28-11-2018



**INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

“Brazo robótico”

Programación Avanzada

**PROFESOR**: Maza Casas Lamberto

**INTEGRANTES**:

RODRÍGUEZ ALDAMA JOEL ISAÍ

GARCÍA MERCADO FAUSTINO

Tabla de contenido

[Resumen 2](#_Toc531976072)

[Objetivos 2](#_Toc531976073)

[Generales 2](#_Toc531976074)

[Específicos 2](#_Toc531976075)

[Antecedentes 2](#_Toc531976076)

[Antecedentes del proyecto 3](#_Toc531976077)

[Recursos Materiales 4](#_Toc531976078)

[Diseño de interfaz gráfica y programación en lenguaje C# 5](#_Toc531976079)

[Código de arduino 7](#_Toc531976080)

[Consideraciones prácticas 9](#_Toc531976081)

[Conclusión 9](#_Toc531976082)

# Resumen

# Objetivos

* Desarrollar un proyecto mecatrónico aplicando los conocimientos adquiridos a lo largo de las unidades académicas cursadas.
* Integrar los conocimientos de electrónica, mecánica y programación en un proyecto mecatrónico.

## Generales

* Aplicar los conocimientos obtenidos en la unidad de aprendizaje “Programación avanzada” en un proyecto mecatrónico.

## Específicos

* Diseñar un proyecto integral que maneje una interfaz gráfica que permita el movimiento de un brazo robótico.

## Antecedentes

La mecatrónica es la sinergia de las áreas de mecánica, electrónica e ingeniería de software que se conjuntan para crear productos y soluciones de vanguardia tecnológica. Dentro de la ingeniería mecatrónica debe existir un proceso administrativo que permita que el desarrollo tecnológico se de en las mejores condiciones y optimizando los recursos disponibles.

Este trabajo se realiza como una práctica que nos ayuda a los ingenieros en mecatrónica a planear y desarrollar proyectos de software aplicando conceptos de programación orientada a objetos, clases, objetos, Windows Forms, métodos, comunicación serial, etc.

Se decide realizar la construcción de un brazo robótico didáctico, debido a la gran cantidad de aplicaciones que este tiene dentro de la manufactura y la ingeniería mecatrónica. Siendo una exigencia de la industria y la investigación la formación de ingenieros mecatrónicos capaces de manipular y programar brazos robóticos. Para cumplir con estas demandas, decidimos construir un robot didáctico que nos ayude a simular a los robots industriales. Para esto no solo se realizó la programación de la interfaz gráfica y la comunicación con arduino, también la construcción del brazo y de los circuitos eléctricos de potencia con los que se mueve el robot.

## Antecedentes del proyecto

Un brazo robótico es un tipo de [brazo](https://es.wikipedia.org/wiki/Brazo) [mecánico](https://es.wikipedia.org/wiki/M%C3%A1quina), normalmente [programable](https://es.wikipedia.org/wiki/Programa_(m%C3%A1quina)), con funciones parecidas a las de un brazo humano; este puede ser la suma total del mecanismo o puede ser parte de un [robot](https://es.wikipedia.org/wiki/Robot) más complejo. Las partes de estos manipuladores o brazos son interconectadas a través de articulaciones que permiten tanto un movimiento rotacional (tales como los de un [robot articulado](https://es.wikipedia.org/wiki/Robot_articulado)), como un movimiento traslacional o desplazamiento lineal.

Los tipos de brazos robóticos son:

* [**Robot cartesiano**](https://es.wikipedia.org/wiki/Robot_cartesiano): Usado para trabajos de “pick and place” (tomar y colocar), aplicación de impermeabilizantes, operaciones de ensamblado, manipulación de máquinas herramientas y soldadura por arco. Es un robot cuyo brazo tiene tres articulaciones prismáticas, cuyos ejes son coincidentes con los ejes cartesianos.
* **Robot cilíndrico**: Empleado para operaciones de ensamblaje, manipulación de máquinas herramientas, soldadura por punto y manipulación en máquinas de fundición a presión. Es un robot cuyos ejes forman un sistema de coordenadas cilíndricas.
* **Robot esférico / Robot polar**, tal como el [Unimate](https://es.wikipedia.org/wiki/Unimate): Utilizado en la manipulación en máquinas herramientas, soldadura por punto, fundición a presión, máquinas de desbarbado, soldadura por gas y por arco. Es un robot cuyos ejes forman un sistema polar de coordenadas.
* [Robot SCARA](https://es.wikipedia.org/wiki/Robot_SCARA): Usado para trabajos de “pick and place” (tomar y colocar), aplicación de impermeabilizantes, operaciones de ensamblado y manipulación de máquinas herramientas. Es un robot que tiene dos articulaciones rotatorias paralelas para proporcionar elasticidad en un plano.
* [Robot articulado](https://es.wikipedia.org/wiki/Robot_articulado): Utilizado para operaciones de ensamblaje, fundición a presión, máquinas de desbarbado, soldadura a gas, soldadura por arco y pintado por spray. Es un robot cuyo brazo tiene como mínimo tres articulaciones rotatorias.
* [Robot paralelo](https://es.wikipedia.org/w/index.php?title=Robot_paralelo&action=edit&redlink=1): Uno de los usos es la plataforma móvil que manipula las cabinas de los simuladores de vuelo. Es un robot cuyos brazos tienen articulaciones prismáticas o rotatorias concurrentes.

En este caso se construyó un brazo mecánico robótico articulado con 7 servomotores, ejes rotatorios o grados de libertad, con los cuales se puede manipular objetos, tomarlos y transportarlos de un lado a otro, así como clasificar cosas.

Para la manufactura del robot se utilizaron tanto piezas de acrílico cortadas con un ROUTER CNC laser, como piezas de impresión 3D, impresas con material PLA.

El corte con láser es una técnica empleada para cortar piezas caracterizada en que su fuente de [energía](https://es.wikipedia.org/wiki/Energ%C3%ADa) es un láser que concentra luz en la superficie de trabajo.

La impresión 3D es un grupo de tecnologías de fabricación por adición donde un objeto tridimensional es creado mediante la superposición de capas sucesivas de material.

## Recursos Materiales

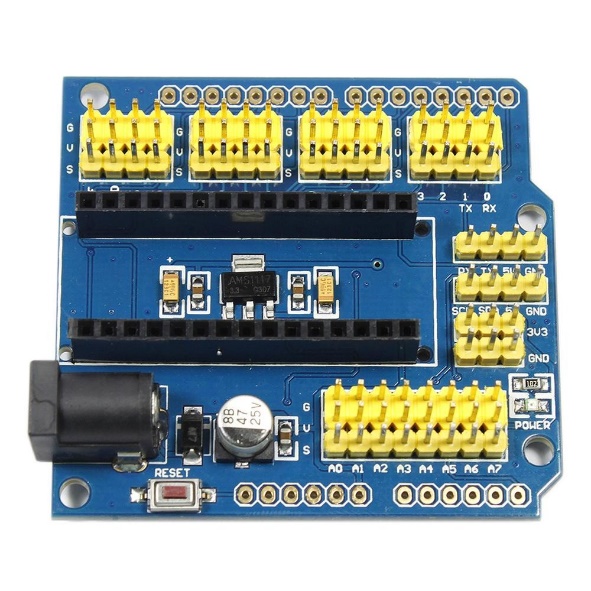
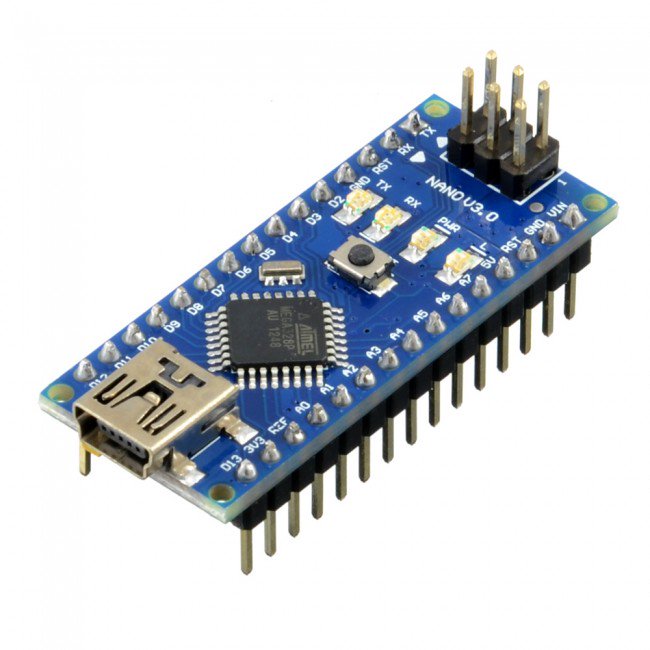


Figure 1.II Arduino Nano

Figura 1 Shield de potencia para Servo-motores Arduino

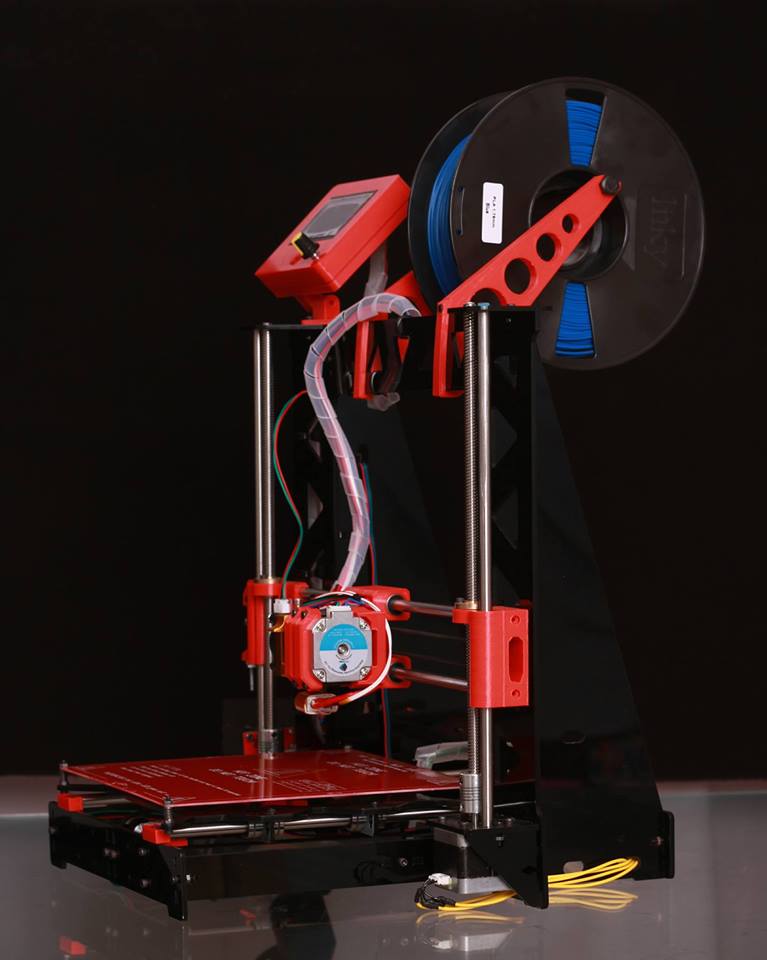


Figure 2 Impresora 3D

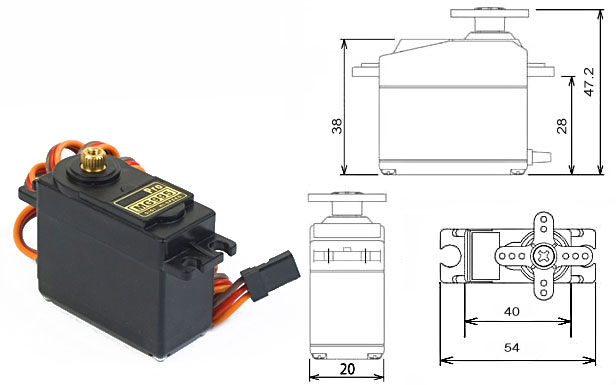


Figure 3 Servomotor 5 kg

Figure 4 Servomotor 5kg

## Diseño de interfaz gráfica y programación en lenguaje C#

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO.Ports;

namespace WindowsFormsApp2

{

public partial class Form1 : Form

{

SerialPort port;

public Form1()

{

InitializeComponent();

init();

}

private void init()

{

port = new SerialPort();

port.PortName = "COM3";

port.BaudRate = 9600;

try {

port.Open();

}

catch (Exception e1) {

MessageBox.Show(e1.Message);

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

if (port.IsOpen) {

port.WriteLine("a");

port.WriteLine(Val\_trackBar1.Value.ToString());

Degree\_label.Text = "Base = " + Val\_trackBar1.Value.ToString();

}

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

if (port.IsOpen)

{

port.WriteLine("d");

port.WriteLine(Val\_trackBar4.Value.ToString());

Degre\_label4.Text = "G3 = " + Val\_trackBar4.Value.ToString();

}

}

private void Degree\_label\_Click(object sender, EventArgs e)

{

}

private void trackBar1\_Scroll\_1(object sender, EventArgs e)

{

if (port.IsOpen)

{

port.WriteLine("b");

port.WriteLine(Val\_trackBar2.Value.ToString());

Degree\_label3.Text = "G1 = " + Val\_trackBar2.Value.ToString();

}

}

private void trackBar2\_Scroll(object sender, EventArgs e)

{

if (port.IsOpen)

{

port.WriteLine("c");

port.WriteLine(Val\_trackBar3.Value.ToString());

Degree\_label4.Text = "G2 = " + Val\_trackBar3.Value.ToString();

}

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label3\_Click\_1(object sender, EventArgs e)

{

}

private void trackBar1\_Scroll\_2(object sender, EventArgs e)

{

if (port.IsOpen)

{

port.WriteLine("e");

port.WriteLine(Val\_trackBar6.Value.ToString());

Degree\_label5.Text = "Gripper = " + Val\_trackBar6.Value.ToString();

}

}

}

}

## Código de arduino

ARDUINO#include <Servo.h>//Biblioteca usada

Servo G1,G2,G3,G4,G5,G6;

char a;

int val;

void setup()

{

Serial.begin(9600);

G1.attach(2);//Base

G2.attach(4);//G1

G3.attach(5);//G1

G4.attach(6);//G2

G5.attach(11);//G3

G6.attach(12);//Gripper

}

void loop(){

a=Serial.read();

if (a=='a')

{

val=Serial.parseInt();

G1.write(val);

}

a=Serial.read();

if (a=='b')

{

val=Serial.parseInt();

G2.write(val);

G3.write(val);

}

a=Serial.read();

if (a=='c')

{

val=Serial.parseInt();

G4.write(val);

}

a=Serial.read();

if (a=='d')

{

val=Serial.parseInt();

G5.write(val);

}

a=Serial.read();

if (a=='e')

{

val=Serial.parseInt();

G6.write(val);

}

}

Ilustración 1Interfaz gráfica del brazo

# Consideraciones prácticas

Durante el proyecto se aplicaron conceptos vistos en Programación Avanzada.

Principalmente utilizamos lo referente a comunicación serial, dentro del lenguaje C#, esto se logró utilizando l el espacio de nombres using System.IO.Ports; el cual contiene clases para controlar el Puerto serial.

Utilizamos el concepto de clases dentro de esta libreria, declarando un nuevo objeto dentro de la clase [SerialPort](https://docs.microsoft.com/es-es/dotnet/api/system.io.ports.serialport?view=netframework-4.7.2), la cual proporciona un marco de trabajo para la E/S sincrónica y orientada a eventos, el ecceso a los estados de punto de conexión e interrupción y el acceso a las propiedades del controlador serie. Se puede usar para ajustar objetos Steam, lo que permite que las clases que utilizan secuencias tengan acceso al puerto serie.

SerialPort port;

port = new SerialPort();

las lineas anteriores son un ejemplo de como se declaró un objeto de la clase SerialPort() con el cual se estableció la comunicación serial con Arduino. Para Arduino se utilzan los métodos

Serial.begin(9600);

val=Serial.parseInt();

La primera linea establece la velocidad a la que se establece la comunicación y la segunda línea extraé datos del Puerto para guardarlos en un entero que después es pasado al servomotor.

Se construyó una interfaz gráfica con una track bar, de esta extraímos un dato del evento de deslizarla para después enviar este dato al Puerto serial y después a Arduino. Todo esto se realizó con ayuda de las herramientas ofrecidas por C# en Visual Studio, a continuación un ejemplo:

private void trackBar1\_Scroll(object sender, EventArgs e)

{

if (port.IsOpen) {

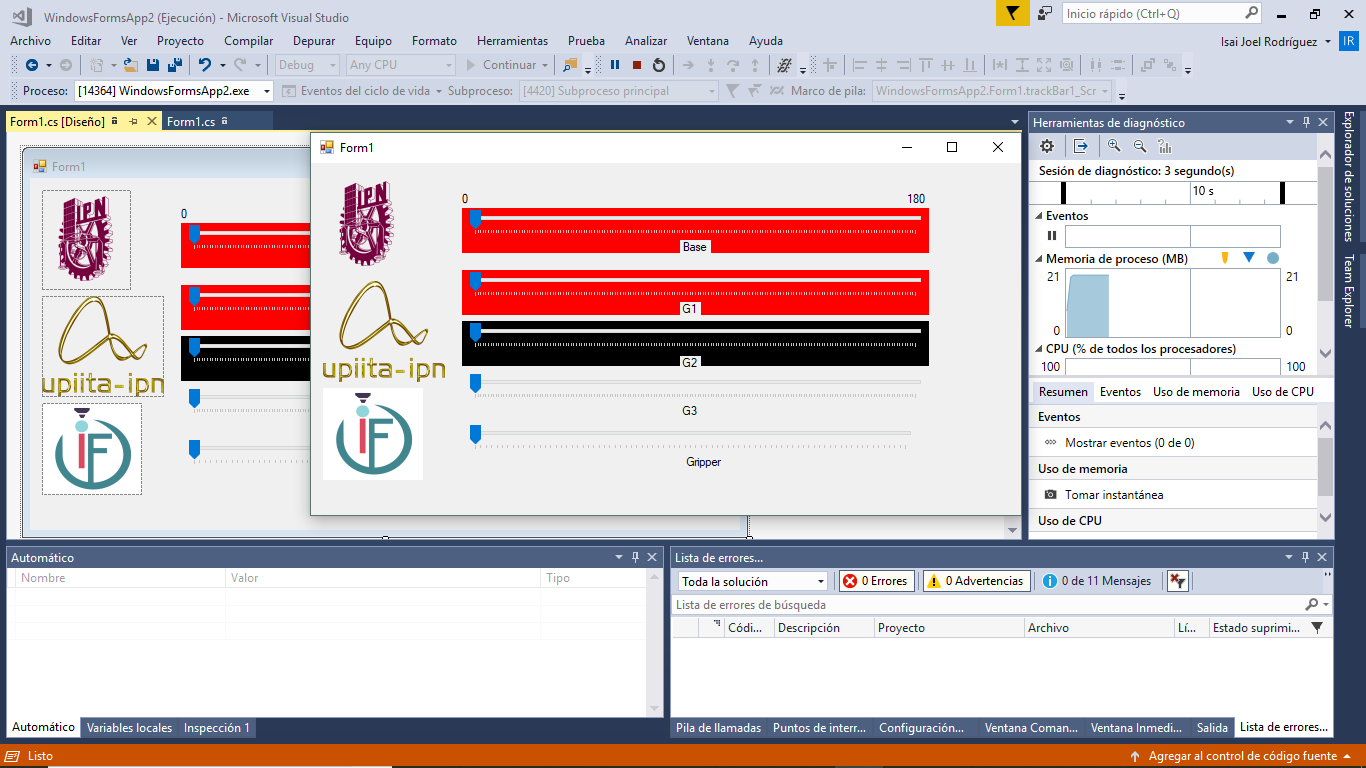
port.WriteLine("a");

port.WriteLine(Val\_trackBar1.Value.ToString());

Degree\_label.Text = "Base = " + Val\_trackBar1.Value.ToString();

}

}



# Conclusión

Al final logramos integrar los conocimientos adquiridos en la materia y aplicarlos a un proyecto mecatrónico en el por medio de software y un microcontrolador nos permitió controlar un circuito electrónico que a su vez controlaba un dispositivo mecánico

Logramos aterrizar los conocimientos aprendidos en clase para aplicarlos a la solución de un problema. Comenzando la programación desde cero y construyendo un proyecto de nuestra elección.