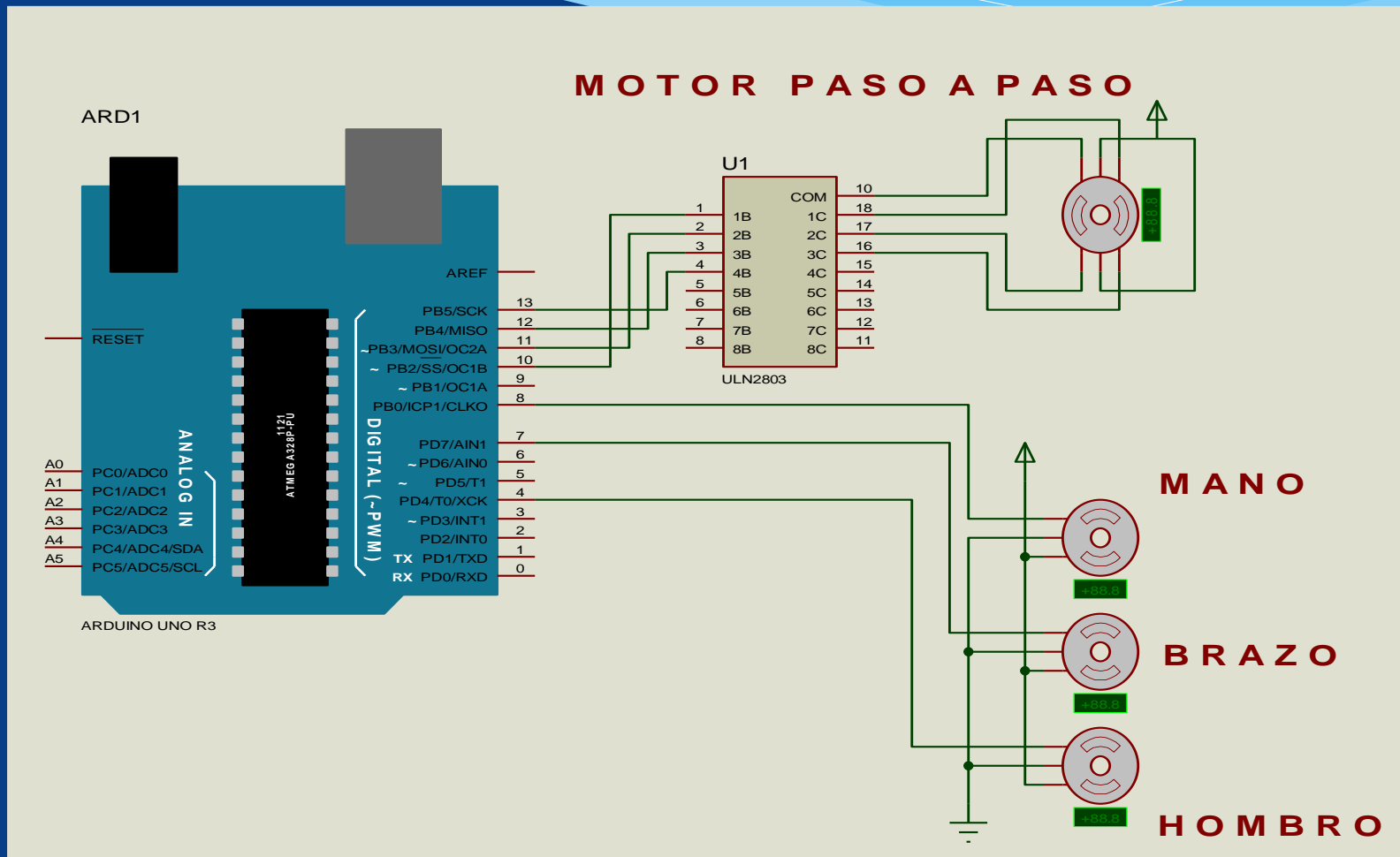
for(int w=0;w<120;w++){
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12, LOW); digitalWrite(13, HIGH);
    delay(5);
    digitalWrite(10, LOW); digitalWrite(11, LOW); digitalWrite(12,HIGH); digitalWrite(13, LOW);
    delay(5);
    digitalWrite(10,LOW ); digitalWrite(11, HIGH); digitalWrite(12, 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<70;w++){
    hombro.write(w); brazo.write(0);
    delay(10);
}

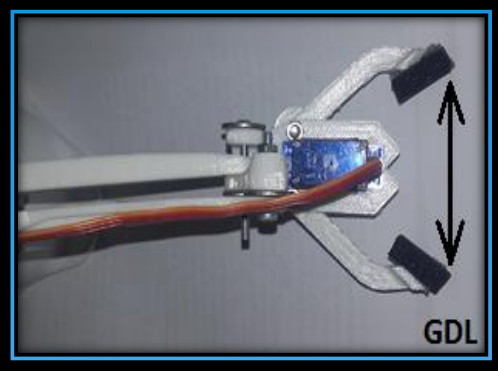
for(int w=0;w<90;w++){
    mano.write(w);
    delay(10);
}
}

```

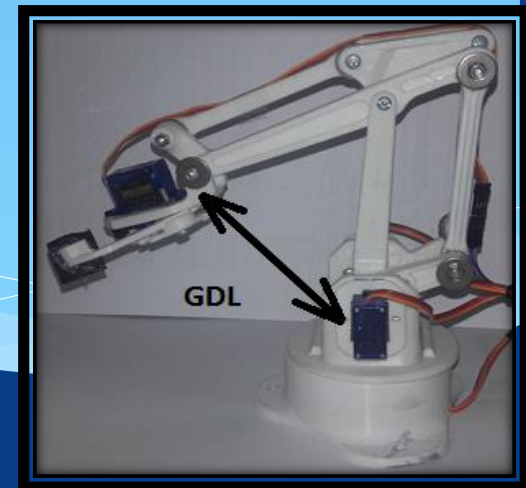
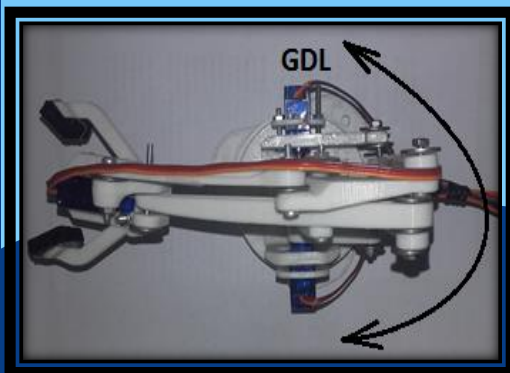
Conexión de los Elementos Utilizados



Estructura Mecánica



Mecánicamente, está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento consecutivo.

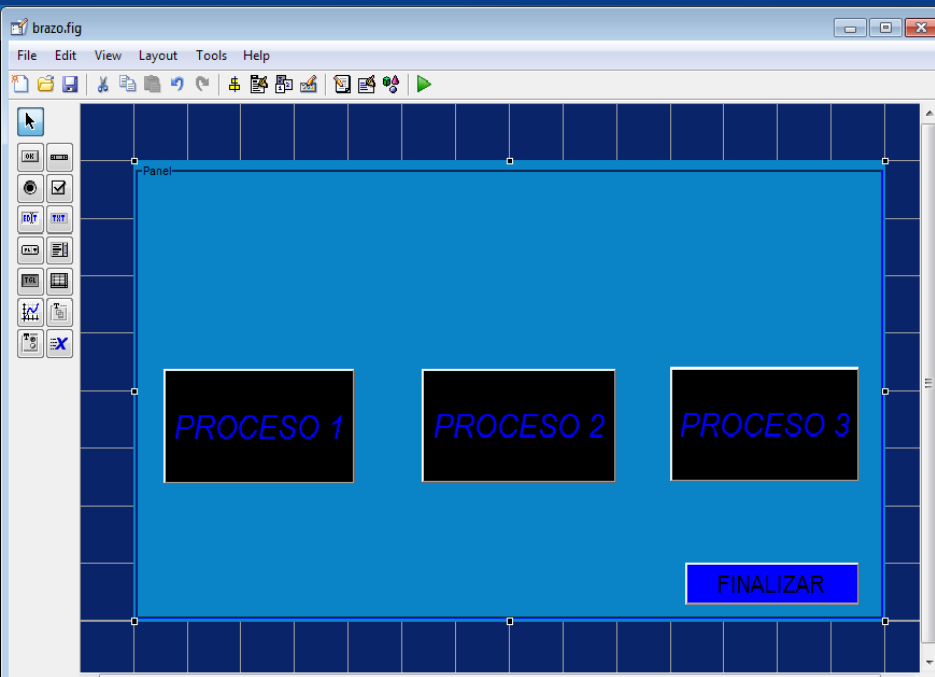


Configuración en Matlab

```
brazo.m x
81 % handles structure with handles and user data (see GUIDA1A)
82 %close all;clc;
83 com='COM3';
84 %Inicializo el puerto serial que utilizaré
85 delete(instrfind({'Port'},{com}));
86 puerto_serial=serial(com);
87 puerto_serial.BaudRate=9600;
88 warning('off','MATLAB:serial:fscanf:unsuccessfulRead');
89
90 %Abro el puerto serial
91 fopen(puerto_serial);
92 pause(2);
93 %unicodestr = native2unicode(i); % Convierte el entero i (0 a 255) a código ASCII
94 %fwrite(puerto_serial,dato,'uint8'); % se envia un dato de tipo entero sin signo de
95 %fwrite(puerto_serial,dato,'uint8'); % se envia un dato de tipo entero sin signo de
96 xyz='<PROCESO1'
97 fwrite(puerto_serial,xyz);
98 pause(2);
99 xyz='<PROCESO1'
100 %fprintf(puerto_serial,xyz);
101 fwrite(puerto_serial,xyz);
102 pause(2);
103
```

brazo / pushbutton1 Callback | Ln 78 Col 1

Funcionamiento



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	

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, se comprobó su correcto funcionamiento, su aplicación en la industria y su sencilla manipulación.

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.

GRACIAS

