

# Trabajo Fin de Grado

## Grado en Ingeniería en Tecnologías Industriales

### Integración de robot manipulador con posicionador basado en arduino

Autor: Antonio Pérez García

Tutor: Luis Fernando Castaño Castaño

**Dpto. Ingeniería de Sistemas y Automática**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2022





Trabajo Fin de Grado  
Grado en Ingeniería en Tecnologías Industriales

# **Integración de robot manipulador con posicionador basado en arduino**

Autor:

Antonio Pérez García

Tutor:

Luis Fernando Castaño Castaño

Profesor Contratado Doctor

Dpto. Ingeniería de Sistemas y Automática  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Grado: Integración de robot manipulador con posicionador basado en arduino

Autor: Antonio Pérez García

Tutor: Luis Fernando Castaño Castaño

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

El trabajo de fin de grado es la culminación de mucho esfuerzo realizado durante años de mis padres Antonio y Rosario. Agradezco su paciencia y apoyo durante todo este tiempo.

*Antonio Pérez García*

*Sevilla, 2022*





# Resumen

---

**E**n el laboratorio de automática



# Abstract

---

In the lab of automatics



# Índice Abreviado

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<b>1 Introducción</b>	<b>1</b>
1.1 Modos de funcionamiento	1
<b>2 Descripción del hardware</b>	<b>5</b>
2.1 Arduino Mega 2560	5
2.2 Ethernet Shield de Arduino	7
2.3 Motor DC y Driver L298N	8
2.4 Encoder Rotativo Incremental	8
2.5 Calibrador digital	9
2.6 LCD	9
2.7 Sensor fotoeléctrico OMRON	9
2.8 Banda transportadora con elementos integrados	9
2.9 Controlador IRC5C	9
<b>3 Desarrollo de placa de conexiones</b>	<b>11</b>
<b>4 Planificación de caja</b>	<b>13</b>
<b>5 Desarrollo en Arduino</b>	<b>15</b>
<b>6 Desarrollo en Robotstudio</b>	<b>17</b>
<b>7 Resultados</b>	<b>19</b>

<b>8 Conclusiones</b>	<b>21</b>
<i>Índice de Figuras</i>	23
<i>Índice de Tablas</i>	25
<i>Índice de Códigos</i>	27
<i>Bibliografía</i>	29

# Índice

---

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
<b>1 Introducción</b>	<b>1</b>
1.1 Modos de funcionamiento	1
1.1.1 Modo local con microcontrolador	2
1.1.2 Modo remoto con microcontrolador	2
1.1.3 Modo local sin microcontrolador	2
1.1.4 Modo remoto sin microcontrolador	2
<b>2 Descripción del hardware</b>	<b>5</b>
2.1 Arduino Mega 2560	5
2.1.1 Alimentación	6
2.1.2 Memoria	6
2.1.3 Entradas y Salidas	6
2.1.4 Comunicación	7
2.2 Ethernet Shield de Arduino	7
2.3 Motor DC y Driver L298N	8
2.3.1 Conexión del módulo L298N	8
2.4 Encoder Rotativo Incremental	8
2.4.1 Especificaciones técnicas	8
2.4.2 Conexión del Encoder	8
2.5 Calibrador digital	9
2.6 LCD	9
2.7 Sensor fotoeléctrico OMRON	9
2.8 Banda transportadora con elementos integrados	9
2.9 Controlador IRC5C	9
<b>3 Desarrollo de placa de conexiones</b>	<b>11</b>
<b>4 Planificación de caja</b>	<b>13</b>
<b>5 Desarrollo en Arduino</b>	<b>15</b>

<b>6 Desarrollo en Robotstudio</b>	<b>17</b>
<b>7 Resultados</b>	<b>19</b>
<b>8 Conclusiones</b>	<b>21</b>
<i>Índice de Figuras</i>	23
<i>Índice de Tablas</i>	25
<i>Índice de Códigos</i>	27
<i>Bibliografía</i>	29



# Notación

---

ABB	Asea Brown Boveri
mA	Miliamperio
V	Voltio
$\mathbb{C}$	Cuerpo de los números complejos
$\ \mathbf{v}\ $	Norma del vector $\mathbf{v}$
$\langle \mathbf{v}, \mathbf{w} \rangle$	Producto escalar de los vectores $\mathbf{v}$ y $\mathbf{w}$
$ \mathbf{A} $	Determinante de la matriz cuadrada $\mathbf{A}$
$\det(\mathbf{A})$	Determinante de la matriz (cuadrada) $\mathbf{A}$
$\mathbf{A}^\top$	Transpuesto de $\mathbf{A}$
$\mathbf{A}^{-1}$	Inversa de la matriz $\mathbf{A}$
$\mathbf{A}^\dagger$	Matriz pseudoinversa de la matriz $\mathbf{A}$
$\mathbf{A}^H$	Transpuesto y conjugado de $\mathbf{A}$
$\mathbf{A}^*$	Conjugado
c.t.p.	En casi todos los puntos
c.q.d.	Como queríamos demostrar
■	Como queríamos demostrar
□	Fin de la solución
e.o.c.	En cualquier otro caso
$e$	número $e$
$e^{jx}$	Exponencial compleja
$e^{j2\pi x}$	Exponencial compleja con $2\pi$
$e^{-jx}$	Exponencial compleja negativa
$e^{-j2\pi x}$	Exponencial compleja negativa con $2\pi$
$\text{Re}$	Parte real
$\text{Im}$	Parte imaginaria
sen	Función seno
tg	Función tangente
arctg	Función arco tangente
$\sin^y x$	Función seno de $x$ elevado a $y$
$\cos^y x$	Función coseno de $x$ elevado a $y$
Sa	Función sampling
sgn	Función signo
rect	Función rectángulo
Sinc	Función sinc
$\frac{\partial y}{\partial x}$	Derivada parcial de $y$ respecto a $x$

$x^\circ$	Notación de grado, $x$ grados.
$\Pr(A)$	Probabilidad del suceso $A$
$E[X]$	Valor esperado de la variable aleatoria $X$
$\sigma_X^2$	Varianza de la variable aleatoria $X$
$\sim f_X(x)$	Distribuido siguiendo la función densidad de probabilidad $f_X(x)$
$\mathcal{N}(m_X, \sigma_X^2)$	Distribución gaussiana para la variable aleatoria $X$ , de media $m_X$ y varianza $\sigma_X^2$
$\mathbf{I}_n$	Matriz identidad de dimensión $n$
$\text{diag}(\mathbf{x})$	Matriz diagonal a partir del vector $\mathbf{x}$
$\text{diag}(\mathbf{A})$	Vector diagonal de la matriz $\mathbf{A}$
SNR	Signal-to-noise ratio
MSE	Minimum square error
:	Tal que
$\stackrel{\text{def}}{=}$	Igual por definición
$\ \mathbf{x}\ $	Norma-2 del vector $\mathbf{x}$
$ \mathbf{A} $	Cardinal, número de elementos del conjunto $\mathbf{A}$
$\mathbf{x}_i, i = 1, 2, \dots, n$	Elementos $i$ , de 1 a $n$ , del vector $\mathbf{x}$
$dx$	Diferencial de $x$
$\leq$	Menor o igual
$\geq$	Mayor o igual
$\backslash$	Backslash
$\Leftrightarrow$	Si y sólo si
$x = a + 3 \underset{\substack{\uparrow \\ a=1}}{=} 4$	Igual con explicación
$\frac{a}{b}$	Fracción con estilo pequeño, $a/b$
$\Delta$	Incremento
$b \cdot 10^a$	Formato científico
$\xrightarrow{x}$	Tiende, con $x$
$\mathbf{O}$	Orden
$\text{TM}$	Trade Mark
$\mathbb{E}[x]$	Esperanza matemática de $x$
$\mathbf{C}_x$	Matriz de covarianza de $\mathbf{x}$
$\mathbf{R}_x$	Matriz de correlación de $\mathbf{x}$
$\sigma_x^2$	Varianza de $x$

# 1 Introducción

---

En las prácticas de laboratorio realizadas por parte de alumnos durante la docencia de los cursos de Robótica del Departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería de Sevilla, los alumnos deben programar el movimiento de un brazo robótico presente en dicho laboratorio. El objetivo es que una pieza sea trasladada por el robot desde un punto de la mesa de trabajo hacia una segunda posición. El proceso se realiza colocando manualmente la pieza, con los problemas que ello conlleva. Por un lado, la precisión es cuestionable, ya que el propio alumno no tiene una referencia sobre la cual poder repetir el proceso de forma eficaz y el error introducido al sistema es alto. Por otro lado, al invadir el espacio de trabajo del brazo robótico continuamente se producen riesgos innecesarios impropios de las normas de seguridad en la industria.

Este trabajo es la continuación de [1] y [2], que sentaron las bases del proyecto. En esta ocasión, el enfoque es en la implementación sobre los equipos del laboratorio. Por ello, el sistema creado debe quedar en una caja donde se realicen las conexiones electrónicas y todos los dispositivos. Además el sistema debe tener componentes fácilmente sustituibles para facilitar las reparaciones.

El sistema cuenta con las siguientes características:

- Posicionamiento de piezas en medidas de ejes X e Y que el usuario requiera para interactuar con el robot.
- Conexión entre Arduino y RobotStudio mediante protocolo TCP/IP para comunicaciones.
- Funcionamiento sin Arduino mediante señales digitales del robot.
- Funcionamiento sin conexión directa entre RobotStudio y Arduino.

## 1.1 Modos de funcionamiento

Todas las funciones especificadas no pueden cumplirse al mismo tiempo ya que existirían conflictos entre las mismas, por lo que el sistema debe tener ciertos modos de funcionamiento en los que se activen o desactiven dichas características.

La primera consideración es determinar el dispositivo que gobierna el sistema o *máster*. Por ello, se puede diferenciar cuando el *máster* es la controladora del robot (o RobotStudio durante una simulación) o el sistema caja (Arduino o la propia electrónica interna). Respectivamente serán los modos remoto y local.

Por otro lado, como el microcontrolador presente en el Arduino puede estar funcionando o no, se deben añadir las dos posibilidades. Se tiene el modo con microcontrolador y sin microcontrolador.

En total, se cuenta con cuatro modos de funcionamiento que se describen a continuación.

#### **1.1.1    Modo local con microcontrolador**

Las órdenes del sistema están proporcionadas por los periféricos de entrada presentes en la caja y el microcontrolador es el encargado de gestionar el posicionamiento y mover el motor cuando le sea indicado.

En caso de que esté disponible la conexión con la controladora del robot, el Arduino comunica mediante conexión TCP/IP la posición de la pieza en los ejes X e Y además del estado del sensor fotoeléctrico y del sistema.

#### **1.1.2    Modo remoto con microcontrolador**

El gobierno del sistema pasa a ser parte de la controladora del robot, convirtiendo al microcontrolador en esclavo. El microcontrolador sigue encargándose del posicionamiento y movimiento del motor, pero las órdenes pasan a ser recibidas mediante conexión TCP/IP.

Como en el caso anterior, se envía la posición, estado del sistema y del sensor fotoeléctrico mediante conexión TCP/IP a la controladora.

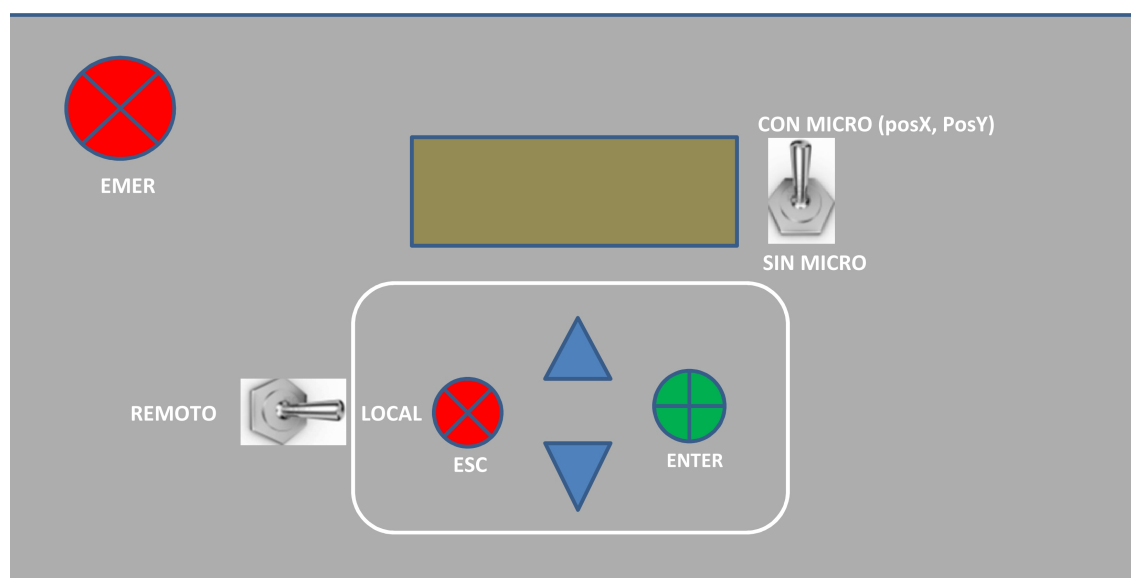
#### **1.1.3    Modo local sin microcontrolador**

Al no tener el microcontrolador operativo, el posicionamiento deja de funcionar y las funciones del sistema pasan a ser más básicas. El movimiento del motor queda a cargo de la electrónica interna del sistema. Para interactuar con el motor y moverlo se realizará mediante los pulsadores de avance y retroceso.

El robot queda no comunicado y solo recibe la señal digital del sensor fotoeléctrico.

#### **1.1.4    Modo remoto sin microcontrolador**

Es un modo similar al anterior, en la cual la interacción con el sistema se produce mediante señales digitales enviadas por el robot.



**Figura 1.1** Estructura de un aplicación que use la plataforma.



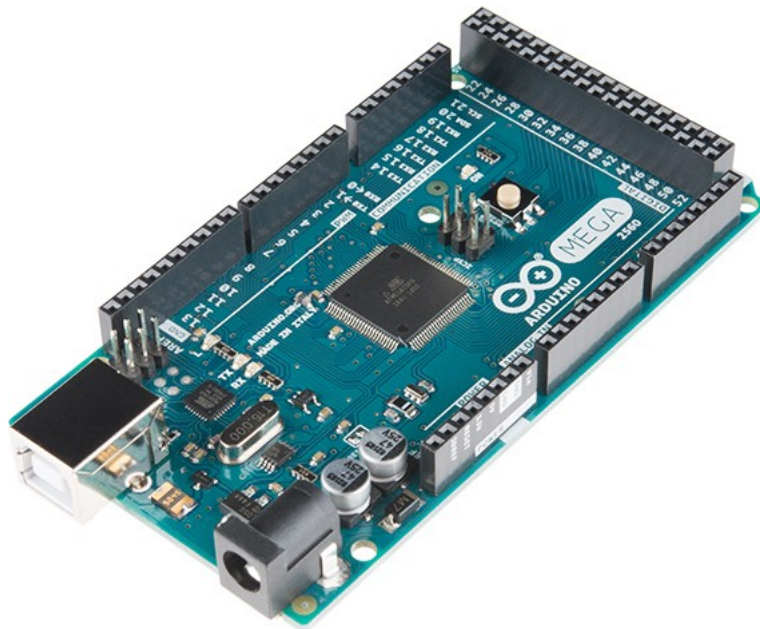
## 2 Descripción del hardware

---

Una vez conocidas las funciones que debe desempeñar el sistema es importante definir el hardware a usar para que sea capaz de cumplimentar los requerimientos establecidos. Los principales dispositivos utilizados en el proyecto son los siguientes:

### 2.1 Arduino Mega 2560

Se trata de una placa de desarrollo que cuenta con el microcontrolador ATmega2560 y todo lo necesario para prototipar un sistema. Cuenta con 54 pines de entrada y salida (GPIO), de los cuales 15 pueden utilizarse como salida PWM, 4 puertos UART, oscilador de 16 MHz, conexión USB tipo B, pines ICSP de programación y botón de reinicio.



**Figura 2.1** Arduino Mega 2560.

**Tabla 2.1** Características Arduino Mega 2560.

Microcontrolador	ATmega2560
Tensión de funcionamiento	5 V
Voltaje de entrada recomendado	7-12 V
Voltaje de entrada límite	6-20 V
Pines de E/S digitales	54 (de los cuales 15 con salida PWM)
Pines de entrada analógica	16
Corriente CC por pin	20 mA
Corriente CC para pin 3.3 V	50 mA
Memoria Flash	256 kB, 8 kB utilizados por el gestor de arranque
SRAM	8 kB
EEPROM	4 kB
Frecuencia de reloj	16 MHz
Longitud	101.52 mm
Anchura	53.3 mm
Peso	37 g

### 2.1.1 Alimentación

El Arduino Mega 2560 puede ser alimentado mediante la conexión USB tipo B proporcionada por un ordenador o por una fuente de alimentación externa. Además, puede ser alimentado por los pines de alimentación de la placa, que se describen a continuación:

- VIN. Entrada de tensión de la placa cuando hay alimentación externa.
- 5V. Pin que produce 5V regulados.
- 3V3. Pin que produce 3.3V regulados con un consumo máximo de 50 mA.
- GND. Pin de conexión a masa.
- IOREF. Referencia de tensión de trabajo del microcontrolador.

### 2.1.2 Memoria

El ATmega2560 cuenta con 256 kB de memoria flash de los cuales 8 kB se usan para el gestor de arranque, 8 kB de memoria SRAM y 4 kB de EEPROM.

### 2.1.3 Entradas y Salidas

El Arduino Mega cuenta con 54 pines que pueden ser utilizados como salida o entrada digital mediante las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Los niveles lógicos son de 5V. Cada pin soporta una corriente de 20 mA y cuenta con una resistencia de pull-up de entre 20 y 50 kΩ. Además, algunos pines tienen funciones especiales:

- Comunicación serie. Se usa para recibir y transmitir datos en serie TTL. Son las parejas 0 y 1, 14 y 15, 16 y 17 y 18 y 19. Los pines 0 y 1 son los correspondientes a la comunicación con el conversor USB-TTL integrado en la placa
- Interrupciones externas. Se utilizan para activar interrupciones en el software. Son los pines 2, 3, 18, 19, 20 y 21.



- Salida PWM. Proporcionan una salida modulada con 8 bits de precisión con la función `analogWrite()`. Son los pines del 2 al 13 y del 44 al 46.
- Comunicación SPI. Pines 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Permiten comunicación SPI con otros dispositivos utilizando la biblioteca SPI.
- LED integrado. En el pin 13 hay un LED integrado que puede encenderse con un valor HIGH y apagarse con un valor LOW.
- Comunicación TWI. Pines 20 (SDA) y 21 (SCL). Permite comunicación I2C/TWI.

#### 2.1.4 Comunicación

La placa Arduino Mega cuenta con diversas formas para comunicarse con un ordenador o con otros dispositivos. El ATmega2560 cuenta con cuatro UART de hardware para comunicación TTL a 5V, una conexión SPI y una conexión I2C. Mediante uno de los puertos UART el ATmega2560 se comunica con un ATmega16U2 que canaliza la conexión mediante USB a un ordenador y permite la comunicación entre ambos. Esto permite tanto programar la placa mediante el IDE de Arduino como recibir mediante monitor serie datos simples.

## 2.2 Ethernet Shield de Arduino

El Ethernet Shield de Arduino se trata de una placa que añade la funcionalidad de conectar una placa de Arduino a una red mediante conexión Ethernet. Se trata de una placa basada en el chip Wiznet W5100 que provee de una pila de red IP capaz de soportar protocolos TCP y UDP. Usa la librería Ethernet para leer y escribir flujos de datos.

La placa cuenta con varios LED que proporcionan información:

- ON. Indica que la placa está alimentada.
- LINK. Indica presencia de enlace de red.
- 100M. Indica la existencia de conexión de red de 100 Mb/s.
- RX. Indica que se reciben datos cuando parpadea.
- TX. Indica que se transmiten datos cuando parpadea.

Unos puntos importantes del Ethernet Shield son:

- Funciona a 5V.
- Tiene un microcontrolador W5100 con 16k de buffer y es independiente de la memoria del ATmega2560.
- Se comunica con el ATmega2560 mediante SPI.
- Soporta 4 conexiones simultáneas.
- Utiliza la librería Ethernet.
- Dispone de lector de tarjetas microSD para guardar ficheros.
- Utiliza los pines 10, 11, 12 y 13 para comunicarse con el W5100 mediante SPI.

## **2.3 Motor DC y Driver L298N**

El principal actuador de este proyecto es un motor de corriente continua que es el que mueve la cinta transportadora, permitiéndola avanzar, retroceder o pararse. Este motor funciona a 24V y tiene infinidad de uso, ya que son sencillos de operar y tienen un bajo coste.

Para controlar dicho motor se utiliza un Driver L298N, que está basada en un puente H. Dicho módulo es capaz de controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2A.

La principal funcionalidad del módulo es separar la parte asociada a la potencia (que funciona a 24V en este caso) de la parte asociada al control (que funciona a 5V).

### **2.3.1 Conexionado del módulo L298N**

El módulo cuenta con un circuito integrado LM7805 que proporciona 5V que puede ser activado o desactivado mediante un jumper.

Mientras el jumper esté activo, la placa admite tensiones de alimentación de entre 6 y 12V. Así, la conexión de 5V será una salida.

Por otra parte, si el jumper está desactivado la placa admite tensiones de alimentación de entre 12 y 35V. En este caso se deberá proporcionar 5V de referencia a la placa para el funcionamiento de la parte lógica.

## **2.4 Encoder Rotativo Incremental**

Para posicionar correctamente las piezas a lo largo de la cinta es necesario contabilizar el movimiento de la misma. Para ello, se cuenta con un encoder rotativo incremental de serie LPD3806-600BM el cual cuenta con una gran precisión.

Éste tiene dos salidas de onda cuadrada con un desfase de 90 grados entre ellas.

### **2.4.1 Especificaciones técnicas**

El encoder rotativo de serie LPD3806-600BM cuenta con las siguientes especificaciones:

- 600 pulsos/revolución por cada fase. Por lo tanto, con las fases combinadas se cuenta con 2400 pulsos/revolución.
- Velocidad máxima: 5000 revoluciones/minuto.
- Respuesta de frecuencia: 0-30KHz

### **2.4.2 Conexionado del Encoder**

El encoder cuenta con cuatro cables de conexión:

- Rojo. Alimentación 5-24V.
- Negro. GND.
- Verde. Fase A.
- Blanco. Fase B.

## **2.5 Calibrador digital**

## **2.6 LCD**

## **2.7 Sensor fotoeléctrico OMRON**

## **2.8 Banda transportadora con elementos integrados**

## **2.9 Controlador IRC5C**



### **3 Desarrollo de placa de conexiones**

---



## **4 Planificación de caja**

---





## **5 Desarrollo en Arduino**

---



## **6 Desarrollo en Robotstudio**

---



## 7 Resultados

---



## 8 Conclusiones

---





# Índice de Figuras

---

1.1	Estructura de un aplicación que use la plataforma	3
2.1	Arduino Mega 2560	5



# Índice de Tablas

---

2.1	Características Arduino Mega 2560	6
-----	-----------------------------------	---



## Índice de Códigos

---



# Bibliografía

---

- [1] Jorge Andrés Tapia Herrera and Luis Fernando Castaño Castaño, *Control de equipo de posicionado de piezas semiautomático en zona de trabajo de robot*, 2018.
- [2] Mauricio Hinojosa Rea, Luis Fernando Castaño Castaño, and David Muñoz de la Peña Sequeda, *Conexión de robotstudio y arduino mediante tcp/ip para la recolección y envío de datos de posicionamiento de cinta transportadora*, 2019.