

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

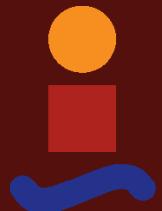
Integración de robot manipulador con
posicionador basado en arduino

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

En 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
1 Introducción	1
1.1 Modos de funcionamiento	1
1.2 Distribución de los controles	2
2 Descripción del hardware	5
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	9
2.5 Calibre digital	10
2.6 LCD	11
2.7 Sensor fotoeléctrico OMRON	11
2.8 Cinta transportadora con elementos integrados	12
2.9 Convertidor CC-CC con salida USB	13
2.10 Controladora IRC5C	13
2.11 Optoacoplador TLP621-4	13
3 Desarrollo de placa de conexiones	15
3.1 Software utilizado. KiCAD.	15
3.2 Niveles lógicos de tensión	15
3.3 Señales digitales en 24V y en 5V	16
3.4 Calibre digital	17

4 Planificación de caja	19
5 Desarrollo en Arduino	21
6 Desarrollo en Robotstudio	23
7 Resultados	25
8 Conclusiones	27
<i>Índice de Figuras</i>	29
<i>Índice de Tablas</i>	31
<i>Índice de Códigos</i>	33
<i>Bibliografía</i>	35

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
1 Introducción	1
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
1.2 Distribución de los controles	2
2 Descripción del hardware	5
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 Conexionado del módulo L298N	8
2.4 Encoder Rotativo Incremental	9
2.4.1 Especificaciones técnicas	9
2.4.2 Conexionado del Encoder	10
2.5 Calibre digital	10
2.6 LCD	11
2.7 Sensor fotoeléctrico OMRON	11
2.7.1 Conexiones del sensor fotoeléctrico	11
2.8 Cinta transportadora con elementos integrados	12
2.9 Convertidor CC-CC con salida USB	13
2.10 Controladora IRC5C	13
2.11 Optoacoplador TLP621-4	13
3 Desarrollo de placa de conexiones	15

3.1	Software utilizado. KiCAD.	15
3.2	Niveles lógicos de tensión	15
3.3	Señales digitales en 24V y en 5V	16
3.3.1	Implementación de optoacopladores	16
3.4	Calibre digital	17
3.4.1	Alimentación calibre	17
3.4.2	Amplificación señales calibre	18
4	Planificación de caja	19
5	Desarrollo en Arduino	21
6	Desarrollo en Robotstudio	23
7	Resultados	25
8	Conclusiones	27
<i>Índice de Figuras</i>		29
<i>Índice de Tablas</i>		31
<i>Índice de Códigos</i>		33
<i>Bibliografía</i>		35

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}^T	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 2π
e^{-jx}	Exponencial compleja negativa
$e^{-j2\pi x}$	Exponencial compleja negativa con 2π
IRe	Parte real
IIm	Parte imaginaria
sen	Función seno
tg	Función tangente
arc tg	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°	Notación de grado, x grados.
$\Pr(A)$	Probabilidad del suceso A
$E[X]$	Valor esperado de la variable aleatoria X
σ_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 σ_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}
$d\mathbf{x}$	Diferencial de \mathbf{x}
\leqslant	Menor o igual
\geqslant	Mayor o igual
\backslash	Backslash
\Leftrightarrow	Si y sólo si
$x = a + 3 \stackrel{\substack{\uparrow \\ a=1}}{=} 4$	Igual con explicación
$\frac{a}{b}$	Fracción con estilo pequeño, a/b
Δ	Incremento
$b \cdot 10^a$	Formato científico
\rightarrow_x	Tiende, con x
O	Orden
TM	Trade Mark
$\mathbb{E}[x]$	Esperanza matemática de x
$\mathbf{C}_{\mathbf{x}}$	Matriz de covarianza de \mathbf{x}
$\mathbf{R}_{\mathbf{x}}$	Matriz de correlación de \mathbf{x}
σ_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.

1.2 Distribución de los controles

Para una interacción simple con el sistema se realiza una

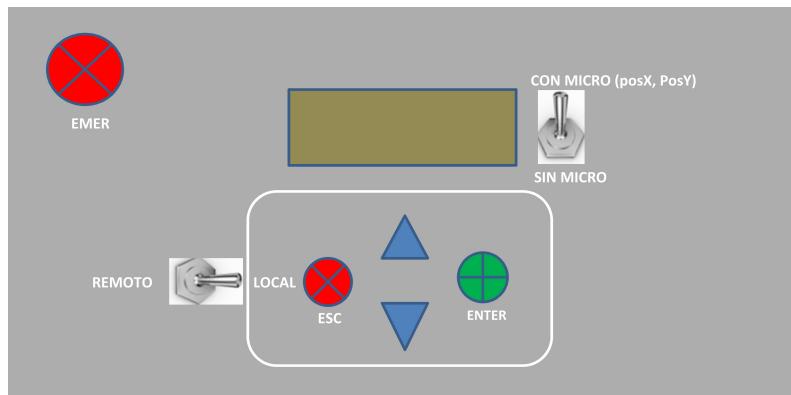


Figura 1.1 Distribución de elementos en la tapa.



Figura 1.2 Distribución de conexiones digitales.

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.

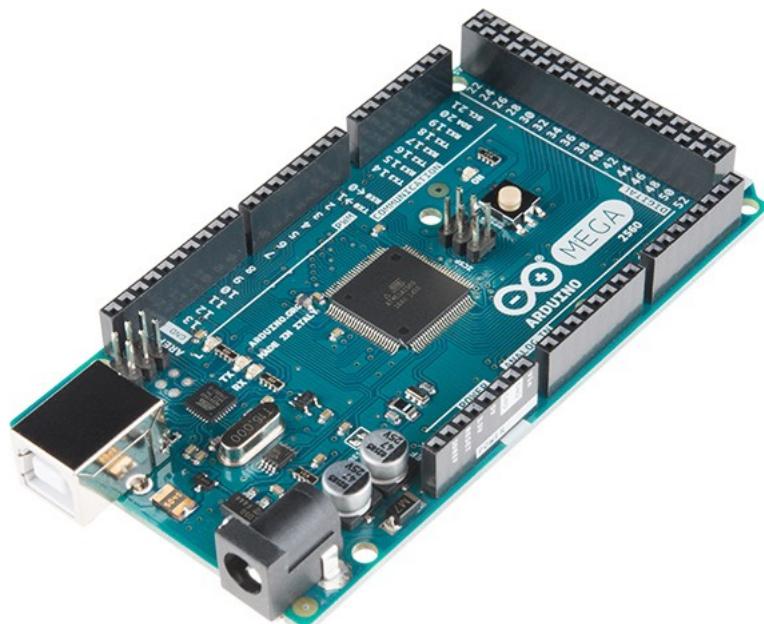


Figura 2.1 Arduino Mega 2560.

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.



Figura 2.2 Arduino Ethernet Shield.

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.

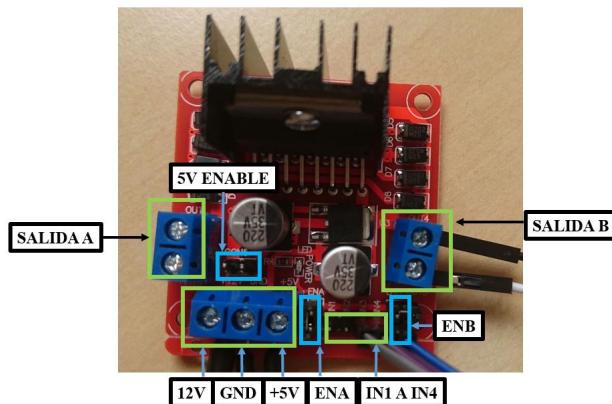


Figura 2.3 Driver L298N.

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 Conexión 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.



Figura 2.4 Tipos de conexiones del Driver L298N.

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. Siempre que se produzca un flanco en A, será leída la señal B. Si B se encuentra en HIGH, el encoder se encontrará girando en sentido horario. Por el contrario, si B se encuentra en LOW, el encoder se encontrará girando en sentido antihorario.



Figura 2.5 Señales encoder rotativo.

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 cuables de conexión:

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

2.5 Calibre digital

Para tomar la medida de la posición a lo ancho de la cinta se utiliza un calibre digital. Este instrumento se encuentra integrado en las cintas transportadoras del laboratorio. Cuentan con un LCD en el que se muestra la información de la medición y, además, cuenta con una tapa desmontable a la que se acceden cuatro pines que permiten comunicaciones con el Arduino. Estos pines son: alimentación a 1.5V, GND, señal de reloj y señal de datos. Todas las señales utilizan un nivel lógico de tensión de 1.5V, por lo que deben ser amplificadas a 5V para que el Arduino pueda comprender dichas señales.



Figura 2.6 Calibre digital y cables usados.

Mientras el calibre cuente con alimentación estará enviando tramas de datos con la medida de cada momento, por lo que es recomendable no utilizar pilas ya que siempre está funcionando. Algunas características importantes son:

- Cada trama cuenta con 24 bits, de los cuales 21 corresponden a la medición, uno para el signo, uno para la unidad (mm o in) y otro bit de acarreo.
- Los datos se transmiten por una señal de reloj (CLK) y una señal de datos (DATA).
- La línea de datos debe leerse en cada flanco de bajada de la señal de reloj.
- Los bits de la trama se inician por el menos significativo (LSB)
- El valor de la medida en mm debe ser multiplicado por 100.

2.6 LCD

Para realizar visualizaciones de información e interactuar con el sistema se utiliza un LCD 16x2. Este LCD cuenta con un módulo basado en microcontrolador que permite controlar la información mostrada mediante comunicación I2C, lo que permite reducir en gran cantidad los pines utilizados.

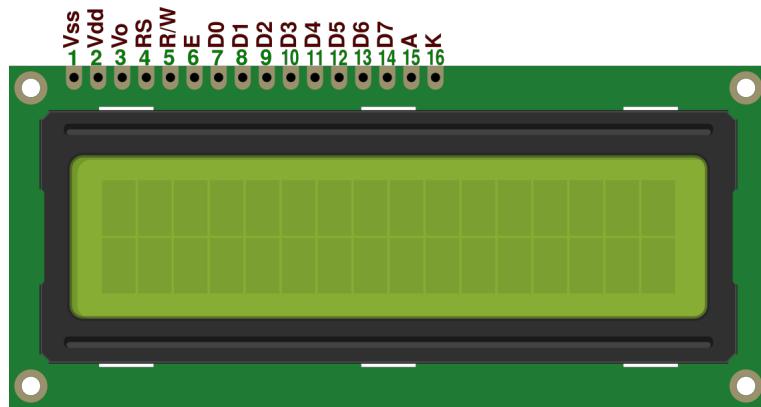


Figura 2.7 LCD 16x2.

La conexión se realiza mediante cuatro pines: alimentación 5V, GND, SCL y SDA. Además, la placa de adaptación cuenta con un potenciómetro que permite regular el brillo del LCD.



Figura 2.8 Módulo de control de LCD por I2C.

2.7 Sensor fotoeléctrico OMRON

Se utiliza un sensor fotoeléctrico para detectar que la pieza ha llegado a cierta posición arbitraria. Para ello se utiliza el sensor E3JK-R4M de la marca OMRON. Este sensor se encuentra incorporado en la cinta transportadora del laboratorio. Tiene dimensiones reducidas y tiene una alta capacidad de conmutación. El sensor es de tipo retroreflectivo polarizado, lo que permite detectar cuerpos brillantes. Además, posee un LED de color rojo que se enciende al detectar un objeto.

2.7.1 Conexiones del sensor fotoeléctrico

El sensor fotoeléctrico cuenta con cinco cables de conexión de diferentes colores:

- Marrón. Alimentación entre 12-24V.



Figura 2.9 Sensor fotoeléctrico OMRON E3JK-R4M.

- Azul. GND.
- Blanco. Salida común del relé.
- Negro. Salida de relé NA (Normalmente Abierto).
- Gris. Salida de relé NC (Normalmente Cerrado).

2.8 Cinta transportadora con elementos integrados

En el laboratorio de automática se encuentra una cinta transportadora con todos los elementos integrados, de modo que el proyecto consiste en definitiva en una interfaz entre estos elementos y el robot que le acompaña.

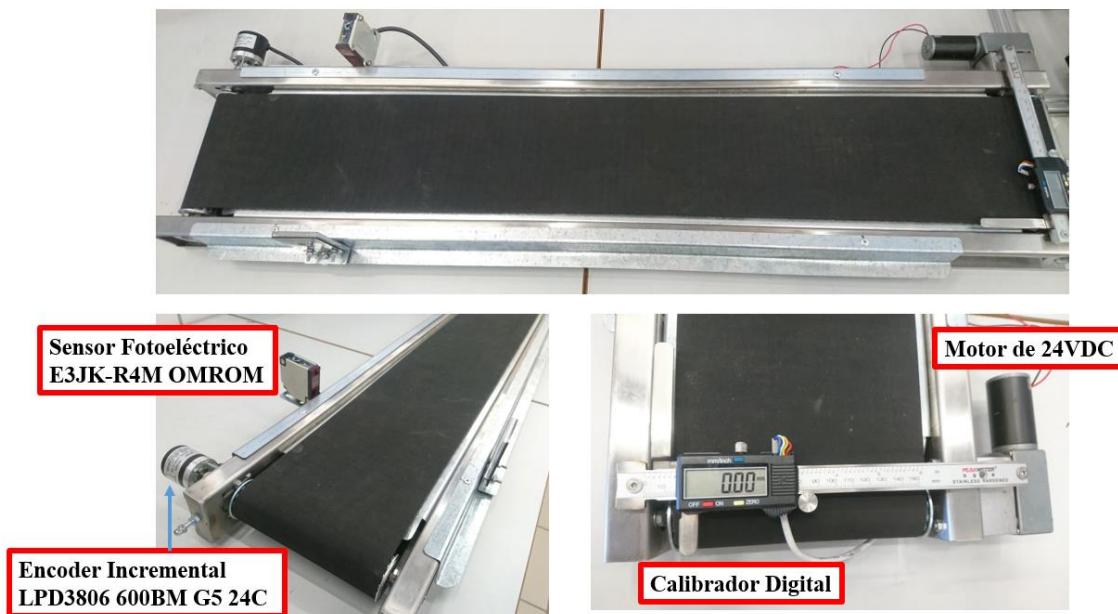


Figura 2.10 Cinta transportadora con elementos integrados.

2.9 Convertidor CC-CC con salida USB

Se trata de un pequeño dispositivo que se puede alimentar con un amplio rango de tensiones. En este caso, se puede alimentar con los 24V que proporciona la controladora del robot. Cuenta con cuatro salidas USB (que funcionan a 5V) con las que se puede alimentar la placa de Arduino y toda la electrónica del sistema.

2.10 Controladora IRC5C

El robot ABB IRB120 presente en los laboratorios del departamento cuentan con unas controladoras IRC5C, que son los dispositivos con los que se tiene que interactuar para comunicar la cinta transportadora con el robot. Estos dispositivos funcionan a 24V y cuentan con su propia fuente de alimentación. Por otra parte, cuentan con todas las conexiones necesarias para la realización del proyecto. Entre ellas, una conexión Ethernet a la que conectar el Ethernet Shield de Arduino.

Por otro lado, cuenta con una placa DSQC 652, que permite la conexión de señales digitales. Estas señales tienen un nivel lógico de 24V, necesitando un mínimo de 15V para detectar un '1' y un máximo de 5V para detectar un '0', por lo que las señales que les sean enviadas por esta vía deben ser adaptadas a dichas tensiones.

2.11 Optoacoplador TLP621-4

El circuito integrado TLP621-4 se trata de un chip que cuenta con cuatro aisladores ópticamente acoplados que constan con un diodo emisor de luz infrarroja y un fototransistor de silicio NPN encapsulado en plástico.

Las principales características son:

- Tensión de aislamiento en AC de 5300Vrms.
- Rango de temperatura de funcionamiento: -30°C a 100°C.
- Sin plomo.
- Paquete DIP de 16 pines.

Como dato relevante, la corriente máxima que debe pasar sobre los diodos es de 50 mA, por lo que las resistencias a colocar deberán condicionar este parámetro.

3 Desarrollo de placa de conexiones

Para simplificar todas las conexiones internas dentro de la caja y posibilitar ciertas funciones específicas, se crea una placa electrónica donde se conectan todos los dispositivos. De este modo se consigue simplificar el montaje y la posible actualización del sistema, ya que todas las conexiones internas desaparecen por completo. Además, para favorecer el orden de los cables dentro de la caja se crea una segunda placa más básica cuya única función es agrupar todas las conexiones existentes en la tapa. Se conocerá como placa *Fondo* a la placa general y placa *Tapa* a la superior.

3.1 Software utilizado. KiCAD.

Para la realización de la placa se ha utilizado el paquete de software KiCAD. Se trata de un software libre bajo licencia GNU General Public License, lo que permite su uso sin coste para el proyecto. KiCAD es un paquete de software orientado hacia el diseño electrónico (EDA por sus siglas en inglés: Electronic Design Automation) y consta de diversas aplicaciones que permiten realizar todo el trabajo.



Figura 3.1 Logo de KiCAD.

Por un lado cuenta con *eeschema*, un editor de esquemas electrónicos donde se puede plantear la lógica de las conexiones de un modo abstracto. Por otro, se encuentra con *pcbnew*, un editor de circuitos impresos. A partir de un esquemático creado se pasa a un circuito impreso de modo fácil y siendo modificable siempre que sea necesario.

3.2 Niveles lógicos de tensión

En primer lugar, el proyecto cuenta con dispositivos que funcionan a distintos niveles de tensión, por lo que se tiene que plantear las transformaciones a realizar. Los dos principales niveles de tensión son los 5V a los que funciona el Arduino y los 24V a los que funciona el motor y la controladora

del robot. Por otra parte, también se cuenta con el nivel lógico de 1.5V del calibre digital.

El sistema se alimenta con 24V provenientes de la controladora y pasa a 5V mediante el convertidor externo con el que se cuenta, por lo que con obtener dichas conexiones del propio sistema ya se dispone las dos líneas. Sin embargo, en el caso de los 1.5V éstos deben ser generados dentro de la placa como se verá en la sección correspondiente más adelante.

3.3 Señales digitales en 24V y en 5V

El sistema cuenta con cuatro salidas digitales que deben estar representadas tanto en 24V como en 5V para que tanto el Arduino como la controladora conozcan de modo inequívoco el modo en el que se encuentra el sistema. Estas señales son:

- Microcontrolador (MICRO). '1'si el microcontrolador se encuentra desactivado y '0'si está operativo.
- Local/Remoto (LR). '1'si el sistema funciona en modo remoto y '0'si funciona en modo local.
- Emergencia (EMER). '1'si el sistema requiere una parada de emergencia.
- Sensor Fotoeléctrico (FOTO). '1'si hay una pieza detectada por el sensor.

Además, se cuenta con dos entradas digitales que deben ser capaces de mover el motor en el caso de que el sistema se encuentre funcionando sin microcontrolador. Para ello estas señales deben actuar sobre los pines de dirección del L298N. Estas dos señales son:

- Avance (AVANCE). Actúa sobre el pin IN1 o IN3 del L298N.
- Retroceso (RETR). Actúa sobre el pin IN2 o IN4 del L298N.

Las señales se generan a 24V por defecto, por lo que es necesario su paso a 5V. Para ello se utilizan optoacopladores, unos dispositivos que mediante fotodioides acoplados a fototransistores. En este proyecto se utiliza el circuito integrado TLP621-4 que cumple dicha función.

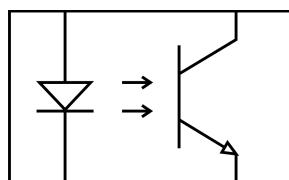


Figura 3.2 Ejemplo de optoacoplador.

3.3.1 Implementación de optoacopladores

En la figura 3.3 se muestra el circuito empleado. Se utiliza un total de dos TLP621-4 para tener un total de 8 optoacopladores. Como la tensión máxima que reciben los diodos es de 24V, se colocan unas resistencias para evitar que se quemen. El valor de las resistencias utilizadas es de $8.2\text{ k}\Omega$, permitiendo que la corriente directa sea algo inferior a 3 mA. Como caso relevante, en el caso de la parada de emergencia, la resistencia será de $4.7\text{ k}\Omega$ para que tenga una corriente directa y, una mayor salida por ello para asegurar que se produzca la parada en caso de ser necesaria.

En el colector de los fototransistores se conectan directamente la línea de 5V para que, en caso de que la señal correspondiente se active, su equivalente de 5V también lo haga. En las señales FOTO, LR, MICRO y EMER se les añade un LED a la salida a modo de test para comprobar su funcionalidad, pero no se verán desde fuera de la caja.

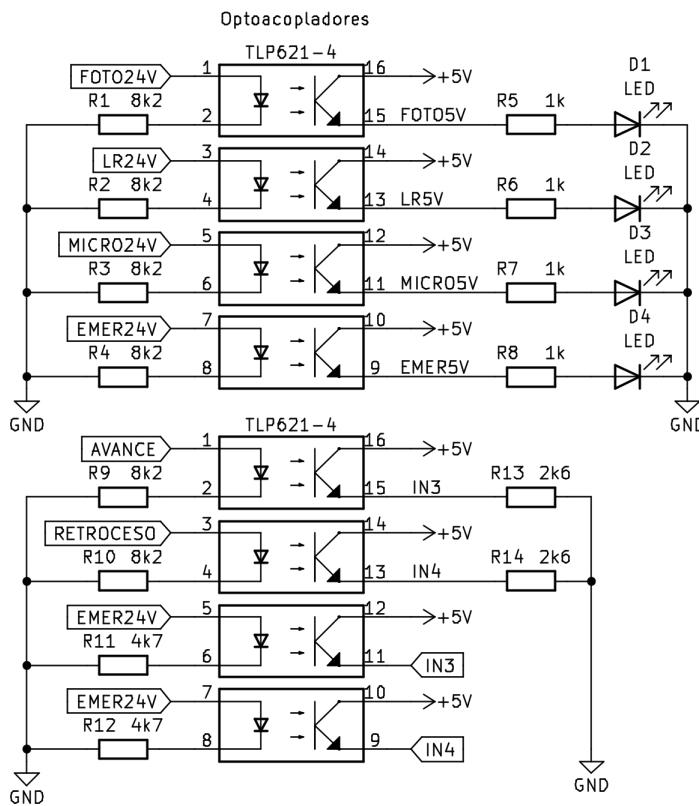


Figura 3.3 Implementación de optoacopladores.

3.4 Calibre digital

3.4.1 Alimentación calibre

Para el funcionamiento del calibre digital, éste debe ser alimentado por una fuente de 1.5V. Como lo que se dispone en este proyecto es de 5V y 24V lo más sencillo es colocar un divisor resistivo a la línea de 5V y utilizar un búfer de tensión para mantener dicho nivel lógico estable. Utilizando resistencias lo suficientemente altas el consumo es despreciable. Por ello se utiliza una resistencia de $39k\Omega$ y otra de $100k\Omega$. La tensión de salida obtenida será de:

$$V_{out} = 5V \frac{39k\Omega}{39k\Omega + 100k\Omega} \approx 1.4V \quad (3.1)$$

Los 1.4V que se obtienen entran dentro del rango de funcionamiento del calibre utilizando resistencias comerciales, por lo que el resultado es válido.

El consumo del divisor resistivo por otro lado será de:

$$I_{divisor} = \frac{5V}{39k\Omega + 100k\Omega} \approx 0.036mA \quad (3.2)$$

Se puede considerar un consumo despreciable.

Para el búfer de tensión se utiliza un circuito integrado TL082, del cual se utiliza uno de los dos amplificadores operacionales con el que cuenta. Como resultado se obtiene el circuito de la figura 3.4.

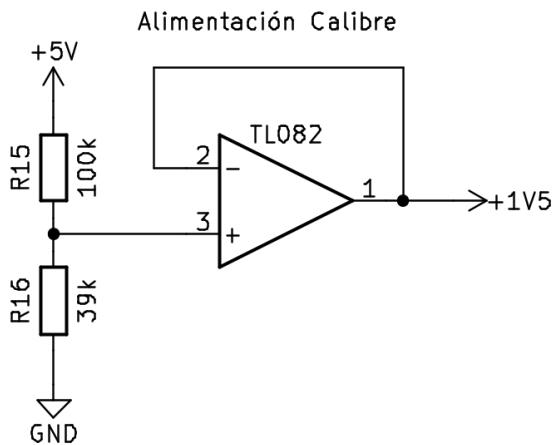


Figura 3.4 Alimentación calibre digital a 1.5V.

3.4.2 Amplificación señales calibre

Para la lectura de la señal del calibre por parte del Arduino es necesario elevar los niveles de tensión a 5V. Para ello se toma como referencia el circuito visto en la web [3]. Éste utiliza un transistor junto con dos resistencias de modo que cuando el calibre envíe un '1' se eleve la tensión a 5V, pero sin distorsionarse cuando haya un '0' y dar falsos positivos. El esquema implementado es el que aparece en la figura 3.5.

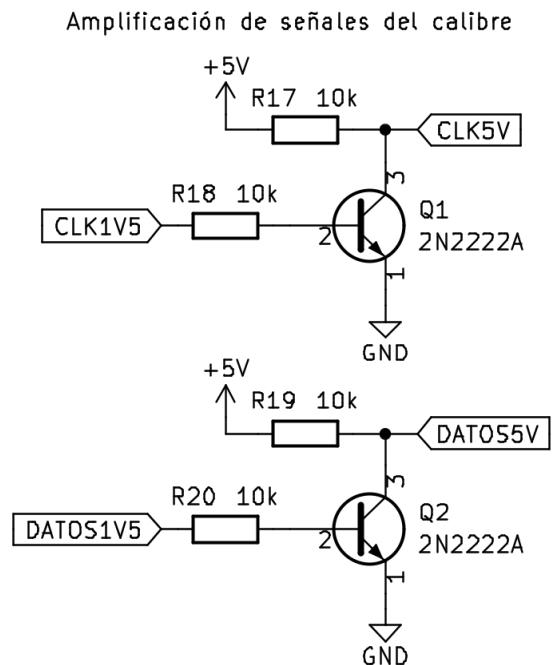


Figura 3.5 Amplificación señales calibre digital a 5V.

4 Planificación de caja

5 Desarrollo en Arduino

6 Desarrollo en Robotstudio

7 Resultados

8 Conclusiones

Índice de Figuras

1.1	Distribución de elementos en la tapa	3
1.2	Distribución de conexiones digitales	3
2.1	Arduino Mega 2560	5
2.2	Arduino Ethernet Shield	7
2.3	Driver L298N	8
2.4	Tipos de conexiones del Driver L298N	9
2.5	Señales encoder rotativo	9
2.6	Calibre digital y cables usados	10
2.7	LCD 16x2	11
2.8	Módulo de control de LCD por I2C	11
2.9	Sensor fotoeléctrico OMRON E3JK-R4M	12
2.10	Cinta transportadora con elementos integrados	12
3.1	Logo de KiCAD	15
3.2	Ejemplo de optoacoplador	16
3.3	Implementación de optoacopladores	17
3.4	Alimentación calibre digital a 1.5V	18
3.5	Amplificación señales calibre digital a 5V	18

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 posicionamiento de piezas semiautomático en zona de trabajo de robot*, 2018.
- [2] Mauricio Hinojosa Rea, Luis Fernando Castaño Castaño, and David Muñoz de la Peña Sequedo, *Conexión de robotstudio y arduino mediante tcp/ip para la recolección y envío de datos de posicionamiento de cinta transportadora*, 2019.
- [3] Martin Thalheimer, <https://sites.google.com/site/marthalprojects/home/arduino/arduino-reads-digital-caliper>.