



Escuela Superior de Ciencias Experimentales y Tecnología

**GRADO EN INGENIERÍA
DE TECNOLOGÍAS INDUSTRIALES**

Trabajo de Fin de Grado

**DISEÑO Y FABRICACIÓN DE UNA CINTA TRANSPORTADORA
CONSTRUIDA CON COMPONENTES RECUPERADOS. COMPARATIVA
DE LA IMPLEMENTACIÓN DEL CONTROL CON ARDUINO Y LÓGICA
PROGRAMABLE.**

Rodellar Suárez, Clara

Director: Machado Sánchez, Felipe

Curso Académico 2019/2020



Universidad
Rey Juan Carlos

Grado en Ingeniería de Tecnologías Industriales

Trabajo de Fin de Grado

El presente trabajo, titulado: **DISEÑO Y FABRICACIÓN DE UNA CINTA TRANSPORTADORA CONSTRUIDA CON COMPONENTES RECUPERADOS. COMPARATIVA DE LA IMPLEMENTACIÓN DEL CONTROL CON ARDUINO Y LÓGICA PROGRAMABLE**, constituye la memoria correspondiente a la asignatura Trabajo de Fin de Grado que presenta **D^a. Clara Rodellar Suárez** como parte de su formación para aspirar al Título de Graduada en Ingeniería de Tecnologías Industriales.

Este trabajo ha sido realizado en la **Escuela Superior de Ciencias Experimentales y Tecnología de la Universidad Rey Juan Carlos** en el **Departamento de Matemática Aplicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica** bajo la dirección de **Felipe Machado Sánchez**.

Móstoles, 8 de octubre de 2019

AGRADECIMIENTOS

Me gustaría agradecer a mi tutor, Felipe Machado, todo su tiempo de dedicación y paciencia, sin su dirección y ayuda no hubiera sido capaz de realizar el proyecto. Su constante apoyo y dedicación ha hecho posible que no sólo realice este trabajo, si no que aprendiera cómo aplicar todos los conocimientos impartidos en la carrera de forma práctica.

También deseo agradecer a Pablo Aguado su inestimable ayuda en el desarrollo práctico del trabajo que ha estado siempre disponible para ayudarme.

Índice

AGRADECIMIENTOS	3
Índice	4
Lista de acrónimos	6
Índice de figuras	7
Índice de tablas	9
RESUMEN	10
1. INTRODUCCIÓN	12
1.1. Ámbito científico-tecnológico	12
1.1.1. Mecatrónica	12
1.1.2. Hardware libre y movimiento Maker	13
1.1.3. Electrónica de control	14
1.2. Cintas transportadoras	16
1.2.1. Descripción de su uso	16
1.2.2. Elementos que lo componen	16
1.2.3. Tipos de cintas transportadoras	18
1.2.4. Estado del arte: cintas transportadoras	19
2. OBJETIVOS	21
3. SOLUCIÓN TÉCNICA	22
3.1. Análisis del prototipo	22
3.2. Sistema mecánico	25
3.2.1. Elementos de la cinta transportadora	25
3.2.2. Piezas diseñadas	26
3.2.3. Impresión 3D	29
3.2.4. Ensamblaje del sistema	30
3.3. Sistema electrónico de potencia	31
3.3.1. Introducción	31
3.3.2. Proporcionar corriente: transistores y optoacopladores	31
3.3.3. Control de velocidad: Pulse Width Modulation (PWM)	32
3.3.4. Inversión del sentido: Puente en H	33
3.3.5. Puente en H integrado	37
3.4. Sistema electrónico de control	38

3.4.1.	Encoder rotativo	38
3.4.2.	Lazo abierto vs lazo cerrado	40
3.4.3.	Arduino	43
3.4.4.	FPGA	44
3.4.5.	Conclusiones del estudio comparativo entre Arduino y FPGA.....	51
3.5.	Estudio económico	53
3.6.	Líneas futuras	55
4.	CONCLUSIONES	56
5.	BIBLIOGRAFÍA	57
6.	ANEJOS	61
Anejo I	Herramientas utilizadas.....	61
Anejo II	Diagrama de Gantt	64
Anejo III	Explicación de los componentes necesarios y cómo elegirlos para el circuito electrónico de potencia.....	65
Anejo IV	Cálculo de la resistencia conectada al optoacoplador en función del dispositivo de control. 70	
Anejo V	Pruebas realizadas al encoder.....	73
Anejo VI	Unidades de velocidad	75
Anejo VII	Análisis del tiempo de lectura del encoder en Arduino	77
Anejo VIII	Diseño de bloques en VHDL	79

Lista de acrónimos

CAD	<i>Computer-Aided Design</i> [Diseño Asistido por Ordenador]
FPGA	<i>Field Programmable Gate Array</i> [Matriz de puertas programables en campo]
DIY	<i>Do It Yourself</i> [Hazlo tú mismo]
VHDL	<i>(VHSIC) HDL, Very High Speed Integrated Circuit Hardware Description Language</i> [Lenguaje de Descripción Hardware de Circuito Integrado de Alta Velocidad]
OSH	<i>Open Source Hardware</i> [Hardware libre]
OSHW	<i>Open Source Hardware Association</i> [Asociación de hardware libre]
CLB	<i>Configurable Logic Block</i> [Bloque Lógico Configurable]
LUT	<i>Look-Up Table</i> [Tabla de consulta]
PWM	<i>Pulse Width Modulation</i> [Modulación por ancho de pulsos]
EVA	Etileno-vinil-acetato
ISE	<i>Integrated Synthesis Environment</i> [Entorno de síntesis integrado]
IDE	<i>Integrated Development Environment</i> [Entorno de desarrollo integrado]
FFF	<i>Fused Filament Fabrication</i> [Fabricación por filamento fundido]
PLA	Ácido poliáctico
ABS	Acrilonitrilo butadieno estireno

Índice de figuras

Figura 1.1: Concepto de mecatrónica [4]	12
Figura 1.2: Funcionamiento de un sistema mecatrónico [Machado, F. Material del aula de mecatrónica, URJC]	13
Figura 1.3: Estructura interna de una FPGA	15
Figura 1.4: Esquema de una cinta transportadora [16]	18
Figura 3.1: Imagen del montaje de la cinta transportadora	22
Figura 3.2: Esquema de las partes del sistema mecatrónico de la banda transportadora	23
Figura 3.3: Desmontaje de la impresora HP Deskjet845c	25
Figura 3.4: Vista del sistema de transmisión de movimiento del motor al eje por engranajes.....	26
Figura 3.5: Vista de acoplamiento del tambor motriz en el extremo donde se encuentra el motor ...	27
Figura 3.6: Soporte de ambos tambores en sus extremos libres	27
Figura 3.7: Soporte del tambor de reenvío en el extremo con engranaje	28
Figura 3.8: Soporte del tambor motriz en el extremo con el motor (no usada)	28
Figura 3.9: Soporte ambos tambores en sus extremos libres (no usada)	29
Figura 3.10: Cómo se debe y no se debe imprimir una pieza en 3D	29
Figura 3.11: Problema de impresión al no fijarse bien la pieza a la base	30
Figura 3.12: Ensamblaje del sistema	30
Figura 3.13: Conexión de dispositivo de control con un motor	32
Figura 3.14: Señal PWM	32
Figura 3.15: Característica V-I de salida de un mosfet	34
Figura 3.16: Puente en H con interruptores [30]	34
Figura 3.17: Esquema del funcionamiento eléctrico del circuito con entradas L-L	35
Figura 3.18: Esquema del funcionamiento eléctrico del circuito con entradas L-H.....	35
Figura 3.19: Esquema del funcionamiento eléctrico del circuito con entradas H-L.....	36
Figura 3.20: Esquema del funcionamiento eléctrico del circuito con entradas H-H.....	36
Figura 3.21: Tipos de encoder [Machado, F. Material del aula de mecatrónica, URJC]	38
Figura 3.22: Encoder recuperado	39
Figura 3.23: Funcionamiento de encoder con dos sensores en cuadratura para detectar el sentido de movimiento [Machado, F. Material del aula de mecatrónica, URJC].....	39
Figura 3.24: Diagrama de bloques del sistema de control en lazo abierto	40
Figura 3.25: Diagrama de bloques del sistema de control en lazo abierto del proyecto.....	40

Figura 3.26: Diagrama de bloques del sistema de control en lazo cerrado	41
Figura 3.27: Diagrama de bloques del sistema de control en lazo cerrado del proyecto	41
Figura 3.28: Interfaz FPGA.....	44
Figura 3.29: Diseño estructural del control de la cinta en lazo abierto en FPGA.....	45
Figura 3.30: Cronograma de detector de flanco	46
Figura 3.31: Señal afectada por efecto rebote.....	47
Figura 3.32: Diseño estructural del control de la cinta en lazo cerrado en FPGA.....	49
Figura 3.33: Simulación del programa de control en lazo cerrado con la FPGA	51
Figura 6.1: Conexión del motor de continua para realizarle los ensayos	65
Figura 6.2: Puente en H con transistores bipolares [30]	67
Figura 6.3: Característica V-I de salida de un mosfet [Romero, B. Material del aula de tecnología electrónica, URJC].....	67
Figura 6.4: Funcionamiento de mosfet en corte y en zona lineal para canal n y p [Machado, F. Material del aula de diseño electrónico digital, URJC].....	68
Figura 6.5: Conexión interna de un optoacoplador [44]	68
Figura 6.6: Índices absolutos máximos del componente 4N35 para el diodo	70
Figura 6.7: CTR normalizado no saturado y saturado vs. Corriente LED.....	70
Figura 6.8: Características eléctricas del led del componente 4N35	71
Figura 6.9: Forma de la señal del encoder vista con el osciloscopio.....	73
Figura 6.10: Lógica de componentes de 5 V y 3,3 V [45]	74
Figura 6.11: Tiempo que tarda Arduino en ejecutar el código en milisegundos	78
Figura 6.12: Esquema del diseño digital para hacer lucir el display de la FPGA	80

Índice de tablas

Tabla 1.1: Comparación entre Arduino y FPGA.....	16
Tabla 1.2: Cintas transportadoras comerciales	19
Tabla 1.3: Cintas transportadoras DIY	20
Tabla 3.1: Resumen del funcionamiento del circuito eléctrico en función de las entradas	36
Tabla 3.2: Ensayo en Arduino para comprobar su capacidad para leer el encoder	43
Tabla 3.3: Resumen de uso de los controladores	52
Tabla 3.4: Estudio económico del proyecto	53
Tabla 6.1: Diagrama de Gantt de programación del proyecto	64
Tabla 6.2: Funcionamiento optoacoplador en función de las entradas.....	69

RESUMEN

Este trabajo fin de grado consiste en el diseño, estudio del control y fabricación de una cinta transportadora de bajo coste construida con componentes recuperados.

Para la construcción de la cinta se utilizan componentes recuperados, contribuyendo a un desarrollo sostenible, concienciando de la problemática actual de la basura electrónica y dando lugar a un proyecto más económico. Asimismo se diseñan piezas en 3D para posteriormente imprimirlas e incluirlas en el diseño de la cinta transportadora.

También se trabaja en el diseño de circuitos analógicos, desarrollando el circuito de electrónica de potencia necesario para el proyecto. El diseño de circuitos constituye uno de los niveles más avanzados en el aprendizaje de la electrónica, ya que se tienen que evaluar las distintas alternativas, poner en práctica todos los conocimientos teóricos adquiridos durante el grado y manejar equipos electrónicos y hojas de características de los componentes. El diseño engloba, además, otros niveles del aprendizaje, como la comprensión de los conceptos teóricos y su aplicación, el análisis y la evaluación.

El estudio de control analiza dos posibles alternativas: con lazo abierto y con lazo cerrado. Se implementan ambos controles con dos dispositivos distintos, Arduino y FPGA, y se estudia con cuál de los dos es mejor trabajar en función del tipo de control que se quiera realizar. En este estudio se ha observado que es mejor el uso de Arduino en control con lazo abierto, por su sencillez frente a la FPGA. En cambio, en lazo cerrado, donde se necesita la lectura de señales en tiempo real, se concluye que es mejor el uso de la FPGA debido a que Arduino no es capaz de leer el sensor utilizado a la velocidad de trabajo.

Además, este trabajo se compromete con la filosofía de hardware libre y el movimiento Maker, de manera que se contribuye a la tendencia del desarrollo de equipos científicos y de laboratorio de bajo coste. Se pretende usar en el laboratorio del Área de Tecnología Electrónica de la Universidad Rey Juan Carlos, dándole aplicaciones docentes. Pero, no es sólo eso lo que impulsan estas filosofías, sino que promueven el trabajo colaborativo y abierto. Por ello, este trabajo es accesible desde la plataforma de Github de la autora del proyecto [1]. Desde este enlace cualquiera tiene acceso a toda la documentación, piezas diseñadas y código necesario para replicar el proyecto. Así, quien lo desee, puede estudiarlo, modificarlo, utilizarlo y distribuirlo, que son los principios del Hardware Libre.

Es importante destacar el elevado impacto formativo que conlleva este proyecto por incluir temáticas diferentes que corresponden, entre otras, a las siguientes asignaturas:

- Robótica Industrial y Mecatrónica, al desarrollar un sistema mecatrónico.

- Diseño de Sistemas Electrónicos, a través de la programación de la FPGA.
- Sistemas Electrónicos Digitales, por la programación con Arduino .
- Control y Automatización, trabajando col el lazo abierto y el lazo cerrado.
- Tecnología Electrónica, por el diseño del puente en H.
- Expresión Gráfica, diseñando piezas en 3D.
- Máquinas Eléctricas, por el control de motores.

1. INTRODUCCIÓN

En este trabajo se diseña y fabrica una cinta transportadora. Sobre ella se hace un estudio comparativo del control a través de un microcontrolador (Arduino) y un dispositivo lógico programable (FPGA: Field Programmable Gate Array), diseñando y construyendo ambas opciones.

Una cinta transportadora es un sistema mecatrónico, por lo que el proyecto abarca:

- La parte mecánica: incluye, entre otras cosas, el montaje de la cinta y el diseño asistido por ordenador (CAD) de las piezas.
- La parte electrónica: se encarga de dar potencia y controlar el circuito. Para realizar el estudio de control se implementan dos tipos distintos de control en ambos dispositivos y así posteriormente hacer una comparación entre ambos.

Además, se usan piezas reutilizadas para contribuir al desarrollo sostenible y abaratar el coste del proyecto.

1.1. Ámbito científico-tecnológico

El TFG puede enmarcarse en tres grandes ámbitos: la mecatrónica, el hardware libre y la electrónica de control.

1.1.1. Mecatrónica

El término mecatrónica surge en Japón en 1969 por el ingeniero Tetsuro Mori cuando trabajaba en la empresa Yaskawa Electric Corporation aunque no se permitió su libre uso hasta 1982 [2]. La definición que suele usarse para describir la mecatrónica es la de J. A. Rietdijk: "Mecatrónica es la combinación sinérgica de la ingeniería mecánica de precisión, de la electrónica, del control automático y de los sistemas para el diseño de productos y procesos" [3].

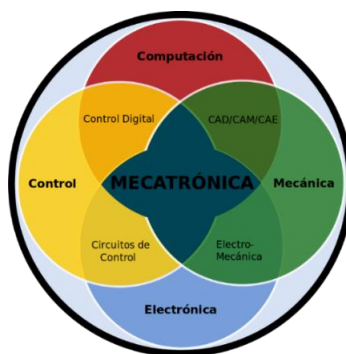


Figura 1.1: Concepto de mecatrónica [4]

Los sistemas mecatrónicos recogen señales mediante sensores que se acoplan al sistema mecánico, procesan la información que le llega mediante la electrónica de control y con esto proporciona una señal que actúa sobre el sistema mecánico. El acondicionamiento de señal manipula una señal para adaptarla a la aplicación que se necesite. Por ejemplo, la conexión de una resistencia en un interruptor es una forma de acondicionar la señal. Por último, los interfaces facilitan la comunicación entre el usuario y la máquina. Puede verse en la Figura 1.2 un esquema donde se aprecia la secuencia de este proceso.

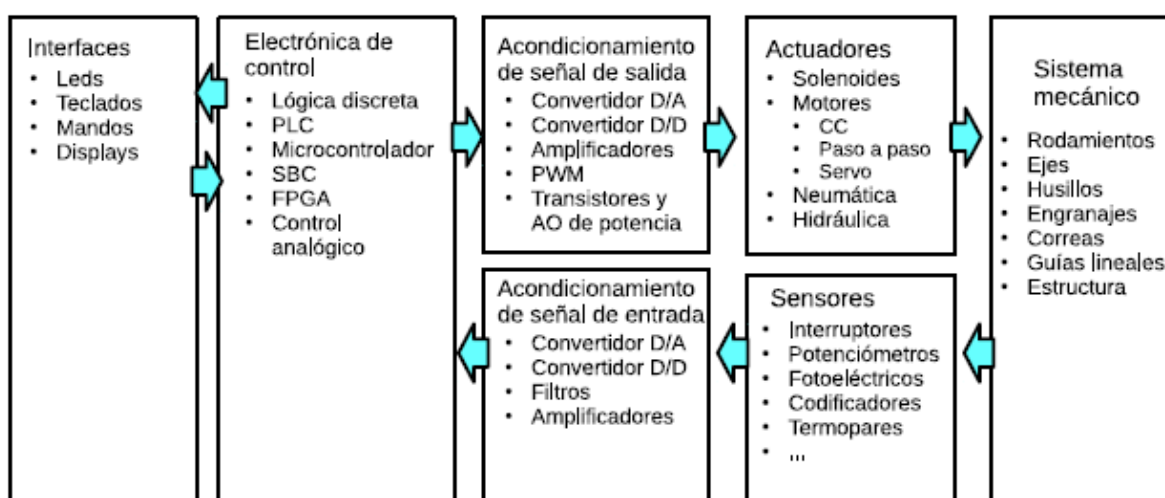


Figura 1.2: Funcionamiento de un sistema mecatrónico [Machado, F. Material del aula de mecatrónica, URJC]

Todo esto consigue que se obtengan productos que al automatizarse sean más inteligentes y ágiles. Entre otros sistemas mecatrónicos se pueden encontrar los robots, los vehículos autónomos, las impresoras 3D, etc. En definitiva, cualquier sistema que haga uso de las cuatro ingenierías mencionadas [2] [5].

1.1.2. Hardware libre y movimiento Maker

El hardware libre, denominado en inglés Open Source Hardware (OSH) es un movimiento que ha surgido en los últimos años buscando la creación y el intercambio de diseños libres. La definición dada por OSHWA (Open Source Hardware Association) de hardware libre se basa en la definición de software libre, donde suele hablarse de cuatro libertades: la libertad para poder estudiarse, modificarse, utilizarlo o distribuirlo, incluyendo su venta. OSHWA es una asociación de hardware libre que se encarga, entre otras cosas, de certificar proyectos como OSH [6] [7] [8].

Por otra parte está el movimiento Maker, que se basa en el desarrollo de dispositivos Do It Yourself (DIY) en lugar de comprarlos listos para usar. Esta corriente, impulsada por las impresoras 3D, Arduino y el trabajo colaborativo, promueve el aprendizaje y que la innovación tecnológica no quede en manos únicamente de grandes empresas [9].

La unión del hardware libre y el movimiento Maker ha dado lugar a que en los últimos años se consolide una tendencia que conduce al desarrollo de equipos científicos y de laboratorio libres, con buenas prestaciones y precios significativamente inferiores a los equipos comerciales. Esto permite que centros de investigación y docentes puedan tener acceso a equipos científicos avanzados sin que suponga una gran inversión.

Este trabajo ha sido realizado basándose en estas filosofías, de manera que se ha trabajado con herramientas libres como FreeCAD y todo el trabajo es accesible desde el perfil de Github de la autora del trabajo [1], adjunto en la bibliografía, para que cualquiera pueda reproducirlo, aprender de él y mejorarlo.

1.1.3. Electrónica de control

Como se ha comentado, un sistema mecatrónico debe tener un sistema de control. Hoy en día los que más se usan a nivel de usuario son los microcontroladores como Arduino o los dispositivos lógicos programables como las FPGAs. Ambos serán usados en el proyecto con el fin de ver sus pros y contras en función de las circunstancias y necesidades que se tengan en el proyecto y para aprender a controlar el sistema con ambos dispositivos.

El mayor problema de los microcontroladores como Arduino es que se limita a la arquitectura y los periféricos de los que disponga. Por tanto, es necesario fijar los requisitos del proyecto para poder elegir un microcontrolador que tenga las prestaciones necesarias. Dado que de los microcontroladores únicamente se puede cambiar el programa que se guarda en memoria, en caso de no haberlo escogido correctamente, habrá que cambiar el microcontrolador, y con esto, rehacer una gran parte del diseño.

En cambio, cuando se diseña con las FPGAs, no se está limitado a la estructura como sucedía en el microcontrolador, sino que hay que diseñar todo desde cero. La parte negativa es que hay que dedicarle más tiempo y esfuerzo al tener que crear todo lo que se necesita. Con un microcontrolador si se tiene el periférico, se puede usar directamente, pero con una FPGA lo debe diseñar el propio usuario. Esto da la libertad de poder diseñar los periféricos que se necesiten. Además, como estas trabajan de forma concurrente, a diferencia de los microcontroladores que lo hacen secuencialmente, se evita tener un procesador que soporte la mayoría de la carga de trabajo. Por tanto, si se tiene un

programa largo en un microcontrolador, puede bajar significativamente el rendimiento, mientras que esto no sucede con las FPGAs [10].

Por otra parte, la gran diferencia que hay entre los microcontroladores y las FPGAs es que con los microcontroladores se programa software mientras que con las FPGAs se describe el hardware que se quiere implementar. Esto es lo que el diseño con FPGA sea más complejo.

FPGA son las iniciales de Field Programmable Gate Array, lo que significa matriz de puertas programables en campo, es decir, con las FPGAs se crean circuitos digitales que pueden ser reprogramados.

Para diseñar el circuito, se describe el diseño usando lenguajes de descripción de hardware (entre los más usados está el VHDL). A continuación, se sintetiza el código. Esto consiste en transformar el código VHDL en hardware real, donde el sintetizador decide qué elementos utilizar. Las sentencias concurrentes de VHDL se convierten en Bloques Lógicos Configurables (CLB), que están formados por una Look-Up Table (LUT), que son tablas de consulta, un multiplexor y un registro. Por último se realiza el emplazado y enrutado, donde se genera el bitstream que es un archivo que tiene la configuración interna de la FPGA [11].

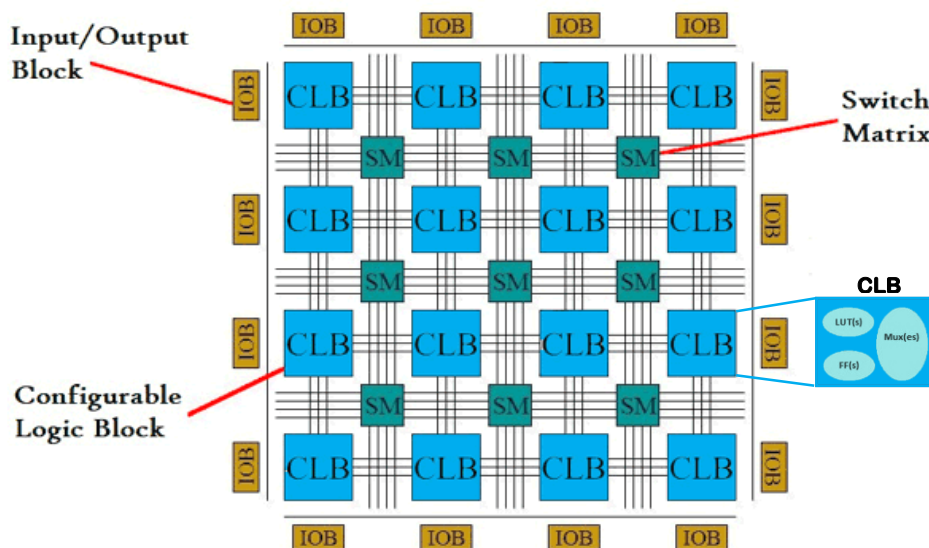


Figura 1.3: Estructura interna de una FPGA

Por último, se muestra una tabla comparativa entre FPGA y los microcontroladores (Arduino):

Tabla 1.1: Comparación entre Arduino y FPGA

	Arduino	FPGA
<i>Diseño</i>	Software	Hardware
<i>Complejidad</i>	Aceptable	Complicado
<i>Flexibilidad</i>	Media, limitado a estructura	Alta
<i>Trabajo</i>	Secuencial	Concurrente

Los microcontroladores como Arduino son una buena solución para la mayoría de proyectos ya que son más baratos y es más sencilla su programación, pero en caso de necesitar recibir señales de alta velocidad o procesar en tiempo real, es mejor el uso de FPGA por su forma de trabajo concurrente.

1.2. Cintas transportadoras

1.2.1. Descripción de su uso

Las cintas transportadoras son equipos auxiliares formados por componentes mecánicos y electrónicos que se usan en la industria para transportar, realizar cambios de nivel (ascender o descender) o para distribuir materiales. También hay cintas transportadoras para el transporte de personas. Son capaces de cubrir grandes distancias y su gran capacidad para adaptarse al terreno hace que sean los elementos utilizados en los procesos agroindustriales y mineros. Además aparecen en cualquier línea de producción automatizada por poder transportar el material en forma continua [12] [13] [14].

1.2.2. Elementos que lo componen

- **Banda.** Es la superficie en movimiento sobre la que se soporta el material. En función del uso que se le dé a la banda transportadora estará compuesta de distintos materiales para que tenga mejor agarre, si el material es abrasivo o no, si el material genera polvo o no...
- **Tambores de accionamiento.** Hay dos tipos de tambores:
 - **Tambores motrices.** Son los encargados de transmitir el movimiento a la banda gracias a la fricción entre la banda y el tambor. En una cinta transportadora puede haber sólo uno o pueden incluirse más si con uno no se alcanza a transmitir la potencia necesaria para el transporte del material.

- **Tambores no motrices.** Se encuentran varios tipos:
 - Tambor de reenvío. Se encarga del retorno de la banda, situada en el extremo opuesto al tambor motriz y suele tener su mismo diámetro.
 - Tambores tensores. Se encargan de conseguir un contacto adecuado entre la banda y el tambor motriz siendo capaz de absorber el estiramiento asociado a la banda. El estiramiento de una banda se mide por un coeficiente que debe facilitar el fabricante para evitar derramamientos de material [15]. Para cintas más largas de 40 metros [14], deberá ser un sistema de tensado por contrapeso mientras que para longitudes inferiores podrá ser un tensado manual. La cinta transportadora diseñada es pequeña y pretende transportar material liviano, por lo que no es necesario este elemento.
- **Rodillos.** Son los encargados del soporte de la banda, tanto en el tramo portante como en el tramo de retorno. Es importante que giren bien ya que si no aumentaría el rozamiento con la banda, incrementando el consumo de energía y el desgaste de la banda. Una cinta industrial debe tener soporte tanto en el tramo portante como en el tramo de retorno. La separación entre los rodillos del tramo portante depende de la anchura de la banda y de la densidad del material a transportar. La distancia entre los rodillos del tramo de retorno se calcula en función de los del tramo portante siendo normalmente entre 2 o 3 veces superiores a la distancia los primeros, ya que únicamente tienen que soportar el peso de la cinta. Otra opción más barata a los rodillos, cuando el peso del material a transportar no es muy elevado (para no aumentar la fricción demasiado), es el deslizamiento sobre cuna continua. En este tipo de soporte la banda se desliza sobre una superficie continua y los rodillos de retorno suelen tener una distancia entre 2 y 3 metros [15]. Para este proyecto al ser la cinta transportadora tan corta y el material a transportar tan liviano, no es necesario el uso de soportes. En caso de querer hacer una cinta transportadora más larga habrá que plantearse la introducción de este elemento.
- **Grupo motriz.** Es el encargado de proporcionar la potencia a los tambores para conseguir el movimiento de la banda. Básicamente está compuesto por el motor aunque a veces las cintas pueden incluir freno.
- **Estructura soporte.** Es todos los componentes que soporten la banda y el resto de elementos de la cinta transportadora.

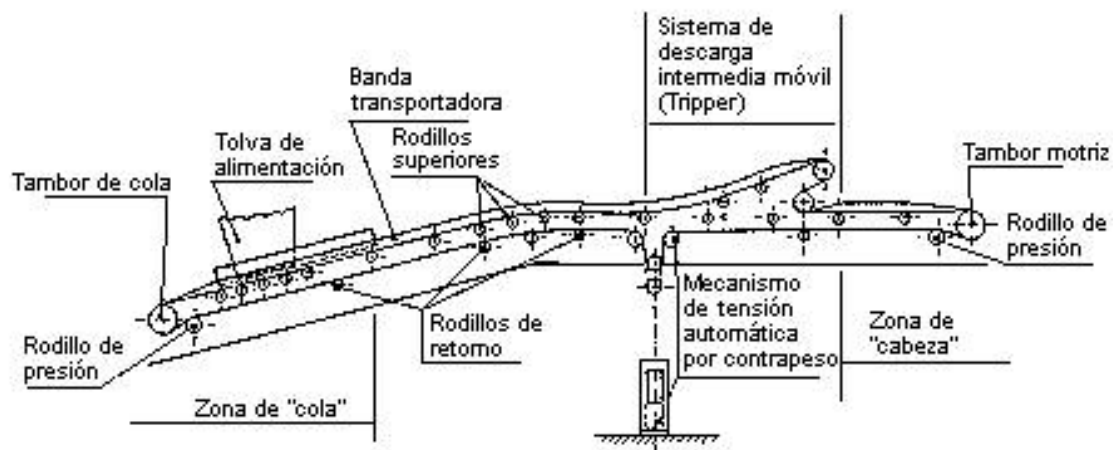


Figura 1.4: Esquema de una cinta transportadora [16]

1.2.3. Tipos de cintas transportadoras

Existen varias formas de clasificar las cintas transportadoras:

- En función del peso transportado [17]:
 - Cinta transportadora de uso pesado: se usan en industria para transporte de varias toneladas de material por hora.
 - Cinta transportadora de uso ligero: son usadas, por ejemplo, en supermercados para el arrastre de la mercancía.
- En función de su posición:
 - Horizontal
 - Inclínada
- En función del material del que está compuesto la banda: dado que la banda es el elemento principal de las cintas transportadoras y el material de ésta influye en el material que pueda transportarse, es una forma habitual de clasificación. Pueden ser de caucho, termoplásticas, modulares... [18]

1.2.4. Estado del arte: cintas transportadoras

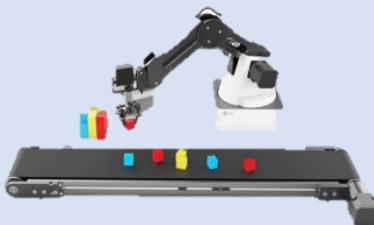

Cintas transportadoras industriales

Varía su precio en función del ancho de la banda, del tamaño de la propia cinta transportadora, de la potencia que desarrolle y con ello la cantidad de material que son capaces de transportar por hora y de su longitud, entre otras características. Hoy en día existen cintas de hasta 30km de longitud y capaces de transportar hasta 50.000 T/h [13]. Pero a nivel industrial también hay cintas transportadoras de uso ligero, usadas en algunas líneas de producción. Por tanto en función de la aplicación para la que se necesite la cinta transportadora industrial, puede variar mucho el precio para que cumpla con las necesidades. Así mismo, dependiendo del tipo de instalación que se realice, los precios podrían variar pudiendo alcanzar precios en instalaciones de medio millón de euros.

Cintas transportadoras comerciales

Las cintas transportadoras de supermercados, las cintas de correr, así como cualquiera de uso ligero a la que se le dé un uso no industrial están en la categoría de cintas comerciales. Se puede ver a continuación los siguientes precios:

Tabla 1.2: Cintas transportadoras comerciales

Cinta	Imagen	Velocidad	Tamaño (WxL) (mm)	Precio
Conveyor Belt Kit (no incluye el brazo robótico) [19]		120 mm/s	215 x 700	499 \$ → 456,08 € + transporte, aduanas, impuestos
Cinta transportadora [20]		417 mm /s	200 x 1.500	356,99 €
Cinta de correr [21]		14 km/h → 3889 mm/s	400 x 1230	271,20 €

Se aprecia que los precios pueden variar bastante en función de la aplicación a la que esté destinada la cinta.

Cintas transportadoras DIY

Para realizar este trabajo se han buscado otros proyectos en los que ha realizado el montaje de una cinta transportadora. De forma general se ha visto que no suelen explicar el control electrónico que tienen, sino que se centran más en el montaje mecánico. Además, en ninguno de los proyectos se hace un análisis de costes. Por tanto, la única comparativa de precios entre una cinta comercial y una cinta DIY será a partir del presupuesto que se haga de este trabajo, aunque resulta evidente que son más baratas que las cintas comerciales.

Tabla 1.3: Cintas transportadoras DIY

Cinta	Imagen	Características
Modular and portable conveyor belt [22]		En este modelo se realiza una cinta de un tamaño pequeño en el que deciden introducir un rodillo intermedio para la sujeción de la cinta. Además se controla la velocidad y el sentido con un motor paso a paso.
DIY conveyor belt [23]		Este proyecto trabaja con una cinta más larga por lo que deciden usar varios rodillos para el soporte de la banda, aunque se puede ver que para su longitud no usan rodillos en el tramo de retorno.
How to build a belt conveyor [24]		Al igual que el anterior proyecto, en éste se realiza una cinta bastante más grande que la del primer proyecto, por lo que es necesario un soporte de la cinta. En este caso en vez de rodillos usan una cuna continua.

En este proyecto se realiza una cinta transportadora con dimensiones más acordes a las del primer proyecto visto en la Tabla 1.3. Por esa razón en el diseño mecánico de nuestra cinta no se incluirán rodillos portantes.

2. OBJETIVOS

El proyecto consiste en el diseño y fabricación de una cinta transportadora, pudiendo variar su velocidad y sentido de giro. En su desarrollo se quiere cumplir una serie de objetivos:

- Comparar las prestaciones de Arduino frente a las de la FPGA para el control de la cinta transportadora. Ver las ventajas e inconvenientes de cada uno y aprender a elegir cuál usar para el control de un sistema mecatrónico en función del tipo de control que se requiera hacer y de los elementos de los que se disponga.
- Diseñar la electrónica de potencia para el control de motores en vez de adquirir una tarjeta de control comercial. Con esto se aprende a usar las hojas de características de los componentes al trabajar con circuitos analógicos en vez de con integrados.
- Utilizar componentes reutilizados. La basura electrónica cada vez es mayor, afectando al medio ambiente, por lo que su reutilización permite un desarrollo más sostenible [25].
- Aprender a realizar ingeniería inversa de los componentes reutilizados.
- Realizar un proyecto abierto, comprometiéndose con la filosofía de hardware libre. Para ello se usan programas de libre acceso y el proyecto está accesible en la plataforma de desarrollo colaborativo de Github.
- Conseguir un bajo coste, lo cual se consigue cumpliendo los dos objetivos anteriores, es decir, usando piezas recicladas, trabajando de forma abierta y usando la impresión 3D. Diseñando piezas se puede usar material reciclado y acoplarlo al montaje haciendo el proyecto más económico.
- Dejar el diseño abierto a modificaciones y ampliaciones para que el que quiera pueda replicarlo.

La programación del proyecto para lograr los objetivos está reflejada en un diagrama de Gantt en el *Anejo II*.

3. SOLUCIÓN TÉCNICA

3.1. Análisis del prototipo

En la introducción se ha visto un pequeño resumen sobre las cintas transportadoras y la electrónica de control que nos servirá para entender el sistema mecatrónico de este proyecto.

En la siguiente figura se ve cómo ha quedado el montaje:

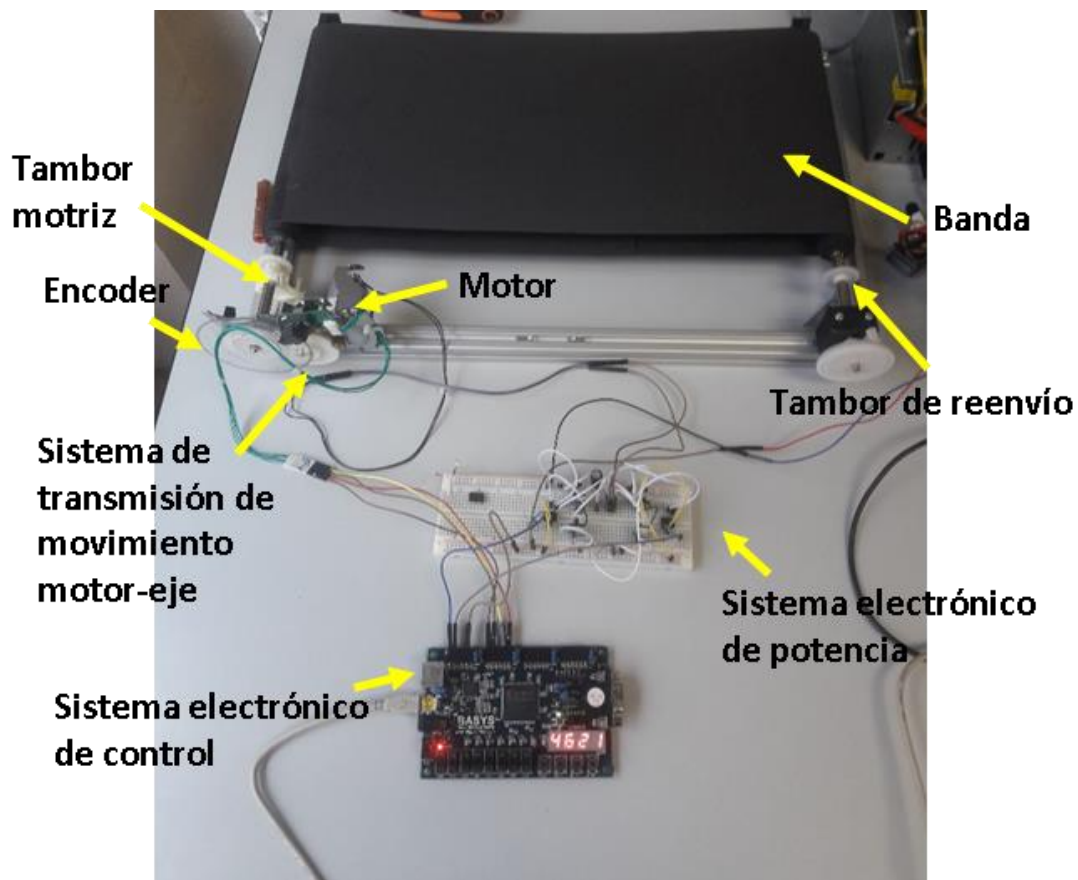


Figura 3.1: Imagen del montaje de la cinta transportadora

Como ya se ha comentado, en este trabajo se hace una comparación entre el control con una FPGA y Arduino. Se verá la comparación y el estudio del control en el apartado de 3.4. *Sistema electrónico de control*.

En este capítulo se pretende hacer una pequeña explicación de la versión más compleja desarrollada para entender las partes que tiene la cinta transportadora.

La Figura 3.2 muestra el esquema de la cinta transportadora, donde se ven las partes del sistema mecatrónico y sus interacciones. Este esquema está basado en la Figura 1.2.

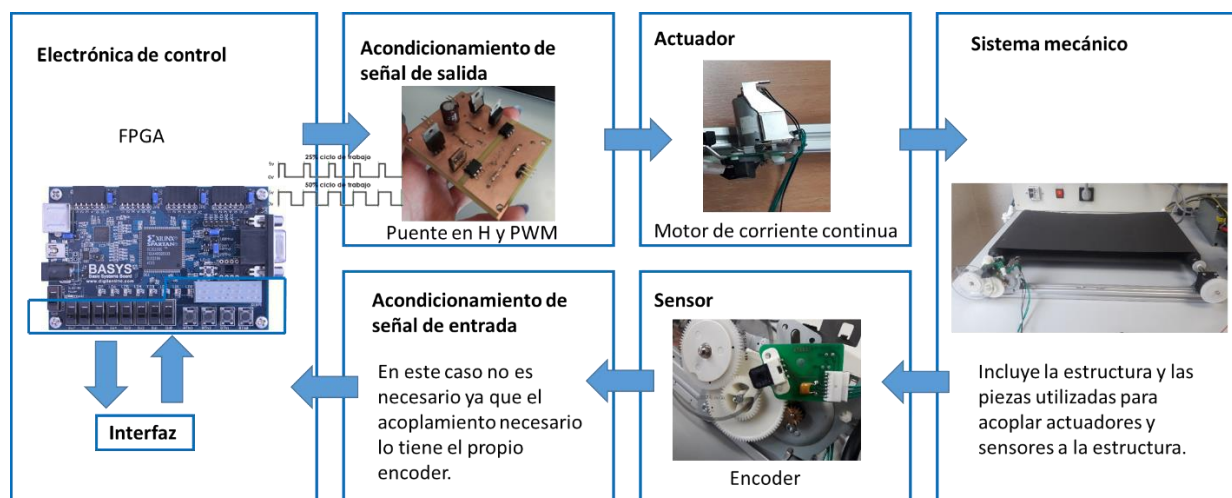


Figura 3.2: Esquema de las partes del sistema mecatrónico de la banda transportadora

A continuación se resume cada una de las partes del sistema:

Electrónica de control. La FPGA (o alternatively Arduino) es la que se encarga del control del sistema. De esta manera, da órdenes sobre los actuadores. En este caso, el controlador da una orden de velocidad al motor, el actuador de este sistema. Pero como la FPGA es un dispositivo electrónico digital, cuyas salidas valen 0 o 1 (0 o 3,3V; 0 o 5V para Arduino) hay que acondicionar estas salidas para que se transformen en una tensión y corriente adecuadas que muevan el motor a la velocidad deseada.

Acondicionamiento de señal de salida. Tiene varias etapas:

- Pulse Width Modulation (PWM), en castellano, modulación por ancho de pulsos: controla la velocidad transformando una señal digital en otra muy similar a una analógica.
- Optoacoplador: sirve para aislar y proteger el sistema de control del de potencia.
- Circuito puente en H: con él se controla el sentido de giro y proporciona la potencia necesaria al motor.

Actuador. El actuador es un motor recuperado de una impresora Deskjet 845c del que se puede controlar su velocidad y su sentido de giro. Por ser un motor de corriente continua, la velocidad se controla por tensión y el par por corriente.

Sistema mecánico. El motor, acoplándose correctamente con el sistema mecánico, consigue que el tambor motriz gire. Una de las características de los motores de corriente continua es que alcanzan mucha velocidad pero poco par. El par es el momento de fuerza que aplica un motor sobre el eje de

transmisión de potencia. Para aumentar el par, se ha utilizado también el sistema de engranajes del motor de la impresora, que reduce la velocidad y aumenta el par [26] [27]. Por último, el tambor de reenvío se mueve solidariamente al estar unido al tambor motriz por la banda, consiguiendo un movimiento lineal a lo largo de la cinta, ambos recuperados también de la misma impresora.

Sensor. Es un encoder rotativo, recuperado de la impresora. Es un sensor óptico capaz de detectar movimiento mediante una rueda con líneas negras sobre una lámina transparente. Al girar de forma solidaria con los ejes, leerá más o menos líneas por unidad de tiempo, siendo el número de líneas leídas proporcional a la velocidad. Se tiene una explicación más amplia en el apartado 3.4.1. *Encoder rotativo.*

Acondicionamiento de señal de entrada. En este caso no es necesaria porque se ha comprobado que el encoder reutilizado ya tiene acondicionada su señal, *Anejo V.*

Electrónica de control. Al controlador le llega la señal de lectura de línea, por lo que contándolas, puede usarlo para saber si realmente la orden de la velocidad que da sobre el actuador se está cumpliendo. A esto se le denomina sistema de retroalimentación.

Interfaz. La interacción entre el usuario y el sistema mecatrónico, se da a través de la FPGA, pudiendo el usuario cambiar la velocidad y el sentido de giro con los botones e interruptores y pudiendo obtener información del sistema a través del display, como por ejemplo la velocidad.

A continuación, se va a explicar los apartados en los que se ha dividido el trabajo y su contenido:

- **Sistema mecánico.** Incluye la estructura de la cinta transportadora y las piezas diseñadas para poder acoplar el motor y el sensor.
- **Sistema electrónico de control.** Como ya se ha mencionado, para un proyecto como este, se puede usar un microcontrolador como Arduino o una FPGA. En este trabajo se utilizan ambas, ya que se pretende implementar distintos tipos de control y así poder ver cuál de los dos es mejor usar en función del control que se quiera realizar. Para poder explicar los distintos tipos de control se incluye en este apartado la explicación del encoder.
- **Sistema electrónico de potencia.** Es el encargado de hacer el manejo de energía del sistema y es donde se explicará el circuito del puente en H y el PWM.

3.2. Sistema mecánico

En este apartado se describe, en primer lugar, los elementos que tiene la cinta transportadora realizada. A continuación, se ve el diseño de las piezas usadas para poder acoplar las distintas partes electromecánicas. En otro subapartado se trata la impresión 3D de estas piezas. Por último se tiene el ensamblaje final del sistema.

3.2.1. Elementos de la cinta transportadora

El sistema mecánico de la cinta transportadora no es complejo. En primer lugar, se analizan los elementos que tiene la cinta transportadora construida.

- **Banda.** Buscando materiales de bajo coste, se probaron bandas de dos materiales distintos. En un primer lugar se probó el papel de forrar adhesivo. El problema que surgió con este material es que en la zona de unión de ambos extremos, al tener un mayor grosor que en el resto de la banda, se frenaba por fricción al pasar por los rodillos, haciendo que no se mantuviera una velocidad constante. Finalmente se usó la goma EVA. Es un material barato, fácil de encontrar y resistente. Con este material se puede unir los extremos por el borde, de manera que el grosor en toda la banda es uniforme y evitando el problema anterior.
- **Tambores de accionamiento.** Se usaron como tambores de accionamiento los ejes extraídos de una impresora HP Deskjet 845c. Dado que uno de los ejes ya tenía incorporado el grupo motriz (el motor), es el tambor motriz y el otro eje el tambor de reenvío. No se dispone en la cinta de un sistema tensor ya que la longitud de la cinta no lo hace necesario. Al ser una banda tan corta se consigue un contacto adecuado y no es notable el estiramiento de la banda. En la siguiente imagen se ve cómo se extrajeron de la impresora:



Figura 3.3: Desmontaje de la impresora HP Deskjet845c

- **Rodillos.** Después de probar la cinta transportadora en funcionamiento se ha visto que, como los materiales transportados son ligeros y la cinta es corta, la banda no cede, por lo que no es necesario el uso de rodillos. En caso de alargar la cinta, habría que verificar si fueran necesarios.
- **Grupo motriz.** Incluye tanto el motor encargado de dar la potencia como el sistema de transmisión de movimiento desde el motor al eje. Por ejemplo, en el proyecto *Modular and portable conveyor belt* [22], se transmite el movimiento a través de una correa. En este proyecto, al usar el motor reciclado de la impresora HP Deskjet 845c, el motor ya venía acoplado con el sistema de transmisión y el eje que se usa de tambor motriz. El sistema de transmisión es un sistema de engranajes para conseguir un mayor par. Puede verse el sistema de transmisión en la Figura 3.4.



Figura 3.4: Vista del sistema de transmisión de movimiento del motor al eje por engranajes

- **Estructura soporte.** La estructura soporte incluye a todos los componentes que soportan la banda y el resto de elementos de la cinta transportadora. Por ello se incluyen los perfiles de aluminio usados en la estructura y las piezas utilizadas para poder acoplar los ejes y el sistema motriz a la estructura. Dado que los ejes son reciclados, para poder acoplarlos a la estructura, es más sencillo diseñar las piezas e imprimirlas que comprar unas piezas que se ajusten al modelo.

3.2.2. Piezas diseñadas

En este apartado se van a ver las piezas que se diseñaron para poder acoplar los ejes a la estructura.

Para el tambor motriz, en el extremo donde se encuentra el motor, se aprovecha la forma que tiene el esqueleto del sistema de transmisión para acoplarlo al perfil de aluminio. Esto facilita el enganche, ya que en un primer momento se diseñó una pieza para poder acoplarlo al eje, quedando el motor

colgando. Esto obligaba a diseñar otra pieza para sujetar el motor, ya que no es conveniente que quede en el aire. Además, para que cupiese el motor, habría que elevar a la cinta transportadora, gastando más material que no es realmente necesario. Puede verse cómo se fijó el eje en este extremo en la siguiente figura:

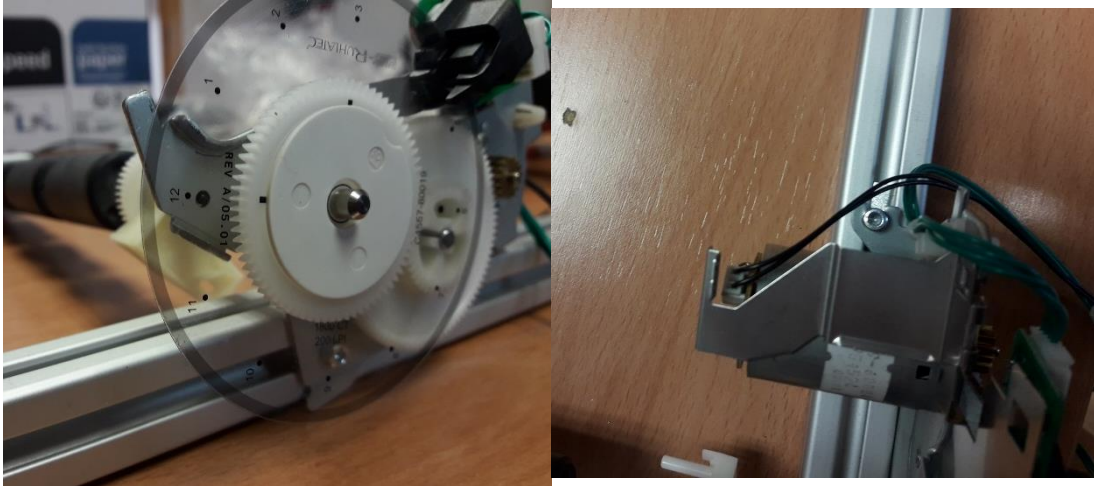


Figura 3.5: Vista de acoplamiento del tambor motriz en el extremo donde se encuentra el motor

En el extremo contrario de este eje se usa una pieza en la que se encaja el eje a presión. Hay que tener en cuenta que no puede sujetarse el eje fuertemente, ya que impediría el giro, pero tiene que fijarse de tal manera que el eje no oscile. Aprovechando una muesca que hay en el extremo del tambor, se diseña la pieza con un resalte para que el tambor quede enganchado y no se mueva a lo largo de su eje de giro pero permitiendo que pueda girar. A continuación se muestra la pieza diseñada:

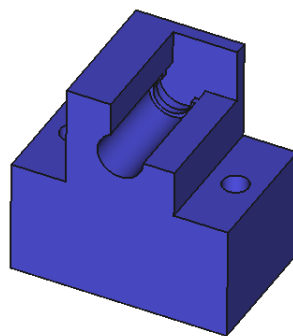


Figura 3.6: Soporte de ambos tambores en sus extremos libres

Esta pieza se usa tanto para el tambor de accionamiento como para el de reenvío en el extremo libre de los ejes, ya que ambos tienen el mismo diámetro y la misma muesca.

Al haber reciclado los ejes, se ha podido aprovechar el sistema de transmisión de movimiento pero hay que adaptarse a la forma y los elementos que no pueden desmontarse. Por ello, al tener ambos

ejes en un extremo engranajes que no se pueden desmontar, no se puede introducir ninguna pieza como se hizo con el extremo contrario. En este extremo hay que diseñar una pieza distinta en dos partes. En el tambor motriz no se usa ninguna pieza como se explicó, pero en el tambor de reenvío sí que se usa esta pieza que se muestra a continuación:

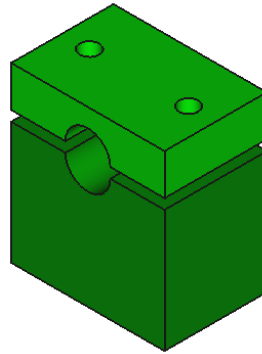


Figura 3.7: Soporte del tambor de reenvío en el extremo con engranaje

Otras opciones contempladas

Como se ha explicado, en un primer momento se pensó en hacer el acoplamiento del extremo del eje con el motor de una forma distinta que resultaba menos eficiente. Por ello se había diseñado una pieza para el extremo del conjunto del motor y encoder óptico que finalmente no se usa:

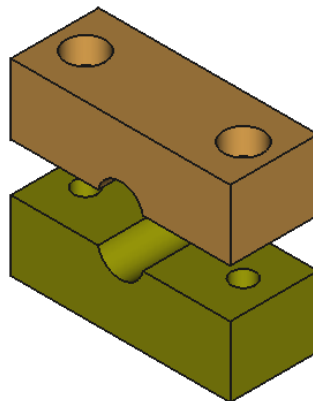


Figura 3.8: Soporte del tambor motriz en el extremo con el motor (no usada)

También se había diseñado una pieza distinta para los extremos libres de los ejes:

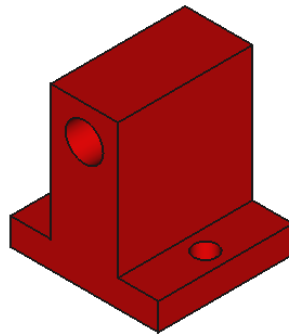


Figura 3.9: Soporte ambos tambores en sus extremos libres (no usada)

El problema de esta pieza, Figura 3.9, es que el tambor podía moverse a lo largo de su eje, no dando estabilidad a la cinta, lo cual se solucionó introduciendo la muesca en la pieza *Figura 3.6*.

El diseño de la nueva pieza, *Figura 3.6*, se hizo cuando todavía no se había cambiado la disposición del extremo donde se encontraba el motor. Esto provocó que al cambiarse, variara la altura. Se debe tener en cuenta que es importante que ambos ejes y a ambos lados tengan la misma altura para que no haya problemas con el giro de la banda. La altura aumentó en 12 mm. Dado que sólo se necesitaba aumentar la altura se aprovechó la pieza ya impresa y simplemente se imprimió un bloque para colocarlo debajo y darle la altura necesaria. Como se necesitaba por duplicado, la segunda se imprimió directamente con la altura necesaria.

3.2.3. Impresión 3D

En el momento de la impresión es necesario prever los posibles problemas.

- **Disposición de las mallas**

La posición en que se imprime la pieza es relevante. Un error común es imprimirla en la posición en que se va a usar, pero esta no tiene por qué ser la adecuada. Si la base es la zona con menos material, al depositar el material caliente puede que éste caiga. Por tanto al imprimir, debe ponerse en la base la zona con más material. En la *Figura 3.10* se ve un ejemplo:

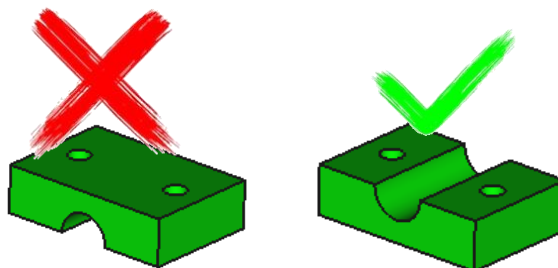


Figura 3.10: Cómo se debe y no se debe imprimir una pieza en 3D

- **Problemas en la impresión**

Surgieron algunos problemas una vez se tenían las piezas impresas.

Por una parte, hubo problemas con la **tolerancia** de las piezas. Todas las tolerancias se definieron en 0,4 mm, pero en la pieza de la Figura 3.6, no era suficiente. Como el eje tiene que entrar a presión, esa tolerancia no era bastante para que pudiera pasar, por lo que fue necesario aumentar la tolerancia a 0,8 mm.

Otro problema que puede surgir imprimiendo, es que la pieza no se **fije bien a la plataforma de impresión**. Es necesario echar una laca fijadora para que la pieza quede bien fija a la base. Si no se fija adecuadamente, lo que sucede es que el inyector arrastra la pieza mientras se mueve, dando el siguiente resultado:

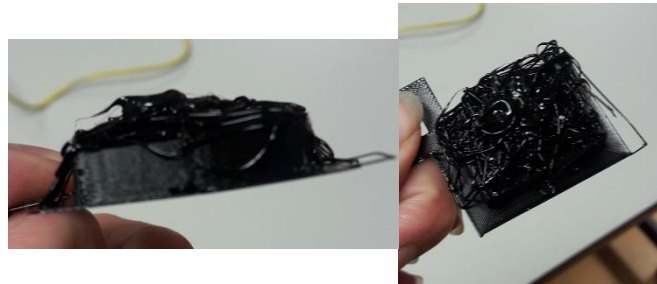


Figura 3.11: Problema de impresión al no fijarse bien la pieza a la base

3.2.4. Ensamblaje del sistema

Con las piezas impresas y los problemas solucionados, se monta la cinta transportadora, quedando de la siguiente manera.

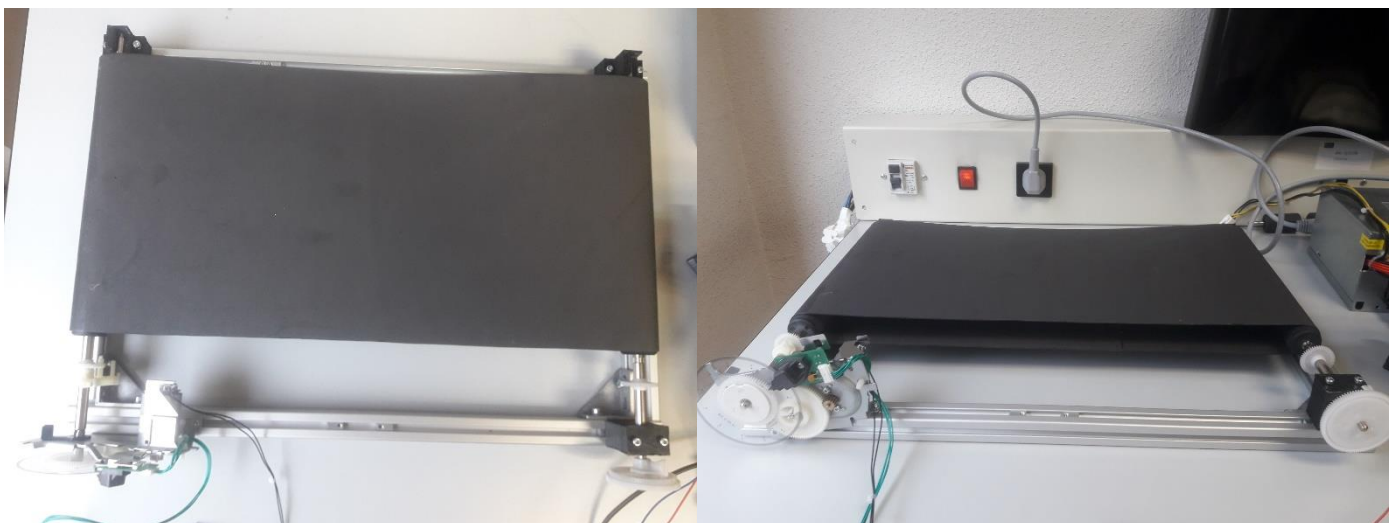


Figura 3.12: Ensamblaje del sistema

3.3. Sistema electrónico de potencia

Uno de los objetivos del proyecto es diseñar la electrónica de potencia para el control de motores, es decir, diseñar el circuito del puente en H. Esta forma de trabajo permite trabajar a un nivel de abstracción más bajo que si se usaran integrados: con circuitos analógicos. Es necesario por tanto controlar la intensidad y el voltaje en el circuito, usando las hojas de características de los componentes para elegir cuáles usar. Se hace el trabajo de ingeniería de elegir qué componentes utilizar en función de su rango de trabajo para evitar que se quemen o simplemente no funcionen. En conclusión, en este apartado se explica el funcionamiento del puente en H. La explicación de los componentes necesarios para montarlo, cómo se ha desarrollado su elección, cómo funcionan, por qué se usan, cómo conectarlos y cómo funciona de forma global el circuito se hace en el *Anejo III* y el *Anejo IV*. También se explica qué es la modulación por ancho de pulsos, que es un acondicionamiento de la señal para el motor para poder variar su velocidad.

3.3.1. Introducción

Se quiere controlar la velocidad y el sentido de giro del motor. Para acondicionar la señal de control del motor es necesario tener tres cosas en cuenta:

- En primer lugar, hay que resolver el problema de la conexión entre el controlador y el motor. Ni Arduino ni la FPGA son capaces de proporcionar la intensidad necesaria para el funcionamiento del motor.
- Para controlar la velocidad del motor, se debe variar la tensión. El problema surge en que los controladores son dispositivos digitales, por lo que la salida es o todo o nada: el motor parado o el motor a la máxima velocidad.
- Por último, el sentido de giro del motor depende del sentido de la corriente. Para conseguir controlar el sentido de giro habrá que invertir el sentido de la corriente, o lo que es equivalente, invertir la polaridad de la tensión.

3.3.2. Proporcionar corriente: transistores y optoacopladores

Los dispositivos de control (tanto Arduino como la FPGA) no son capaces de proporcionar la intensidad que se necesita para que un motor funcione. Por tanto, la conexión del circuito del motor con los dispositivos de control suele hacerse a través de un transistor, que suministra la corriente que necesita el motor, permitiendo tener una intensidad baja en la rama que conecta con el controlador y una intensidad más alta en la rama del motor.

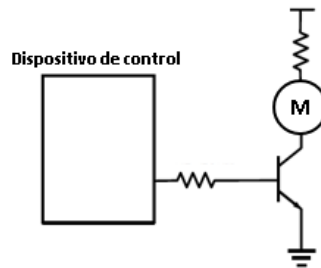


Figura 3.13: Conexión de dispositivo de control con un motor

3.3.3. Control de velocidad: Pulse Width Modulation (PWM)

La velocidad de un motor se controla con tensión. A mayor tensión mayor velocidad. Por ello se necesita un método para poder cambiar gradualmente la tensión que se le pasa al motor. Para ello se usa la modulación por ancho de pulsos (PWM).

El PWM simula una señal analógica usando pulsos digitales. Un ejemplo sería la intensidad a la que luce un led. Con una señal digital, únicamente se podría encender a la máxima intensidad o tenerlo apagado. Pero si lo que se hace es encender y apagar tan rápidamente la señal digital que el ojo humano no aprecie ese cambio, lo que se consigue es que el led brille a una intensidad distinta de la máxima. En función de cuánto tiempo se tenga encendido y cuánto apagado se podrá regular la intensidad de ese led.

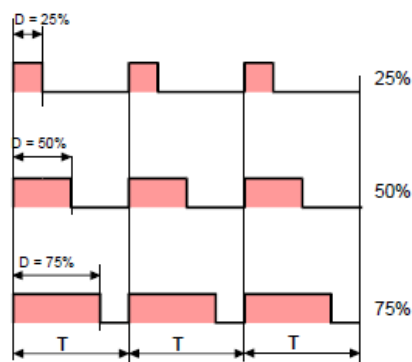


Figura 3.14: Señal PWM

Por tanto, una señal PWM funciona variando la proporción de pulsos por tiempo. El periodo, T en la Figura 3.14, es fijo, y debe ser suficientemente pequeño para no notar las variaciones que se darán. La D es lo que se denomina el ciclo de trabajo, y en función de su duración, que se puede cambiar, el valor medio del tren de pulsos será mayor o menor. Este valor medio es el valor que toma la salida, consiguiendo una regulación de la señal [28].

En Arduino esta función viene como un periférico del mismo, por lo que puede usarse fácilmente con la función `analogWrite`. En cambio, con la FPGA se tiene que crear un componente que implemente esta funcionalidad para poder usarla, por lo que es necesario entender cómo funciona para poder implementarla. Cabe mencionar que Arduino trabaja con un ciclo de trabajo que tiene 8 bits, por lo que puede variar los valores entre 0 y 255. En la FPGA se escoge implementar el PWM con el mismo número de bits.

3.3.4. Inversión del sentido: Puente en H

- *Por qué es necesario el uso de este circuito*

El motor usado para nuestra cinta transportadora es uno de corriente continua reciclado de una impresora. Podía haberse usado también un motor paso a paso, lo que haría más sencillo su control de velocidad (o posición) y de sentido frente al de corriente continua, del cual es difícil saber cuál es su velocidad o lo que realmente se ha desplazado. Es por eso que en impresoras de tinta, donde se necesita saber con precisión la posición de los ejes, suelen usarse motores paso a paso o motores de continua con un encoder. Los encoders actúan como sistemas de realimentación, de manera que leen las marcas que van pasando y con esto se puede conocer su posición y velocidad. Aunque, esto ya entra en el sistema de control, así que se hablará de ello más adelante [29].

En este caso fue reciclado de la impresora HP Deskjet 845c, por lo que se buscan las características tanto del motor como del encoder rotatorio.

Como el motor es de corriente continua, hay que invertir el sentido de la corriente para cambiar el sentido de giro del motor. Si solo se necesitara en un sentido concreto, bastaría con probar a conectar la fuente, ver el sentido en el que va, y si se quiere en el opuesto, desconectar la fuente y conectarla al revés. Pero si se quiere cambiar el sentido del motor durante su funcionamiento, esta forma no es ni eficaz ni práctica. Debe diseñarse un circuito denominado puente en H, que permite invertir la polaridad de la tensión en los bornes del motor [30].

- *Funcionamiento global del puente en H*

A continuación se muestra el esquema eléctrico del circuito desarrollado para el proyecto:

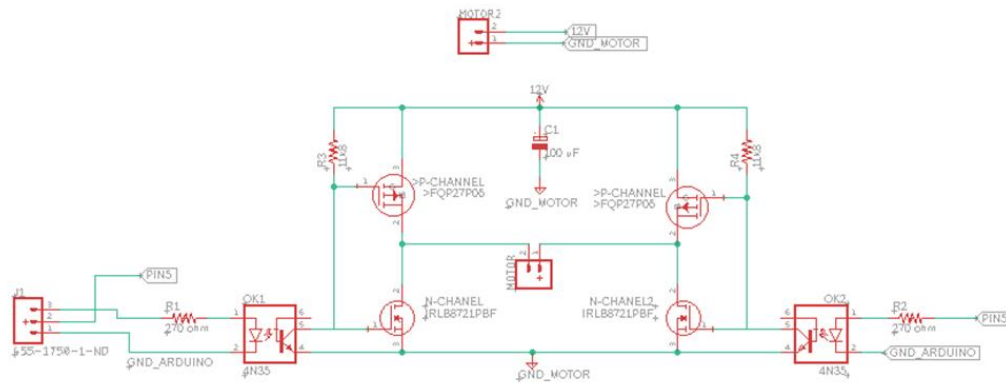


Figura 3.15: Característica V-I de salida de un mosfet

Como se ha comentado, para conseguir un cambio en el sentido de rotación del motor, es necesario invertir la polaridad de la tensión en los bornes del motor, para lo cual se usa un puente en H (ya que invertir la polaridad de la tensión es más complicado). Para realizar el puente en H básicamente la idea está en el uso de interruptores.

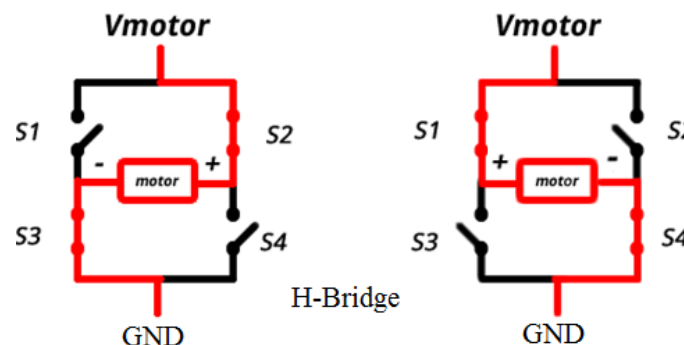


Figura 3.16: Puente en H con interruptores [30]

Como se puede ver, alternando la conmutación de interruptores, se puede conseguir la inversión de polaridad en bornes del motor y con ello cambiar el sentido de giro del motor sin cambiar la polaridad de la fuente de tensión. Pero, la inversión del giro con interruptores de forma manual no es práctico, por lo que se busca que ese cambio se haga mediante señales eléctricas. Para ello se sustituyen los interruptores por transistores, que pueden funcionar como interruptores controlados [27] [30].

En este proyecto se han usado dos optoacopladores. Lo que hacen es separar eléctricamente el circuito del puente en H del dispositivo de control. Esta separación eléctrica de ambos sistemas, permite que si hay un pico de corriente en el circuito de potencia (el puente en H), no llegue ese pico de corriente al sistema de control, evitando que se queme o se dañe. Como se verá en el apartado 3.5. Estudio

económico, el sistema de control es el más caro, por lo que es importante su protección, mientras que los optoacopladores son muy económicos.

En las figuras Figura 3.17, Figura 3.18, Figura 3.19 y Figura 3.20 se usa el esquema eléctrico para explicar cómo se invierte el giro en función de la señal que se mande a través del dispositivo de control. Se usa L para referirse a LOW, que significa que no hay tensión (y por tanto tampoco corriente), y se usa H para HIGH, correspondiéndose con que sí hay tensión, y en consecuencia, corriente. Los transistores se activan a distintas tensiones de puerta porque los de arriba son de canal p mientras que los de abajo son de canal n. Para una explicación más extensa, se puede ver el *Anejo III*.

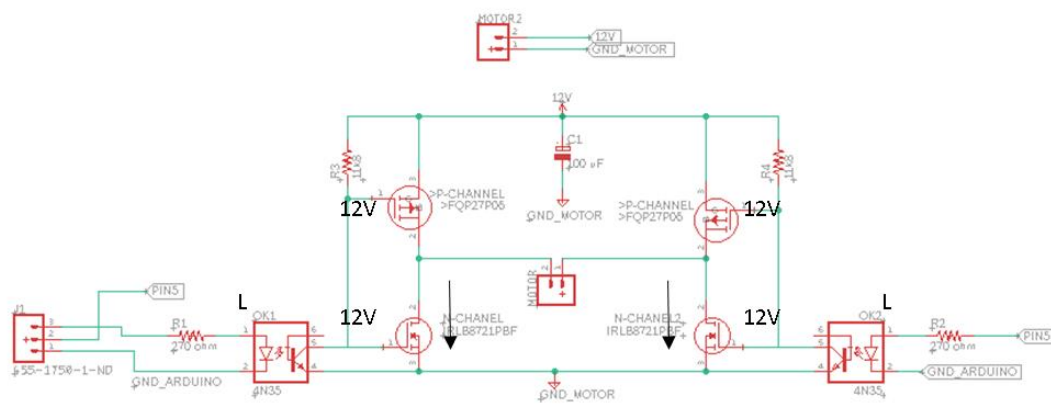


Figura 3.17: Esquema del funcionamiento eléctrico del circuito con entradas L-L

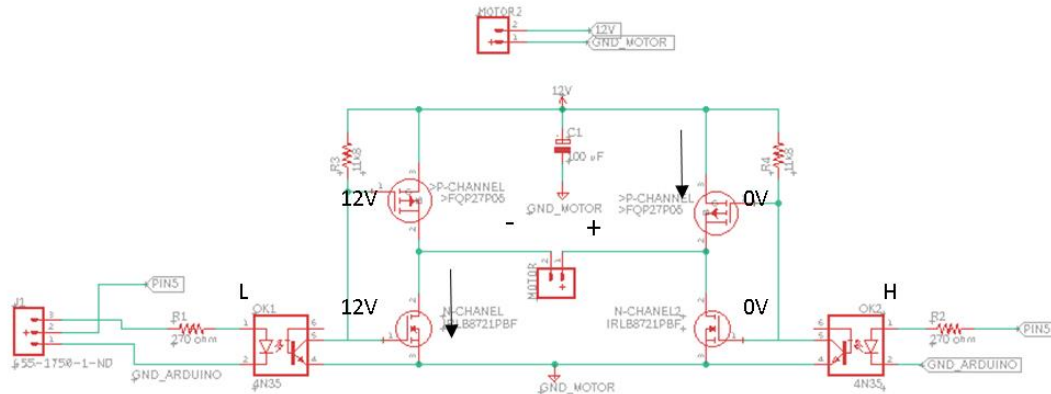


Figura 3.18: Esquema del funcionamiento eléctrico del circuito con entradas L-H

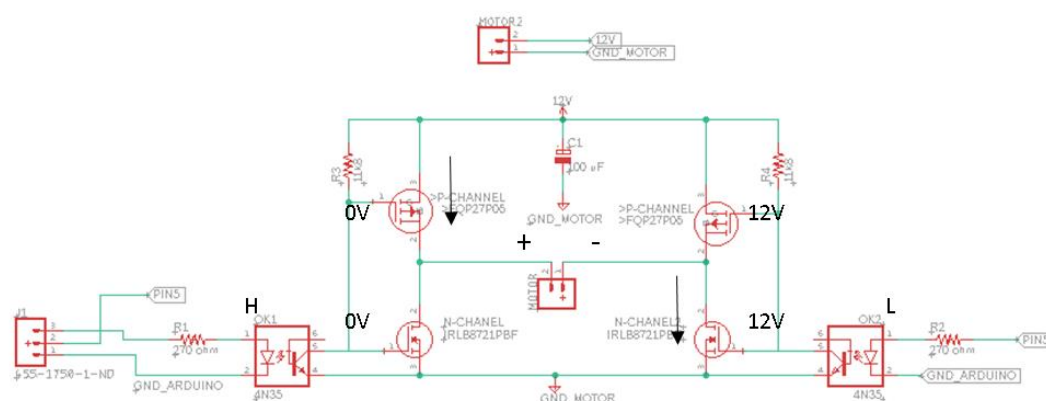


Figura 3.19: Esquema del funcionamiento eléctrico del circuito con entradas H-L

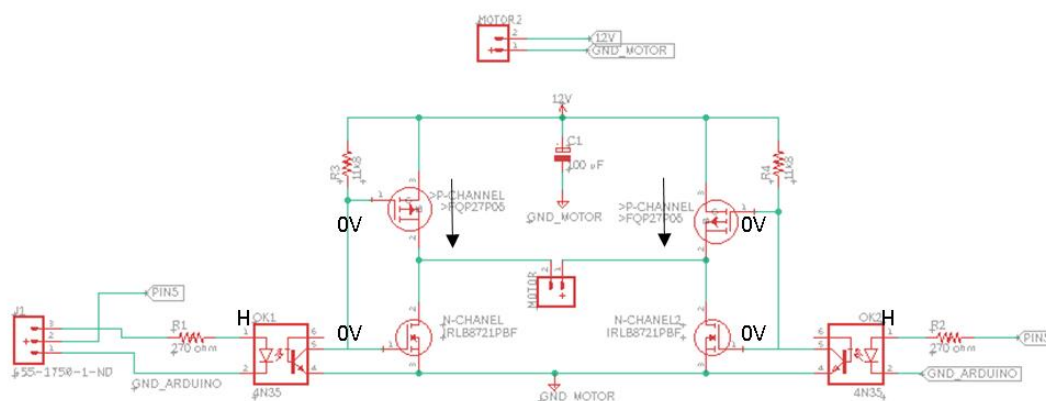


Figura 3.20: Esquema del funcionamiento eléctrico del circuito con entradas H-H

A continuación se hace una tabla resumen del funcionamiento del puente en H en función de las señales de entrada del dispositivo (que entran a través de los optoacopladores):

Tabla 3.1: Resumen del funcionamiento del circuito eléctrico en función de las entradas

Entrada		Salida	
Optoacoplador 1	Optoacoplador 2	Extremo 1 motor	Extremo 2 motor
L	L	Apagado	
L	H	L	H
H	L	H	L
H	H	Apagado	

Si se quiere comprender por qué con las entradas de estas señales a los optoacopladores los componentes responden de esta manera, se pueden consultar los anejos: *Anejo III* y *Anejo IV*. Las señales de entrada a los optoacopladores son reguladas por los dispositivos de control.

Con esta tabla del funcionamiento del circuito se puede implementar el control. No se usará la combinación de dar corriente a ambos optoacopladores para apagar el motor porque puede dar lugar a problemas con los mosfet.

3.3.5. Puente en H integrado

El montaje de este circuito no es demasiado complejo, pero sí que puede llegar a ser tedioso. Además, la inversión del giro de un motor de continua es una función que es necesaria en múltiples aplicaciones, por eso en el mercado existen integrados de este circuito. Podría haberse usado un chip con el integrado del puente en H, pero se hubiera perdido la comprensión del funcionamiento del circuito al trabajar en un nivel de abstracción más alto.

3.4. Sistema electrónico de control

En este trabajo se pretende hacer un estudio comparativo entre el control del sistema con Arduino y con FPGA. Para ello se explican los distintos tipos de control que se implementan y a continuación se verá cómo se han aplicado cada uno de ellos a los dos dispositivos. En este apartado se incluyen, además, las explicaciones necesarias para entender el encoder, ya que es el sensor que se usa para la retroalimentación.

Dado que con la FPGA se debe diseñar cada una de las funciones que se quieran utilizar, en el *Anejo VIII* se tiene cómo se han desarrollado y los códigos fuente se encuentran en el perfil de Github de la autora del proyecto [1].

3.4.1. Encoder rotativo

Como se ha comentado, el encoder rotativo es un sensor óptico que es capaz de detectar cambios de color. Su funcionamiento consiste en un emisor, generalmente un led infrarrojo que emite la radiación. El receptor suele ser un fototransistor que recibe más o menos radiación en función del tipo de superficie. Según la disposición del emisor respecto del receptor, se clasifican los encoders en reflexivos o transmisivos.

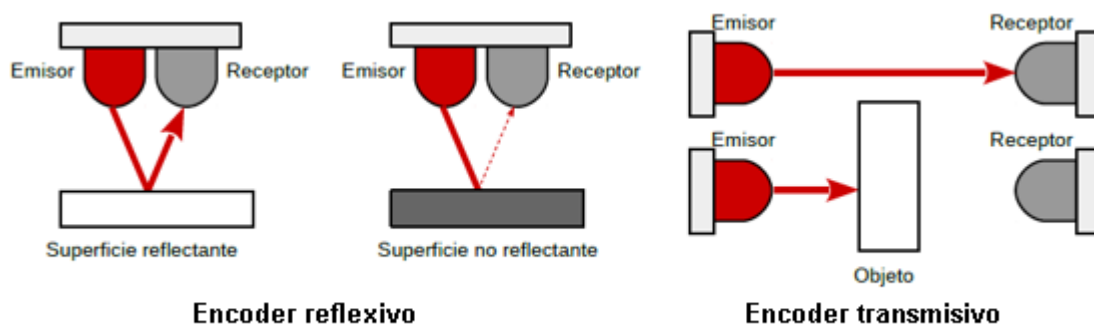


Figura 3.21: Tipos de encoder [Machado, F. Material del aula de mecatrónica, URJC]

Al desmontar el encoder de la impresora HP Deskjet 845c (Figura 3.22), se pudo ver que es un encoder transmisor, con una rueda que alterna líneas negras y transparentes. Una superficie de color negro, hará que la radiación que le llegue al receptor sea muy baja, mientras que si la superficie es transparente, le llegará al receptor una radiación alta. Esto permite detectar líneas cuando cambian de color. Y, dado que la rueda gira solidariamente con el eje, es decir, va a su misma velocidad, se puede calcular la velocidad a la que va la cinta con este sensor.



Figura 3.22: Encoder recuperado

Al ser un encoder recuperado, no se tenía la hoja de características y se necesitaba saber qué dos cables eran los sensores, cuál era alimentación y cuál era tierra. Para saberlo se hizo ingeniería inversa. Además, para poder hacer las conexiones eléctricas y el código, al no tener la hoja de características, fue necesario hacer una serie de pruebas que se encuentran en el *Anejo V*.

Cabe destacar que el encoder tiene dos sensores ópticos. Como se ha visto, con un sensor óptico se puede saber que la cinta se está moviendo, pero no el sentido en el que se mueve. Con el segundo sensor, desplazado respecto al primero, se puede saber el sentido dependiendo del primer color que detecte este último sensor. A este tipo de encoder se les denomina encoders de cuadratura. Para entenderlo se muestra la siguiente figura:



Figura 3.23: Funcionamiento de encoder con dos sensores en cuadratura para detectar el sentido de movimiento [Machado, F. Material del aula de mecatrónica, URJC]

Esto es lo que se implementa en el código (una vez sabida si la señal es analógica o digital, *Anejo V*), tanto en Arduino como en FPGA para detectar el movimiento y poder medir la velocidad.

3.4.2. Lazo abierto vs lazo cerrado

Dado que en el trabajo se hace un estudio comparativo entre Arduino y FPGA trabajando con distintos tipos de control, se va a explicar estos controles.

Un sistema de control se define por la conexión de los componentes que lo conforman para dar la respuesta deseada, encargándose de dirigir o regular otro sistema, un actuador. Se pueden encontrar dos formas de que se relacionen los componentes entre sí, dando lugar a dos tipos de control:

- **Lazo abierto**

Los sistemas de control suelen representarse a través de diagramas de bloques. En la *Figura 3.24* se puede ver el diagrama de bloques típico que tiene un sistema de control en lazo abierto.

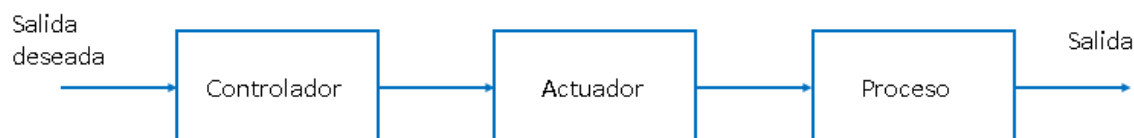


Figura 3.24: Diagrama de bloques del sistema de control en lazo abierto

El funcionamiento de estos sistemas consiste en comunicar al controlador la salida que se desea tener, de manera que maneje al actuador que interviene en el proceso dando una salida. En conclusión, la salida depende únicamente de la entrada [31]. Este tipo de control se usa por ejemplo en los tostadores eléctricos. La tostadora opera en una base de tiempo, de manera que no mide la salida, que es si realmente se ha tostado el pan, sólo lo calienta durante ese tiempo.

En este proyecto el sistema en lazo abierto quedaría de la siguiente manera:

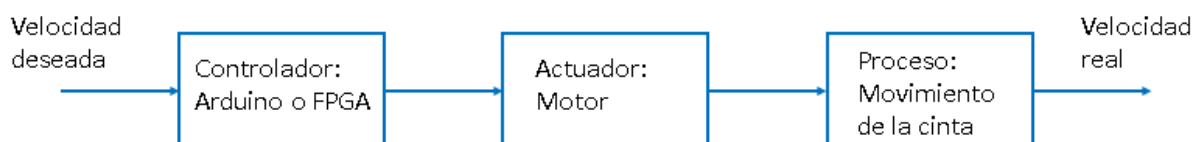


Figura 3.25: Diagrama de bloques del sistema de control en lazo abierto del proyecto

La velocidad deseada se le da como entrada al controlador a través del interfaz desarrollado para la comunicación entre el usuario y el dispositivo de control. Con esta información el controlador actúa

sobre el motor, dando lugar al movimiento de la cinta y teniendo como salida del sistema la velocidad a la que realmente se mueve el motor, que no tiene por qué ser la misma que la deseada.

- **Lazo cerrado**

En la siguiente figura se representa un sistema de control en lazo cerrado:

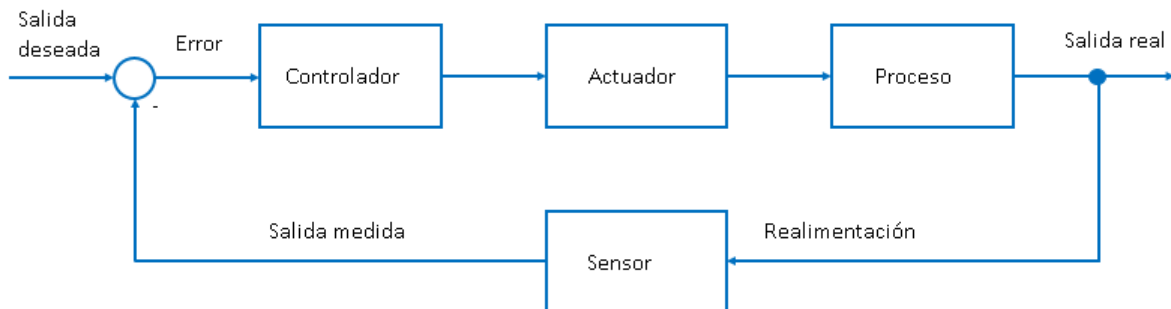


Figura 3.26: Diagrama de bloques del sistema de control en lazo cerrado

En este tipo de control, a diferencia del de lazo abierto, se mide la salida real mediante un sensor. Con esta información se realimenta la entrada y se compara con el valor que se desea tener a la salida. Así el controlador es capaz de ajustar la señal que manda al actuador en función del error que le haya llegado.

Esta forma de control tiene una serie de ventajas sobre el control con lazo abierto. Es capaz de corregir perturbaciones (como pueden ser el rozamiento de un motor o el peso de los objetos transportados), estabiliza procesos inestables y da tolerancia a variaciones en los parámetros, entre otras cosas [31].

A continuación se ve el sistema de control implementado en el proyecto para el trabajo en lazo cerrado:

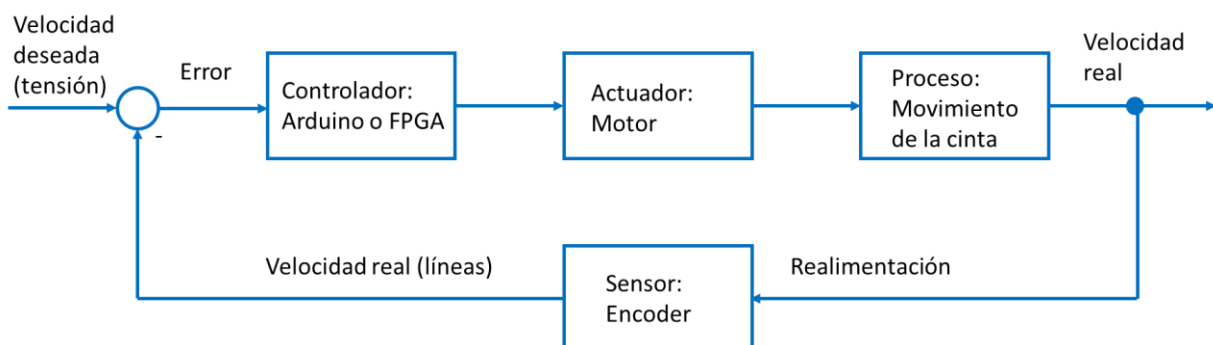


Figura 3.27: Diagrama de bloques del sistema de control en lazo cerrado del proyecto

En este proyecto se aplica este sistema de la siguiente manera:

El controlador recibe la velocidad a la que se quiere que funcione el motor. Este valor se da en tensión. Debido a una serie de agentes externos, puede que la velocidad real no sea la misma que la deseada. Por ello se mide la velocidad real por medio de un sensor, el encoder. Es necesario el uso del encoder porque no se puede conocer de otra forma la velocidad o desplazamiento de un motor de corriente continua. Como se explicó en el apartado 3.1. *Análisis del prototipo*, el encoder lo que hace es detectar las líneas que tiene en su rueda. Contando el número de líneas, se puede medir la velocidad real como el número de líneas que lea el encoder por unidad de tiempo.

Las unidades que se usan para enviar la velocidad deseada no son las mismas en las que se mide la velocidad real. Para comparar ambas magnitudes es necesario que estén en las mismas unidades. La conversión se explica en el *Anejo VI*. Con la información del error entre la velocidad deseada y la velocidad real, el controlador puede ajustar el valor en tensión que tiene que mandar al motor para que alcance la velocidad deseada.

En resumen, la velocidad de un motor se controla con el voltaje, pudiendo variarlo con el PWM. En lazo abierto únicamente se pasa una tensión a los bornes del motor, pero la velocidad a la que gire no tiene por qué ser la esperada, puede verse afectada, entre otras posibles razones, por:

- El rozamiento de la banda con los rodillos.
- La tensión de la banda.
- La carga a la que se vea sometida el motor, es decir, el peso de los objetos transportados en la cinta.

Por ejemplo, al aumentar la tensión de la banda, con un control en lazo abierto la cinta no iría a la misma velocidad antes de tensarla que después. La orden de voltaje es la misma en ambos casos pero no se alcanza la misma velocidad porque la resistencia al movimiento es mayor después de tensar la cinta. Esto provoca que la velocidad sea inferior a la que se tenía antes.

En cambio cuando se trabaja con el lazo cerrado, sí que se controla la velocidad gracias a la realimentación que provee el encoder y que permite ajustar la tensión de salida.

Continuando con el ejemplo anterior, en lazo cerrado la cinta iría a la misma velocidad antes y después de tensar la banda. La velocidad está determinada por el número de líneas que lee el encoder en un determinado tiempo. Al tensar más la banda, el sistema leerá menos líneas, por lo que el controlador enviará una mayor tensión para conseguir leer las mismas líneas. En conclusión, ante una velocidad

deseada, siempre se alcanzará la misma con el lazo cerrado, ya que es capaz de compensar eventuales pérdidas de velocidad.

3.4.3. Arduino

En este apartado se explica la implantación del control en lazo abierto y lazo cerrado con Arduino.

- **Lazo abierto**

La implementación de este tipo de control con Arduino no es complejo. El código está disponible en el enlace de Github de la autora del trabajo [1]. Con ese código se puede controlar el sentido de giro del motor con un botón y la velocidad de giro con un potenciómetro. A través del potenciómetro se da un valor de velocidad en tensión con valor entre 0 y 5V, que se pasa a digital entre 0 y 1023. Posteriormente se convierte a tensión por PWM con valores entre 0 y 255, correspondiendo los 255 a 12 V y 0 a 0 V. Funciona correctamente.

- **Lazo cerrado**

En primer lugar, se debe comprobar si Arduino es capaz de leer las líneas del encoder, ya que son 1800 y debe hacerlo a una velocidad muy alta. Para ello se hace un código de prueba en el que el motor tiene tensión hasta que se cuenten las 1800 líneas y se pare cuando las alcance. De esta manera, si cuenta menos líneas de las que debiera, se verá que da más de una vuelta, mientras que si es capaz de contar todas las líneas, dará exactamente una vuelta. Se comienza la prueba con velocidades pequeñas, ya que cuanto mayores sean las velocidades, mayor probabilidad tendrá Arduino de perder la cuenta de alguna línea. Además, lo que se busca es ver la velocidad a partir de la cual ya no es capaz de contar todas las líneas que tiene el encoder, por lo que es lógico comenzar a una velocidad pequeña e ir aumentándola para ver ese punto.

Se hicieron las siguientes pruebas (dado que se trabaja con PWM, se pasan valores que se transforman a voltaje, siendo 0 el mínimo, parado, y 255 el máximo, dándole una alimentación de 12V):

Tabla 3.2: Ensayo en Arduino para comprobar su capacidad para leer el encoder

Valor PWM	Tensión equivalente	Vueltas	% líneas perdidas
50	2,35	Motor no se mueve	
100	4,71	15	93,33%
75	3,53	8	87,50%
60	2,82	5	80,00%

No es necesario realizar más pruebas. Puede verse que no es capaz de leer todas las líneas que tiene el encoder y que, además, pierde la cuenta de muchas líneas. Puede que a una velocidad entre 50 (parado) y 60 sí sea capaz, pero son velocidades muy bajas y con muy poco margen. Para comprobar en qué grado no era capaz de leerlo, se hizo una medida del tiempo que tardaba en leerlo, explicado en el *Anejo VII*.

Por tanto, no se puede implementar el sistema de control de lazo cerrado con Arduino, porque es demasiado lento.

3.4.4. FPGA

En este apartado se explica la implantación del control en lazo abierto y lazo cerrado con FPGA.

- **Lazo abierto**

Para determinar la velocidad que se desea tener, se usan tres interruptores y dos botones. Los interruptores permiten cambiar el dígito que se quiere aumentar o disminuir: las unidades, decenas o centenas. Los botones permiten aumentar o disminuir, de uno en uno, el dígito elegido con el interruptor. Así por ejemplo, cuando se activa el interruptor que modifica las decenas, si se pulsa el botón situado más a la izquierda, se aumentará la velocidad en 10 unidades, mientras que si se pulsa al otro botón, se disminuirá la velocidad en 10 unidades. Para tener una confirmación visual de qué dígito se está cambiando, este parpadeará. Se muestra por el display la velocidad que se desea tener. Para el cambio de sentido se usa otro de los interruptores de la FPGA. En la *Figura 3.28* se muestra gráficamente lo explicado en este párrafo.

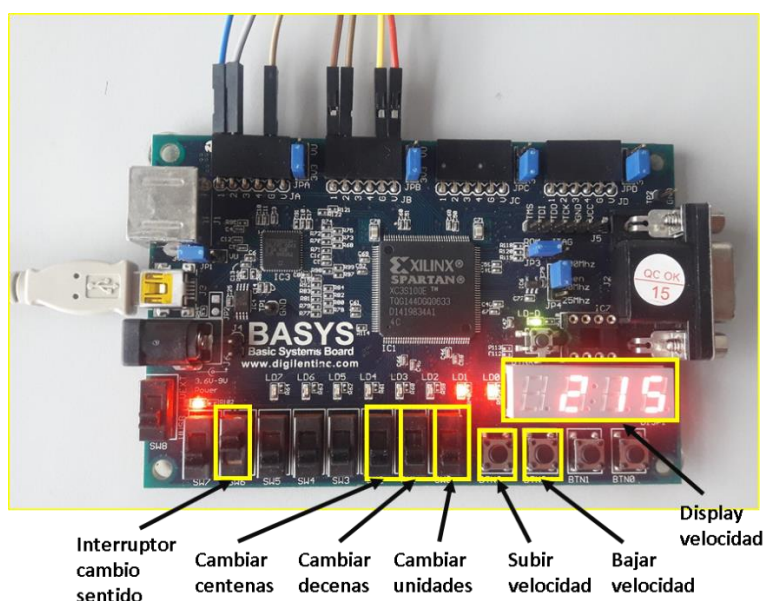


Figura 3.28: Interfaz FPGA

El código desarrollado para la FPGA es más complejo que el de Arduino, ya que hay que crearlo todo desde cero. Cuando se diseñan circuitos grandes es conveniente separarlos en bloques funcionales. Estos bloques se relacionan entre ellos a través de una arquitectura que los referencia. A esto se le denomina diseño estructural y es como se ha realizado el código. A continuación se muestra un esquema del diseño estructural desarrollado y se mencionan brevemente los bloques de la estructura, por qué son necesarios y cómo funcionan.

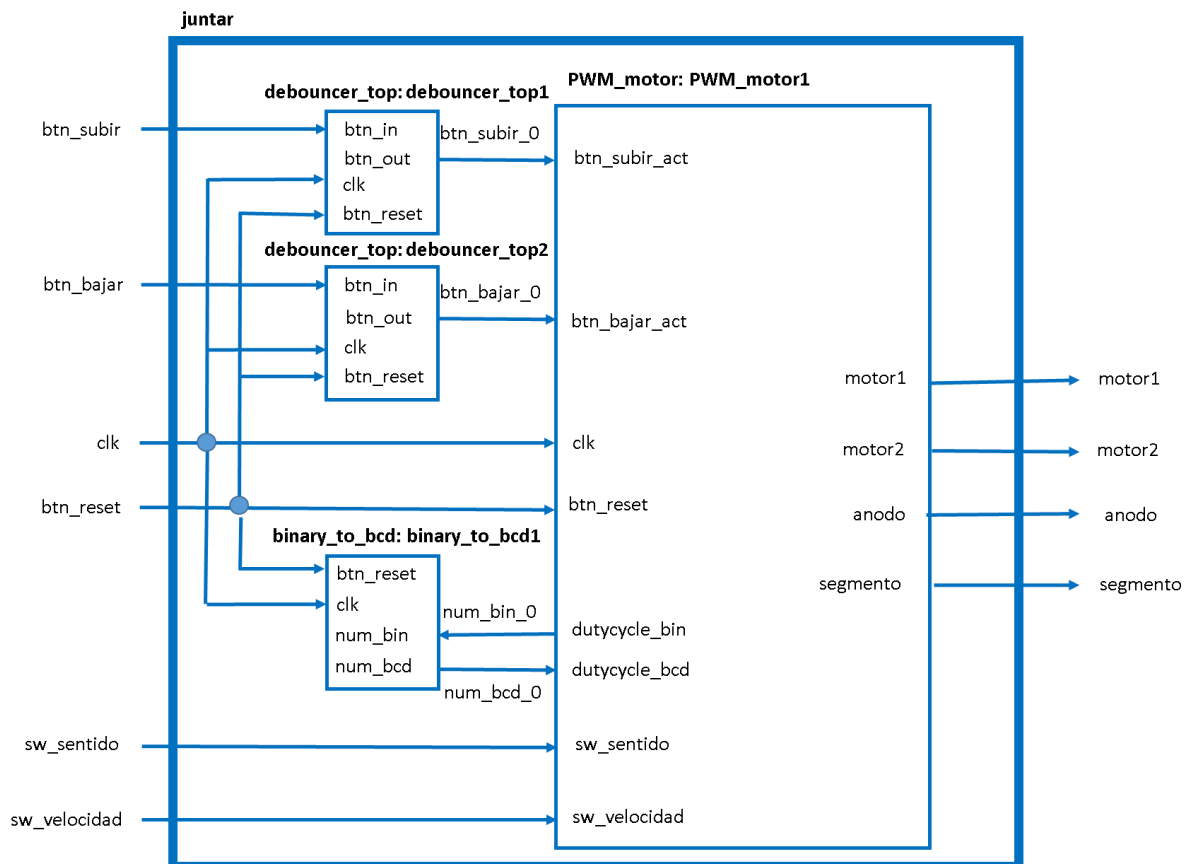


Figura 3.29: Diseño estructural del control de la cinta en lazo abierto en FPGA

○ juntar

Es el bloque que referencia y junta el resto de bloques. Esta forma de diseño permite dividir el circuito en partes más sencillas para poder entenderlo más fácilmente. Además, evita tener que repetir un bloque cuando se necesitan varios iguales, como, por ejemplo, en este diseño el bloque debouncer_top. Se necesita usar dos veces, pero basta diseñarlo una sola vez y describir dos referencias a él en el bloque juntar.

○ PWM_motor

En este bloque se crea el PWM, por lo que se controla la velocidad del motor, también se impone el cambio de sentido y se muestra la velocidad por el display. Se recuerda que la complejidad del diseño con FPGA radica en que se diseña el hardware, por lo que se tiene que trabajar con los cronogramas de las señales.

De esta forma, por ejemplo, al pulsar un botón, se espera que una pulsación se corresponda con un cambio. Sin embargo, en la práctica, se observan varios cambios. Así, por ejemplo, al pulsar el botón para aumentar la velocidad en una unidad, se observa que aumenta muchas más unidades. Esto es porque la frecuencia de reloj es muy alta respecto de la velocidad humana. Cuando una persona pulsa un botón, lo mantiene pulsado durante varios cientos de miles de ciclos de reloj por muy rápido que lo suelte. Por tanto, el controlador detectará que debe realizar tantos cambios como ciclos de reloj haya detectado que el botón estaba pulsado. Para evitar que suceda esto, se debe crear una señal que dure un solo ciclo de reloj cuando se detecte la pulsación del botón. Para ello se usan dos biestables. Los biestables son elementos capaces de guardar un valor. Esto que se realiza es lo que se llama un detector de flanco. Guardando los valores del botón, como se retrasa un ciclo de reloj, puede detectarse la secuencia 1 – 0, con lo que se conseguirá crear la señal que dure un solo ciclo de reloj. Para una explicación más avanzada y extensa, se puede recurrir a la bibliografía [32]. El cronograma de este detector de flanco puede verse en la Figura 3.30:

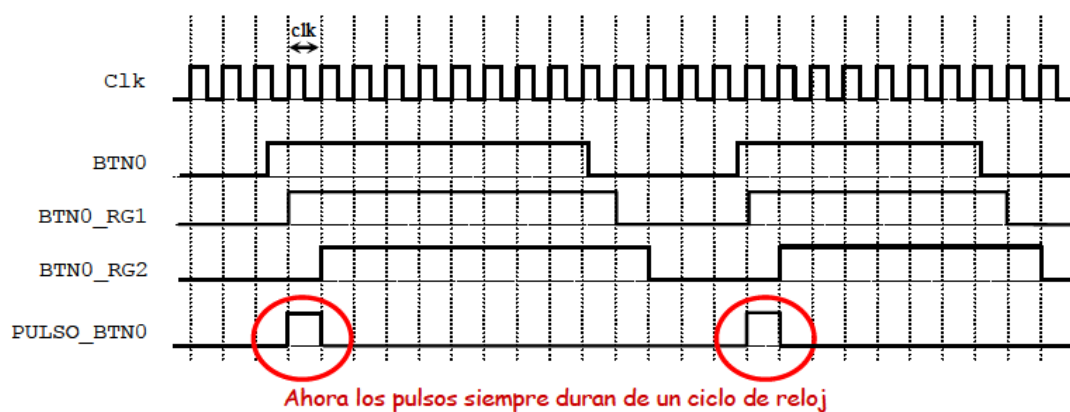


Figura 3.30: Cronograma de detector de flanco

Las explicaciones del diseño del PWM y de cómo mostrar la velocidad por el display son más complejas que la del detector de flanco, y dado que no son necesarias para comprender el conjunto del trabajo, se explican en el Anejo VIII.

- **debouncer_top**

Una vez realizado el programa se probó y se observó que pulsando una sola vez el botón se veían varios cambios a pesar de haber realizado el detector de flanco. Aquí se ve un reflejo de la complejidad de los sistemas mecatrónicos. La combinación de la electrónica y la mecánica complica el diseño, ya que hay que aprender a tratarlos conjuntamente. Lo que ocurría era lo que se denomina efecto rebote. Este fenómeno da lugar a que al pulsar el botón rebote mecánicamente, dando lugar a varios pasos por 0 y 1. Esto resulta en una señal que oscila durante un tiempo antes de estabilizarse, y como no es filtrada por el detector de flanco, se detectan varias pulsaciones del botón aunque no las haya. El proceso de eliminación de este rebote se denomina debounce. Consiste en ignorar la lectura del botón durante sus primeros instantes para detectarla cuando ya sea estable. En la Figura 3.31 se puede ver cómo es una señal afectada por el efecto rebote.

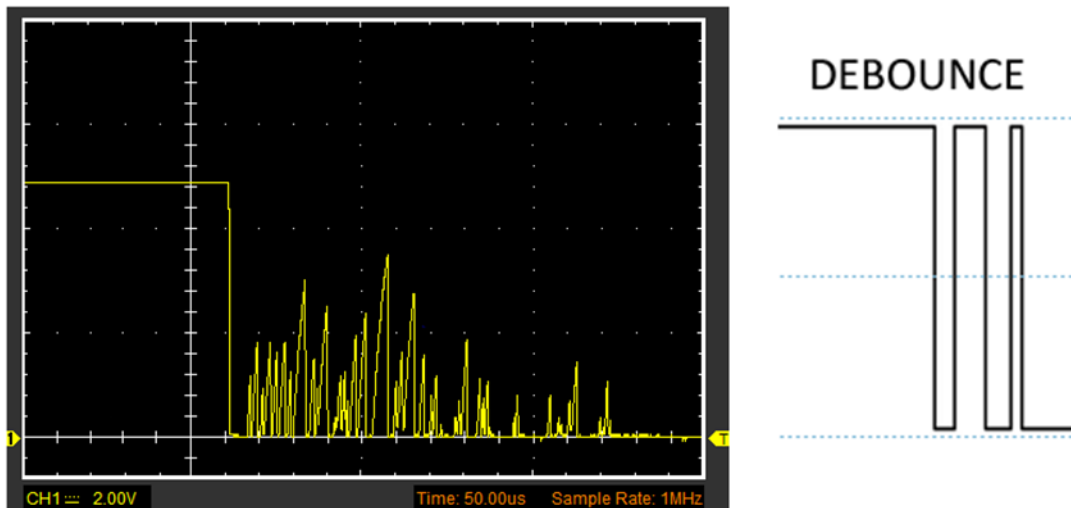
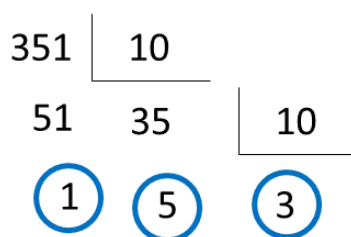


Figura 3.31: Señal afectada por efecto rebote

- **binary_to_bcd**

Como ya se ha comentado, con la FPGA se diseña hardware. Esto hace que operaciones tan comunes en software como una división, no se puedan realizar directamente en el lenguaje VHDL. Pueden realizarse divisiones entre 2 o potencias de 2, ya que consiste en el desplazamiento de bits, pero no se pueden hacer divisiones entre otros números.

Teniendo un número decimal de 3 dígitos, para mostrarlos por el display, se necesita conseguir los 3 dígitos por separado. De esta manera lo más lógico sería dividir entre 10 sucesivamente y los restos de las divisiones y el cociente de la última división daría los dígitos separados:



Pero, dado que no se pueden realizar divisiones, se debe buscar otro método. La codificación Binary-Coded Decimal (BCD) es una codificación que representa cada dígito decimal con cuatro bits. Esto permite poder identificar unidades, decenas y centenas. Por tanto, este bloque lo que hace es transformar el número, que se tiene en binario, en codificación BCD.

El código en lazo abierto con FPGA funciona correctamente, aunque su diseño ha sido mucho más complicado que el que se tuvo que hacer para Arduino.

- **Lazo cerrado**

En primer lugar, al igual que se hizo con Arduino, se debe comprobar si la FPGA es capaz de leer las líneas del encoder. Para ello se desarrolla un código en el que se cuentan las líneas que lee la FPGA, de manera que el motor se pare cuando alcance las 1800 líneas. Se observa que da una vuelta, por lo que sí que es capaz de leer todas las líneas. A continuación se realiza otro código en el que se hace que pare el motor después de dar 10 vueltas. Por una parte, se muestra por el display el número de líneas que lee para comprobar que lea 18000 y así hacer una segunda prueba que demuestre que es suficientemente rápido para leer el encoder. Por otra parte, sirve para medir el tiempo en que tarda en dar las 10 vueltas a la máxima velocidad que puede alcanzar. Con este dato se podrá hacer el cambio de unidades de velocidad real en barras a velocidad real en tensión (*Anejo VI*). Además, se puede calcular el tiempo que se necesita para leer una marca de rueda, y midiendo el tiempo que tarda Arduino en ejecutar todo su código, se comprueba, con datos numéricos por qué Arduino no es capaz de leer el encoder (*Anejo VII*). En conclusión, sí puede realizarse el control en lazo cerrado con la FPGA.

El interfaz entre usuario y controlador es el mismo que en lazo abierto con un añadido: se usa un interruptor para alternar lo que se ve en el display. En una posición se muestra la velocidad que fija el usuario, es decir, la velocidad deseada, mientras que en la otra posición se ve la velocidad real.

Para realizar el control en lazo cerrado fue necesario añadir dos bloques: un bloque denominado **PID** y otro que se llama **calcular_posicion**. Se muestra su diseño estructural (Figura 3.32) para poder

comprender cómo funciona y se explican los dos nuevos bloques. Los bloques **debouncer_top** y **binary_to_bcd** no se modificaron para este control. En cambio, el módulo **PWM_motor** sí que se modificó.

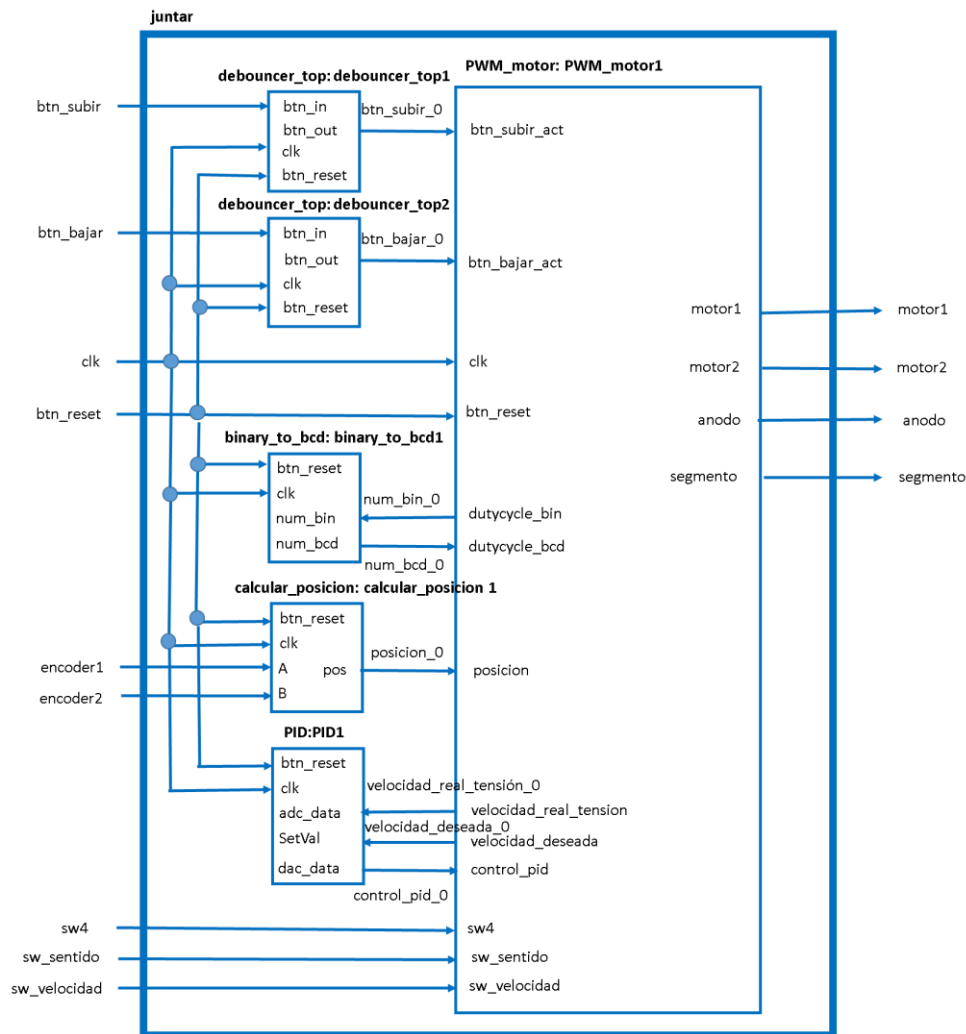


Figura 3.32: Diseño estructural del control de la cinta en lazo cerrado en FPGA

○ PWM_motor

En este bloque es necesario añadir el cálculo de la velocidad real en unidades de tensión, que es función del número de líneas leídas por el encoder, contadas en el bloque **calcular_posicion**.

- **calcular_posicion**

Recibe las señales del encoder y con ellas se contabiliza el número de líneas por unidad de tiempo. El bloque juntar hace que el dato de velocidad se lo pase al PWM_motor para poder calcular la velocidad real en unidades de tensión.

- **PID**

Es el bloque en el que se hace el control. Para ello necesita la información de la velocidad que se desea tener y la velocidad real en tensión. Aquí es donde se hace la comparación y se ve si hay un error. En función de ese error se calcula y envía la señal de tensión modificada al motor para alcanzar la velocidad deseada. A esta señal se le ha llamado control_pid.

En control se pueden hacer varias acciones para controlar el error y aproximar el valor que se tiene al que se desea tener:

- Acción proporcional: se encarga de disminuir el error en régimen permanente y estabilizarlo. Este tipo de control es el que siempre se introduce.
- Acción derivativa: se añade este tipo de control cuando hay demasiada oscilación en el régimen transitorio. Consigue disminuir esa oscilación añadiendo amortiguamiento. Actúa sobre la velocidad de cambio del error. Así, por ejemplo, en este proyecto una oscilación demasiado grande daría lugar a corrientes elevadas, pudiendo quemar el motor. Por tanto, será un punto crítico determinar si es necesario este control o no.
- Acción integral: disminuye o elimina el error en régimen permanente.

Para este proyecto se ha usado únicamente un control proporcional. Se ha comprobado que aunque haya oscilación, no es una oscilación peligrosa que pueda quemar el motor, y dado que alcanza rápidamente el régimen permanente con un error prácticamente nulo, no son necesarios ni el control integral ni el control derivativo.

Se prueba el control y se observa que funciona correctamente. Se comprueba frenando el motor a propósito. Como por el display se puede mostrar la velocidad real que a la que está yendo el motor, la velocidad que muestre debe ser la misma que se había fijado, aun cuando se está frenando el motor. De esta forma, se puede comprobar que el control está funcionando correctamente.

Cabe mencionar que se puede simular el programa en lo que se llama un banco de pruebas. En esta simulación se ven los cronogramas de las señales del programa y sirve para comprobar si se ha

realizado correctamente el diseño. Para este control se realizó el banco de pruebas, pudiendo ver el resultado de la simulación en la siguiente figura:

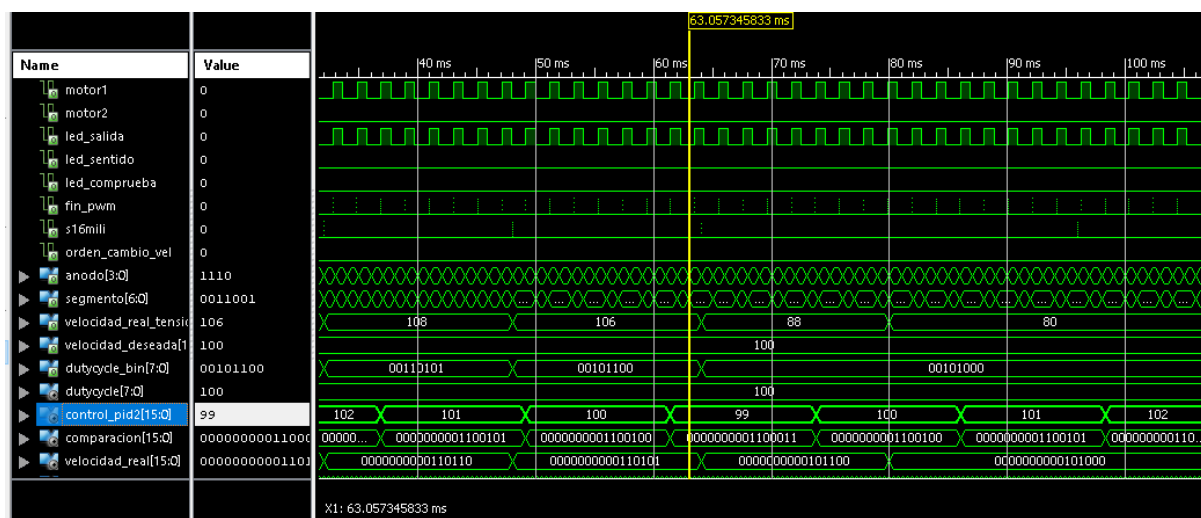


Figura 3.33: Simulación del programa de control en lazo cerrado con la FPGA

Con esto, se tienen los dos distintos tipos de control explicados (lazo abierto y lazo cerrado) e implementados tanto en Arduino como en FPGA.

3.4.5. Conclusiones del estudio comparativo entre Arduino y FPGA

A partir de este estudio se pueden extraer varias conclusiones.

El control en lazo abierto funciona correctamente, tanto en un microcontrolador (Arduino) como en la FPGA. Pero, como ha podido verse en este apartado, es mucho más sencillo el diseño del programa en Arduino que en FPGA. Por tanto, ante una igualdad de resultados experimentales, es mejor el uso de Arduino por su sencillez y precio.

En cuanto al control en lazo cerrado, es necesario el uso de un sensor externo para realizarlo, ya que si no, no se puede conocer la velocidad o desplazamiento de un motor de corriente continua. Se usa un encoder como realimentación. Se ha comprobado que no podía implementarse el control en lazo cerrado en Arduino porque el microcontrolador no tiene suficiente rapidez para poder leer las líneas del encoder. En cambio, la FPGA sí que es suficientemente rápida para leer el encoder, por lo que se ha podido realizar este control con la FPGA. Aparte que la frecuencia de reloj de la FPGA es mayor (de 100 MHz) que de la de Arduino (16 MHz), haciendo la FPGA más rápida de partida, hay otra razón que acelera la lectura de la FPGA: su forma de trabajo en paralelo. La forma de trabajo en Arduino o cualquier microcontrolador, es secuencial en vez de concurrente, provocando que otros procesos que se realizan en el programa, como el control del motor, ralenticen la lectura del encoder.

En conclusión, es mejor el uso de FPGA cuando se afronta un proyecto en el que se tienen señales de alta velocidad o procesamiento en tiempo real, como sucede con el encoder cuando se trabaja en lazo cerrado. Esto es debido a su capacidad para realizar procesos en paralelo. Para esta aplicación, como se ha visto anteriormente, ni si quiera puede usarse Arduino porque no es capaz de leer el encoder. En cambio, para el control en lazo abierto, que no necesita lectura de señales en tiempo real, es mejor el uso de microcontroladores como Arduino por ser más sencillos de programar y ser más baratos.

Otra posibilidad para realizar el control en lazo cerrado con Arduino, sería realizar un encoder con líneas más anchas. Esto daría un mayor margen de tiempo para la lectura de líneas, con lo que Arduino podría ser capaz de leer las líneas del encoder en tiempo real. Hay que tener en cuenta que esto daría lugar a una menor resolución en la observación de la velocidad, resultando en un peor control.

En función de la aplicación que se le quiera dar a la cinta transportadora, se elegirá un tipo de control u otro y con ello se elegirá el uso de un controlador u otro. En caso de que se quiera integrar la cinta transportadora con otros dispositivos, será necesario realizar un control en lazo cerrado para conocer la velocidad. Si no, no sería necesario conocer la velocidad exacta y conseguir mantenerla constante a pesar de la carga que tenga o de la tensión de la banda, por lo que bastaría con un control en lazo abierto, pudiendo trabajar con un microcontrolador.

En la *Tabla 3.3* se hace un resumen de lo explicado anteriormente. De esta manera se ve qué tipo de controlador escoger, Arduino o FPGA, en función de la aplicación que se le quiera dar a la cinta transportadora y reflejando el tipo de control que se debe realizar.

Tabla 3.3: Resumen de uso de los controladores

<i>Control</i> <i>Aplicación</i>	Integración con otros dispositivos	Conseguir desplazamiento de objetos
Lazo abierto	-	Arduino
Lazo cerrado	FPGA	-

3.5. Estudio económico

En la Tabla 3.4 se presenta el presupuesto de este proyecto.

Tabla 3.4: Estudio económico del proyecto

DESCRIPCIÓN	CANT.	UD	PRECIO UNITARIO	IMPORTE (€)	PROVEEDOR
Sistema mecánico					
Perfil de aluminio (20x20) Tipo 1 ranura 5 450 mm de longitud	2	ud.	1,46	2,92	Motedis [33]
Perfil de aluminio (20x20) Tipo 1 ranura 5 260 mm de longitud	2	ud.	0,85	1,70	Motedis
Escuadra 20x40 tipo 1 ranura 5	4	ud.	0,66	2,64	Motedis
Tornillo M4x10	8	ud.	0,0099	0,08	Rationalstock [34]
Tuerca para ranura 5 tipo 1 M4	8	ud.	0,18	1,44	Motedis
Tornillo M3x40	6	ud.	0,0139	0,08	Rationalstock
Tuerca para ranura 5 tipo 1 M3	6	ud.	0,18	1,08	Motedis
Piezas impresas	0,05	kg	15	0,75	Amazon [35]
Banda de goma EVA	2	ud.	0,50	1,00	Motedis
Sistema de transmisión	1	ud.	Reciclado	0,00	
Subtotal				11,69	
Sistema electrónico de potencia					
Fuente de alimentación	1	ud.	Reciclado	0,00	
Transistor canal n irlb8721	2	ud.	0,73	1,46	Farnell [36]
Transistor canal p fqp27p06	2	ud.	1,04	2,08	Farnell
Optoacoplador 4N35	2	ud.	0,38	0,76	Farnell
Resistencias	6	ud.	0,10	0,60	Farnell
Condensador electrolítico de 100 µF	1	ud.	0,52	0,52	Farnell
Motor de corriente continua	1	ud.	Reciclado	0,00	
Encoder	1	ud.	Reciclado	0,00	
Subtotal				5,42	
Sistema electrónico de control					
Arduino UNO	1	ud.	20,00	20,00	Arduino [37]
FPGA Basys de Digilent	1	ud.	90,83	90,83	Trenz electronic [38]
TOTAL OPCIÓN LAZO ABIERTO (Arduino)				37,11	
TOTAL OPCIÓN LAZO CERRADO (FPGA)				107,94	

Si se quisiera implementar el control en lazo abierto, la mejor opción sería el uso de Arduino, lo que llevaría a un coste del proyecto de 37,11 €. En cambio, si se quisiera trabajar con lazo cerrado, sería mejor el uso de una FPGA, lo que implicaría un coste de 107,94 € para implementar el proyecto.

Dado que en este trabajo se han implementado ambas opciones, el coste total asciende a 127,94 €.

La mayor parte del coste del proyecto está en los controladores, es decir, en la electrónica de control, ya que en el resto del proyecto el coste asciende únicamente a 17,11 €. Los controladores utilizados eran los que estaban disponibles en el laboratorio del Área de Tecnología Electrónica de la URJC, pero podrían usarse versiones más baratas. Podría sustituirse el Arduino UNO, por ejemplo, por la placa UNO R3 de HTA3D [39] (compatible con Arduino), con un coste de 6,50 €, frente a los 20 € que cuesta el Arduino UNO. Esto dejaría un coste de **23,61 €** para implementar el control en lazo abierto. En cuanto a la FPGA, podría usarse una más barata como la Alhambra [40], que cuesta 49,90 € frente a los 90,83 € que costaba la placa usada y además es hardware libre. Eso hace que el proyecto más caro quede en un precio de **67,01 €**.

3.6. Líneas futuras

Este proyecto tiene muchas ramas por las que continuar.

Una de las opciones más interesantes sería integrar varios trabajos fin de grado juntos. En el Área de Tecnología Electrónica de la Universidad Rey Juan Carlos se fomenta el desarrollo de aplicaciones de bajo coste en automatización y robótica en trabajos fin de grado. Algunos de los TFG ya desarrollados en este ámbito son el desarrollo de un brazo robótico [41] y el software de visión artificial [42]. Ambos proyectos fueron desarrollados en FPGA, de manera que es compatible la unión de estos proyectos. Esto daría como resultado un sistema en el que se controle la clasificación de piezas con el brazo robótico en función de su color gracias a la visión artificial mientras se mueven en una cinta transportadora a velocidad constante. Es importante que la velocidad de la cinta transportadora sea constante y conocida, de ahí la importancia de trabajar con la cinta en lazo cerrado si se busca trabajar con otros sistemas, ya que de no hacerse así, no podría programarse este funcionamiento conjunto. Esto daría lugar a un sistema típico en las industrias.

Arduino es una plataforma electrónica de código abierto, pero la FPGA utilizada de Digilent no lo es. Hay una FPGA llamada Alhambra que sí que es abierta [40], por lo que se podría trasladar el diseño a esta plataforma para acercarse más a la filosofía de hardware libre.

Otra opción sería la ampliación de la parte mecánica. Podrían hacerse clasificaciones de objetos a distinto nivel, de manera que se elevaran los objetos que se quisieran por un sistema con un tornillo sin fin. También podría usarse un sistema de separación rotativo con un motor paso a paso o mediante un sistema de empuje con otro motor.

Podrían agregarse distintos sensores a la cinta transportadora. Por ejemplo un sensor de infrarrojos, ultrasonidos o una cámara para conocer la posición de los objetos en la cinta transportadora.

Por otra parte, dado que en el estudio de control se ha visto que no se podía trabajar con el encoder reciclado, podría usarse un sistema de posicionamiento más grueso que permita a Arduino su lectura. Además, podría añadirse un control integral y derivativo.

En resumen, las ampliaciones que pueden realizarse son muchas gracias al uso de la FPGA, que tiene una gran flexibilidad, y por el estudio de control, que permite ir por diferentes ramas. Además, al ser la cinta transportadora un equipo utilizado en tantos ámbitos, es sencillo añadirle más funcionalidades fáciles de aplicar gracias a su diseño.

4. CONCLUSIONES

Por una parte, se puede comprobar que se han cumplido los objetivos fijados en el apartado

2.OBJETIVOS:

- Se ha realizado un estudio en el que se implementa el control en lazo abierto y lazo cerrado, tanto en Arduino como en la FPGA, haciendo una comparación entre ambos.
- Se ha diseñado el circuito de potencia del puente en H, utilizado para controlar el sentido del motor y proporcionarle la potencia necesaria para funcionar. Para ello fue necesario trabajar con las hojas de características de los componentes y realizar ensayos a aquellos componentes reciclados de los cuales no se tenían sus hojas de características.
- Se han utilizado componentes reutilizados, tanto mecánicos (los tambores de la cinta transportadora y el sistema de transmisión eje-motor) como electrónicos (el motor y el encoder). Esto contribuye a tener un desarrollo sostenible y ayuda a concienciar a todo aquel que estudie este proyecto.
- El proyecto es accesible desde el perfil de Github de la autora del trabajo [1], comprometiéndose por tanto con la filosofía de hardware libre.
- Es un proyecto económico, pudiendo hacerlo por menos de 100 €, como puede comprobarse a través del presupuesto de este trabajo, véase en el apartado 3.5.*Estudio económico*.
- Dejar el diseño abierto a modificaciones y ampliaciones para que quien quiera pueda replicarlo.

En cuanto a las conclusiones del estudio comparativo entre Arduino y FPGA, se tiene:

Que en función de la aplicación que se le quiera dar a la cinta, será necesario un control en lazo cerrado o lazo abierto. En caso de querer realizar un control en lazo cerrado, apto para aplicaciones en las que se necesita mantener una velocidad constante y conocida, se ha comprobado que debe usarse la FPGA por su forma de trabajo concurrente. Las FPGAs son óptimas cuando se trabaja con señales en tiempo real, como sucede con el encoder. Para el control en lazo abierto es mejor el uso de un microcontrolador como Arduino, ya que son más baratos y es más sencilla su programación.

Además, cabe añadir que este proyecto ha tenido un impacto formativo importante debido a su multidisciplinariedad, estando relacionado con varias asignaturas estudiadas a lo largo de la carrera como: expresión gráfica, a través del diseño de piezas; diseño de sistemas electrónicos y sistemas electrónicos digitales, a través del trabajo con la FPGA y Arduino; control y automatización, trabajando con el lazo abierto y lazo cerrado; y tecnología electrónica, por el diseño del puente en H, entre otras.

5. BIBLIOGRAFÍA

- [1] RODELLAR, C. *TFG* [en línea]. 23 septiembre 2018. Disponible en:
<https://github.com/clararodellar/TFG>
- [2] BOLTON, W. *Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica*. 4ª edición. Barcelona: Marcombo, S.A., 2010. ISBN 978-84-267-1632-3.
- [3] RIETDIJK, A. *Ten propositions on mechatronics*. Mechatronics in Products and manufacturing Conference, Lancaster, England, 1989.
- [4] NAJERA, S. Meca [imagen digital en línea] n.d. [Consulta: 4 septiembre 2019]. Disponible en:
<https://commons.wikimedia.org/w/index.php?curid=7999623>
- [5] ALCIATORE, D.G. y HISTAND, M.B. *Introduction to Mechatronics and Measurement Systems*. 4ª edición. Nueva York: McGraw Hill, 2012. ISBN 978-0-07-338023-0.
- [6] *Open Source Hardware Definition*. Oshwa. 13 febrero 2019 [consulta: 3 septiembre 2019].
Disponible en: <https://www.oshwa.org/definition/>
- [7] *The open Source Definition*. n.d. [consulta: 3 septiembre 2019] Disponible en:
<https://opensource.org/docs/osd>
- [8] MACHADO, F., MALPICA, N. y BORROMEO, S. *Parametric CAD modeling for open source scientific hardware: comparing OpenSCAD and FreeCAD Python scripts*.
- [9] ARANGO, S. *Maker Movement, una nueva cultura de invención, innovación e intercambio* [en línea]. 6 febrero 2017 [Consulta: 5 septiembre 2019]. Disponible en:
<http://www.youngmarketing.co/la-cultura-del-maker-movement-y-como-esta-cambiando-el-mundo/>
- [10] RAJEWSKI, J. *Learning FPGAs*. Sebastopol: O'Reilly Media, 2017. ISBN 978-1-491-96549-8.
- [11] MAXFIELD, C. *The Design Warriors Guide to FPGAs*. Burlington: Elsevier, 2004. ISBN 0-7506-7604-3.
- [12] MAIDANA, A. *Selección de bandas transportadoras* [en línea]. 6 julio 2017 [Consulta: 4 septiembre 2019]. Disponible en:
<https://slideplayer.es/slide/5420321/>
<https://slideplayer.es/slide/5420321/>

- [13] LÓPEZ, J. *Bandas* [en línea]. n.d. [Consulta: 4 septiembre 2019]. Disponible en: <http://biblio3.url.edu.gt/Libros/2013/ing/pim/3.pdf>
- [14] CONDOR ARDINO S.R.L. *Curso: cintas transportadoras* [en línea]. n.d. [Consulta: 4 septiembre 2019]. Disponible en: https://www.academia.edu/6029577/CURSO_CINTAS_TRANSPORTADORAS_Especialistas_en_Bandas_Transportadoras
- [15] FERNÁNDEZ, A. *Funcionamiento de correas transportadoras* [en línea]. n.d. [Consulta: 4 septiembre 2019]. Disponible en: <http://www.fernandezantonio.com.ar/documentos%5C020.pdf>
- [16] Definición de cinta transportadora [en línea] n.d. [Consulta: 4 septiembre 2019]. Disponible en: <http://www.proyectosfindecarrera.com/definicion/cinta-transportadora.htm>
- [17] SANTIBÁÑEZ, A. *Bandas transportadoras* [en línea] n.d. [Consulta: 4 septiembre 2019]. Disponible en: <https://www.slideshare.net/arvimar/bandas-transportadoras>
- [18] ALVARADO, D., MARCELA, A., LARA, S. Y PÉREZ, N. *Bandas transportadoras* [en línea] n.d. [Consulta: 4 septiembre 2019]. Disponible en: <https://slideplayer.es/slide/5258251/>
- [19] *Conveyor Belt Kit | Simplest Mini Production Line* [en línea] n.d. [Consulta: 5 septiembre 2019]. Disponible en: <https://www.dobot.cc/products/conveyor-belt-kit-overview.html>
- [20] *Autovictoria Cinta transportadora 150 cm x 19,8 cm* [en línea] n.d. [Consulta: 5 septiembre 2019]. Disponible en: <https://www.amazon.es/dp/B074GV9LW1/> [Archivado: <http://archive.is/i3lcd>]
- [21] *RunnerFit Sprint Vibrator* [en línea] n.d. [Consulta: 5 septiembre 2019]. Disponible en: https://www.storecotec.com/es/cintas-de-correr-y-andar/217-runnerfit-sprint-vibrator.html?gclid=CjwKCAjwzdLrBRBiEiwAEHrAYrMEwrwEOUkuUp3oMhHVwhQKdmDq90h6WkMMeVmy0OG_AVqBUtS2-xoCo70QAvD_BwE
- [22] *Modular & portable conveyor belt. speed control by Arduino* [en línea]. 21 septiembre 2017 [Consulta: 17 noviembre 2018]. Disponible en: <https://www.instructables.com/id/MODULAR-PORTABLE-CONVEYOR-BELT-SPEED-CONTROL-BY-AR/>
- [23] *DIY Conveyor Belt* [en línea]. 9 octubre 2017 [Consulta: 6 septiembre 2019]. Disponible en: <https://www.instructables.com/id/DIY-Conveyor-belt/>

- [24] *How to Build a Conveyor Belt DIY with Kit Conveyors Top Tips* [en línea]. n.d. [Consulta: 6 septiembre 2019]. Disponible en: <http://www.kit-conveyors.co.uk/how-to-build-a-conveyor.html>
- [25] TANSKANEN, P. Management and recycling of electronic waste. En: *Acta Materiala*. Elsevier, 2013, no 61, pp. 1001-1011
- [26] FRAILE, J. *Máquinas eléctricas*. 6ª edición. Madrid: McGraw Hill, 2008. ISBN 978-84-481-6112-5.
- [27] SCARPINO, M. *Motors for makers*. Indiana: Pearson Education, 2016. ISBN: 978-0-13-403283-2.
- [28] MARGOLIS, M. *Arduino Cookbok*. Sebastopol: O'Reilly Media, 2011. ISBN 978-0-596-80247-9.
- [29] *Motores paso a paso* [en línea]. n.d. [Consulta: 18 noviembre 2019]. Disponible en: <https://www.prometec.net/motores-paso-a-paso/>
- [30] *Más sobre motores de corriente continua* [en línea]. n.d. [Consulta: 18 noviembre 2019]. Disponible en: <https://www.prometec.net/hbridge/>
- [31] OGATA, K. *Ingeniería de control moderna*. 5ª edición. Madrid: Pearson Educación, S.A., 2010. ISBN 978-84-8322-660-5.
- [32] MACHADO, F. y BORROMEO, S. *Diseño de circuitos digitales con VHDL*. Móstoles: 2010. ISBN 978-84-693-4652-5.
- [33] Motedis [en línea] n.d. [Consulta: 27 septiembre 2019]. Disponible en: <https://www.motedis.es/shop/>
- [34] RationalStock [en línea] n.d. [Consulta: 27 septiembre 2019]. Disponible en: <https://www.rationalstock.es/>
- [35] Farnell [en línea] n.d. [Consulta: 27 septiembre 2019]. Disponible en: <https://es.farnell.com/>
- [36] Amazon [en línea] n.d. [Consulta: 27 septiembre 2019]. Disponible en: <https://www.amazon.es>
- [37] Arduino [en línea] n.d. [Consulta: 27 septiembre 2019]. Disponible en: <https://store.arduino.cc>
- [38] Trenz electronic [en línea] n.d. [Consulta: 30 septiembre 2019]. Disponible en: <https://www.trenz-electronic.de/>
- [39] HTA3D [en línea] n.d. [Consulta: 30 septiembre 2019]. Disponible en: <https://www.hta3d.com/es/placa-desarrollo-uno-r3-compatible-arduino>

[40] Alhambra bits [en línea]. n.d. [Consulta: 30 septiembre 2019]. Disponible en:

<https://alhambrabits.com/buy/>

[41] MEQUISEDEC DE LA FUENTE, S. *Diseño y construcción del sistema electromecánico de un escáner 3D de hardware libre controlado por FPGA*. Trabajo de Fin de Grado, Universidad Rey Juan Carlos, Móstoles, 2018.

[42] PASTOR, N. *Sistema de control basado en tamaño y color de piezas, utilizando un brazo robótico y visión artificial*. Trabajo de Fin de Grado, Universidad Rey Juan Carlos, Móstoles, 2018.

[43] Surplus Sales. *Motor* [en línea]. n.d. [Consulta: 9 marzo 2019]. Disponible en:

<https://www.surplussales.com/motors/motors-3.html>

[44] Colaboradores de Wikipedia. *Optoacoplador* [en línea]. Wikipedia, La enciclopedia libre, 2019

[Consulta: 8 septiembre]. Disponible en: <https://es.wikipedia.org/wiki/Optoacoplador>

[45] *Adaptadores de nivel entre 5V y 3.3V* [en línea]. 23 febrero 2018. [Consulta: 2 septiembre 2019].

Disponible en: <https://www.inventable.eu/2017/05/03/adaptadores-nivel-5v-3-3v/>

6. ANEJOS

Anejo I Herramientas utilizadas

Componentes utilizados

Para el sistema mecánico

- 2 Perfiles de aluminio (20x20) Tipo 1 ranura 5 450 mm de longitud
- 2 Perfiles de aluminio (20x20) Tipo 1 ranura 5 260 mm de longitud
- 4 Escuadras 20x40 tipo 1 ranura 5
- 8 Tornillos M4x10
- 8 Tuercas para ranura 5 tipo 1 M4
- 6 Tornillos M3x40
- 6 Tuercas para ranura 5 tipo 1 M3
- Piezas impresas diseñadas
- Banda de goma EVA
- Sistema de transmisión (reutilizado)

Para el sistema electrónico de potencia

- Fuente de alimentación
- 2 transistores canal n irfb8721
- 2 transistores canal p fqp27p06
- 2 optoacopladores 4N35
- 6 resistencias (2 de 270 Ω , 2 de 200 Ω y 2 de 11,8 k Ω)
- Condensador electrolítico de 100 μ F
- Motor de corriente continua
- Encoder

Para el sistema electrónico de control

- Arduino UNO
- FPGA Basys de Digilent

Herramientas software utilizadas

FreeCAD

Es la herramienta de diseño CAD utilizada para diseñar las piezas 3D necesarias para el montaje mecánico de este proyecto. Es un software de código abierto y puede usarse tanto en Windows como en Linux o Mac OSX. Genera archivos tipo FCSTD, pero es capaz de leer y escribir otros tipos de ficheros. Por ejemplo, genera archivos de extensión STL. Este tipo de archivos guardan la malla que necesita el software de la impresora para la impresión 3D.

Repetier Host

Es un software gratuito y abierto que se encarga de transformar el archivo STL en órdenes para que la impresora 3D las interprete e imprima la pieza según las especificaciones dadas a través de este programa. Es capaz de controlar parámetros como la temperatura del extrusor de la impresora o la densidad que tenga la pieza, entre otros.

Github

Es una plataforma de desarrollo colaborativo de código abierto donde se comparten proyectos para aprender y contribuir. Aquí está subido este proyecto para que cualquiera pueda estudiarlo o replicarlo.

ISE

ISE son las iniciales de Integrated Synthesis Environment, que significa entorno de síntesis integrado. Es una herramienta de software de desarrollo de circuitos creada por Xilinx. Permite sintetizar los diseños HDL, simular la reacción del diseño a distintos estímulos y cargarlos a la FPGA. Es el entorno utilizado para programar la FPGA.

Arduino IDE

Es el software utilizado para controlar Arduino. Es un software libre, de código abierto. Al igual que FreeCAD, se puede usar desde distintos sistemas operativos.

Herramientas hardware utilizadas

Impresora 3D

La impresora 3D de la que se dispone en el laboratorio del Área de Tecnología Electrónica de la Universidad Rey Juan Carlos es la impresora iDeator12. Su funcionamiento se basa en la tecnología FFF (Fused Filament Fabrication), que es fabricación por filamento fundido. Para ello se usa un material termoplástico, normalmente PLA o ABS, que se funde en el extrusor para posteriormente depositarse en una superficie denominada cama. Va formando la pieza por capas, que al enfriarse le da la rigidez a la pieza.

FPGA

El dispositivo lógico programable utilizado es la FPGA Basys de Digilent, la Spartan 3E. El término FPGA significa matriz de puertas programables en campo. Es decir, con una FPGA se pueden crear circuitos digitales reprogramables mediante un lenguaje de descripción de hardware. El lenguaje utilizado en este proyecto es VHDL. Para desempeñar su función, está formado por CLBs (Figura 1.3).

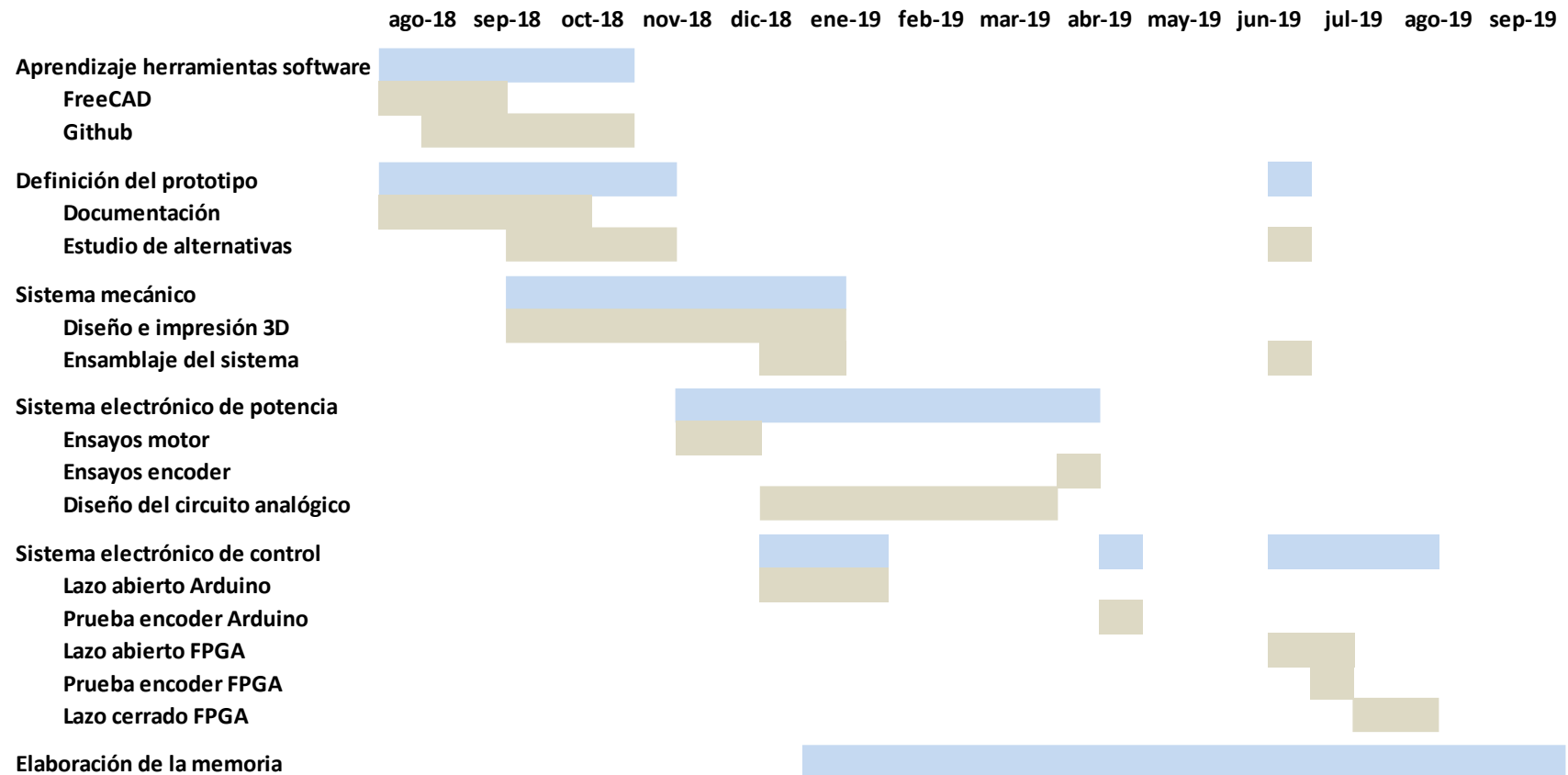
Arduino

Arduino UNO es la placa de hardware libre utilizada para el desarrollo de este trabajo. Consta de un microcontrolador y de pines digitales y analógicos programables a través del software Arduino IDE (Integrated Development Environment). Una de las características de Arduino es que usa una memoria flash para guardar los programas. Esta es una memoria no volátil, por lo que se mantiene el programa guardado anteriormente incluso después de haberlo desconectado de la corriente. No es modificable en tiempo real por lo que las variables se guardan en la SRAM.

Anejo II Diagrama de Gantt

Se realiza un diagrama de Gantt para ver la programación del proyecto para poder alcanzar los objetivos fijados:

Tabla 6.1: Diagrama de Gantt de programación del proyecto



Anejo III Explicación de los componentes necesarios y cómo elegirlos para el circuito electrónico de potencia

1. Motor

El motor de la impresora viene designado de la siguiente manera: C6409-60004 (RN951524). El único dato que se consigue encontrar sobre este motor es que su tensión nominal (la que no se debe superar) es de 12 V [43]. La complicación que tiene trabajar con elementos recuperados es que, si no se consigue encontrar la hoja de características de ese componente, habrá que hacer diversas pruebas para conocer su rango de funcionamiento. En este caso, ante la falta de información del motor, se realizan una serie de ensayos.

Estos son un **ensayo de rotor libre** y otro de **rotor bloqueado** para ver la intensidad que consume y poder determinar los componentes necesarios en nuestro circuito. También se hizo un ensayo con carga. En un ensayo con carga lo que se hace es frenar el eje pero sin que termine de pararse, para simular que esté soportando un peso.

La conexión eléctrica que se usa para estos ensayos es la siguiente:

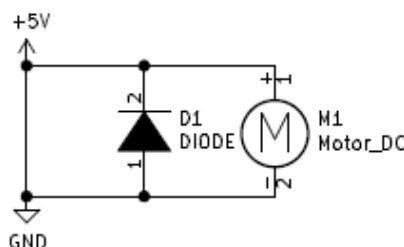


Figura 6.1: Conexión del motor de continua para realizarle los ensayos

Únicamente es necesario conectar el motor a una fuente de corriente continua y un diodo en paralelo conectado en inversa. Su función es descargar el motor cuando se desconecte el circuito y se le denomina diodo de recirculación. De esta manera, cuando la fuente está alimentando al circuito, al tener el cátodo conectado al borne positivo no consume corriente (el diodo está en inversa por lo que no conduce). Pero, cuando el motor no esté conectado a la fuente, el motor podrá descargarse al estar conectado con el diodo, ya que se cerrará el circuito con el diodo. Esto es necesario porque un motor se modeliza como una bobina y las bobinas almacenan energía en forma de corriente. Por tanto, al desconectar la fuente de tensión, se corta la corriente de golpe, de manera que la siguiente derivada tiende a infinito (físicamente lo que sucede es que se da un valor muy alto):

$$\varepsilon = L \frac{dI}{dt}$$

Esta fórmula define el comportamiento de una bobina en un circuito eléctrico en corriente continua. Puede verse en la fórmula que una rápida variación de corriente provocaría un pico de tensión.

De esta manera, el diodo pretende proteger al motor de los picos de tensión. Esto es porque al quedar corriente almacenada en el motor, aunque se desconecte la fuente de tensión, la corriente tiene por donde circular, a través del diodo, cerrándose el circuito y disipándose poco a poco la corriente almacenada en el motor. De esta manera no se da esa alta variación de corriente tan rápidamente sino que se disipa poco a poco.

A continuación se muestran los resultados de los ensayos realizados.

- Ensayo de rotor libre o de vacío:
 - A 5 V $\rightarrow I_0 = 0,15$ A
 - A 12 V $\rightarrow I_0 = 0,2$ A
- Ensayo con carga:
 - A 5 V $\rightarrow I_0 = 0,65$ A
 - A 12 V $\rightarrow I_0 = 0,9$ A
- Ensayo de rotor bloqueado:
 - A 5 V $\rightarrow I = 0,8$ A
 - A 12 V $\rightarrow I = 1,6$ A

De esta manera, para componentes del puente en H, hay que comprobar en su hoja de características que sean capaces de soportar corrientes de 1,6 A. Aunque el motor probablemente nunca llegue a aguantar ese amperaje, es justo el máximo que puede alcanzar, por lo que el resto de componentes es mejor que también puedan soportarlo para que no se quemen en caso de que se pare el motor mientras está alimentado a 12 V. A pesar de ello, la intensidad con la que normalmente trabajará será alrededor de 0,65 A en caso de trabajar con un voltaje de 5 V o de 0,9 A si se trabaja con 12 V.

2. Transistores

Como ya se ha comentado, los transistores se usan para conseguir la conmutación de interruptores con señales eléctricas.

Podrían usarse tanto transistores bipolares como mosfet. En caso de que se usaran transistores bipolares, se debe tener en cuenta que sería necesario colocar diodos de recirculación, tal y como se explicó en la conexión eléctrica de los ensayos realizados al motor, quedando algo parecido a lo siguiente:

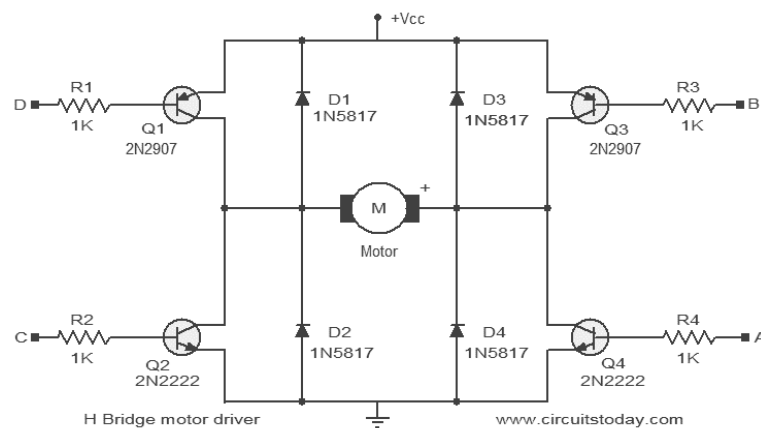


Figura 6.2: Puente en H con transistores bipolares [30]

Pero en el circuito diseñado se usan transistores mosfet. Una de las ventajas de trabajar con mosfets es que es más sencillo el diseño del circuito, ya que la corriente de salida es controlada por la tensión de entrada, a diferencia de los bipolares, que es controlada por la corriente de entrada. Además, éstos tienen incorporados los diodos de recirculación en su encapsulado, ahorrando ponerlos en el circuito. Por ello, se eligió el uso de transistores mosfet.

Por otra parte, debe tenerse en cuenta que interesa trabajar con estos transistores como si fueran interruptores, no para amplificar, sólo para que pase corriente o no. Fijándose en la Figura 6.3, se ve que no interesa trabajar en la zona activa o de saturación, ya que consume mucho y no es necesario para la aplicación que se necesita. Lo que se busca es que trabaje en corte cuando no se quiere que corra corriente y trabaje en la zona lineal cuando se quiere que circule la corriente, que es donde se pueden conseguir intensidades menores y que, por tanto, el consumo sea menor.

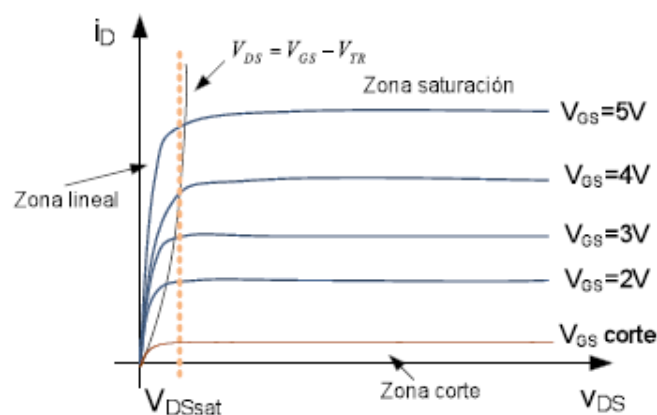


Figura 6.3: Característica V-I de salida de un mosfet [Romero, B. Material del aula de tecnología electrónica, URJC]

Para trabajar en la zona lineal, por tanto, se necesita un transistor con un voltaje entre fuente y drenador de saturación ($V_{DS\text{ sat}}$) superior a 12 V, que es la tensión con la que se trabaja en el circuito. Esto fue un requisito cuando se buscó qué transistores elegir.

Dado que el funcionamiento que va a darse a los mosfet es de interruptor, en la Figura 6.4 se ve cuándo permite el paso de corriente y cuando no en función de su entrada y del tipo de canal que sea:

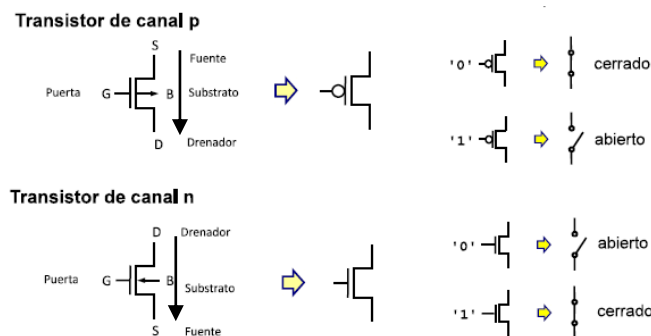


Figura 6.4: Funcionamiento de mosfet en corte y en zona lineal para canal n y p [Machado, F. Material del aula de diseño electrónico digital, URJC]

Es importante darse cuenta que para un mosfet de canal n, la intensidad va de drenador a fuente, mientras que para uno de canal p la intensidad va de fuente a drenador.

Al colocar de forma alternativa los transistores de canal p y n, se consigue la conmutación que invierte la polaridad de la tensión en los bornes del motor, tal y como se vio en el apartado 3.3.4. *Inversión del sentido: Puente en H.*

3. Optoacoplador

En el circuito se usan optoacopladores, concretamente el 4N35. Su función es aislar eléctricamente el circuito de control del circuito de potencia externo. De esta manera, si se quemara la conexión del motor, no se quemaría ni el dispositivo de control ni el dispositivo al que estuviera éste conectado, como podría ser un ordenador.

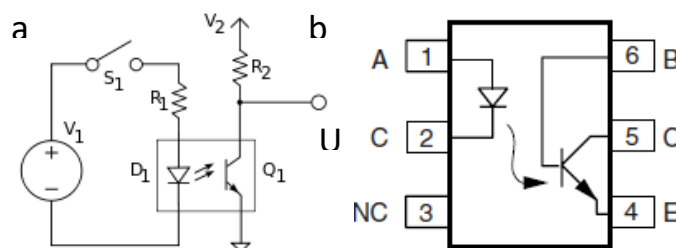


Figura 6.5: Conexión interna de un optoacoplador [44]

Para conseguir la separación eléctrica, internamente su conexión es óptica. Para ello hay un fotoemisor y un fotorreceptor. En el caso del optoacoplador 4N35 se tiene un diodo como fotoemisor y un fototransistor que hace la función de fotorreceptor. En la tabla [Tabla 6.2](#) se muestra su funcionamiento. En la [Figura 6.5](#), si se cerrara el interruptor S1, atravesaría una corriente al led D1, que emitirá por tanto una luz. Si la luz que emite es suficiente, saturará al fototransistor, haciendo que circule una corriente a través de R2. Esto hará que en U el voltaje que haya sea de 0 V. En cambio, si el interruptor S1 se deja abierto, el diodo no emite luz y por tanto Q1 está en zona de corte, por lo que no pasa corriente por R2. Esto hará que el voltaje en U sea de 12 V. Esto será necesario saberlo para entender el funcionamiento de nuestro circuito de puente en H [5] [44].

Tabla 6.2: Funcionamiento optoacoplador en función de las entradas

S1	U (V)
Abierto: 0	V2
Cerrado: 1	0

El cálculo de la resistencia R1 según la [Figura 6.5](#) para el proyecto se hace en el [Anejo IV](#).

Anejo IV Cálculo de la resistencia conectada al optoacoplador en función del dispositivo de control.

El optoacoplador es el dispositivo que se encarga de separar eléctricamente el circuito de potencia del de control. Está conectado al circuito de control en el lado del diodo, por lo que hay que calcular la resistencia conectada al led del optoacoplador. Para ello es necesario ir a la hoja de características del componente y mirar tanto la corriente máxima que puede circular por el diodo como la tensión umbral o de codo del diodo.

ABSOLUTE MAXIMUM RATINGS ⁽¹⁾				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
INPUT				
Reverse voltage		V_R	6	V
Forward current		I_F	50	mA
Surge current	$t \leq 10 \mu s$	I_{FSM}	1	A
Power dissipation		P_{diss}	70	mW

Figura 6.6: Índices absolutos máximos del componente 4N35 para el diodo

Puede verse que el valor de intensidad máximo es:

- $I_F = 50 \text{ mA}$

Este valor de intensidad no puede superarse, ya que si no se quemaría el componente. Para elegir la intensidad que circule por el diodo, se escoge la menor intensidad posible (para consumir poco) que maximice la transferencia. Para ello hay que fijarse en la gráfica CTR (current transfer ratio, relación de transferencia de intensidad en español):

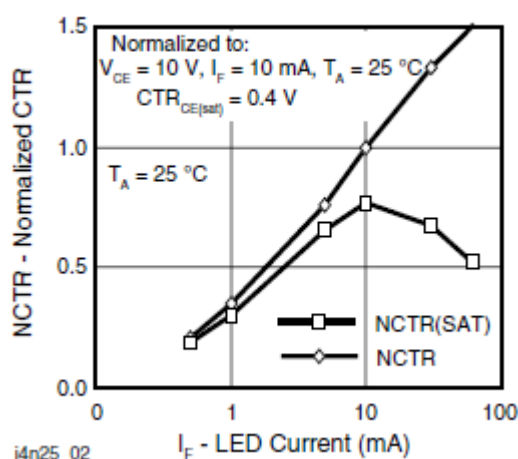


Figura 6.7: CTR normalizado no saturado y saturado vs. Corriente LED

La intensidad que maximiza la transferencia es:

- $I_F = 10 \text{ mA}$
- Siendo $CTR = i_c / i_F = 0,7$

Conociendo la intensidad del led y con la Figura 6.8 se puede ver cuál es la tensión de codo o umbral para esta intensidad:

ELECTRICAL CHARACTERISTICS (1)							
PARAMETER	TEST CONDITION	PART	SYMBOL	MIN.	TYP.	MAX.	UNIT
INPUT							
Junction capacitance	$V_R = 0 \text{ V}, f = 1 \text{ MHz}$		C_j		50		pF
Forward voltage (2)	$I_F = 10 \text{ mA}$		V_F		1.3	1.5	V
	$I_F = 10 \text{ mA}, T_{amb} = -55^\circ\text{C}$		V_F	0.9	1.3	1.7	V
Reverse current (2)	$V_R = 6 \text{ V}$		I_R		0.1	10	μA
Capacitance	$V_R = 0 \text{ V}, f = 1 \text{ MHz}$		C_O		25		pF

Figura 6.8: Características eléctricas del led del componente 4N35

- $V_{umbral} = 1,3 \text{ V}$

Con estos datos se puede calcular la resistencia a través del cálculo de la malla del diodo, que será distinta en función del dispositivo de control que se tenga:

Arduino

El voltaje que suministra Arduino es de 5 V, por lo que la malla queda de la siguiente manera:

$$\begin{aligned}
 \text{Malla diodo: } 5 &= i_F \cdot R + V_{umbral} \\
 R &= \frac{5 - V_{umbral}}{i_F} = \frac{5 - 1,3}{10 \cdot 10^{-3}} = 370 \, \Omega
 \end{aligned}$$

De esta manera, con esa resistencia se conseguiría la máxima transferencia. Por disponibilidad de componentes la resistencia usada es de 270 ohmios, de manera que la intensidad que realmente atraviesa es:

$$i = \frac{5 - V_{umbral}}{R} = \frac{5 - 1,3}{270} = 13,7 \text{ mA}$$

Como Arduino puede proporcionar hasta 40 mA por un mismo pin, no hay problemas de quemar el dispositivo y esta intensidad sigue dando una buena relación de transferencia.

FPGA

Dado que el nuevo sistema de control es la FPGA y ésta proporciona un voltaje distinto al de Arduino (3,3V en lugar de 5V), será necesario reajustar las resistencias que conectan con la fuente de control para mantener la misma intensidad. Además, se debe tener en cuenta que la intensidad que soporta

la FPGA es menor a la que soporta Arduino. La intensidad máxima que soportan los pines de una FPGA es de 16 mA mientras que Arduino puede soportar 40 mA.

Se vio anteriormente que la intensidad óptima era de 10 mA, aunque por disponibilidad se usa una resistencia de 270 Ω , teniendo en Arduino una intensidad de 13,7 mA. Esta intensidad cumplía con nuestras expectativas en Arduino, pero estaría muy cerca del máximo que puede dar la FPGA y además está alimentado por una fuente distinta, por lo que hay que rehacer el cálculo:

$$\text{Malla diodo: } 3,3 = i_F \cdot R + V_{umbral}$$

$$R = \frac{3,3 - V_{umbral}}{i_F} = \frac{3,3 - 1,3}{10 \cdot 10^{-3}} = 200 \, \Omega$$

Con una resistencia de 200 Ω da los 10 mA óptimos. Con la resistencia de 270 Ω , despejando de la ecuación anterior, se tendría una intensidad de 7,4 mA. Esta podría ser demasiado baja para el optoacoplador, dando una mala relación de transferencia, por lo que se decide usar la nueva resistencia de 200 Ω cuando se controle el sistema con la FPGA.

Anejo V Pruebas realizadas al encoder

Se debe tener en cuenta que trabajar con elementos reciclados de los cuáles no se tiene su hoja de características es complicado, ya que para averiguar sus rangos de funcionamiento se deben hacer pruebas.

En primer lugar, como se comentó, es necesario analizar los distintos cables para poder hacer las conexiones eléctricas, lo que se hace por ingeniería inversa. Para ello se desmontó la carcasa que tenía el encoder y así ver las conexiones (Figura 3.22). En este se ve que hay un condensador. Los condensadores normalmente se conectan entre la fuente y la tierra, por lo ya se sabe que los otros dos cables son los de los sensores ópticos. Para poder diferenciar la tierra de la alimentación, dado que no hay más componentes en la misma tarjeta para poder encontrar la tierra, se estudian hojas de características de otros encoder para ver la disposición que suelen tener. Se concluye que el pin del extremo es la tierra y el otro la fuente.

Por otra parte, al ser reciclado, no se sabe si la señal que se recibe es analógica o si ya está acondicionada de forma digital, pudiendo leer unos y ceros en función del color. Para saberlo, se usa un osciloscopio para poder ver la señal. En la Figura 6.9 se muestra la forma que se obtuvo de la señal:

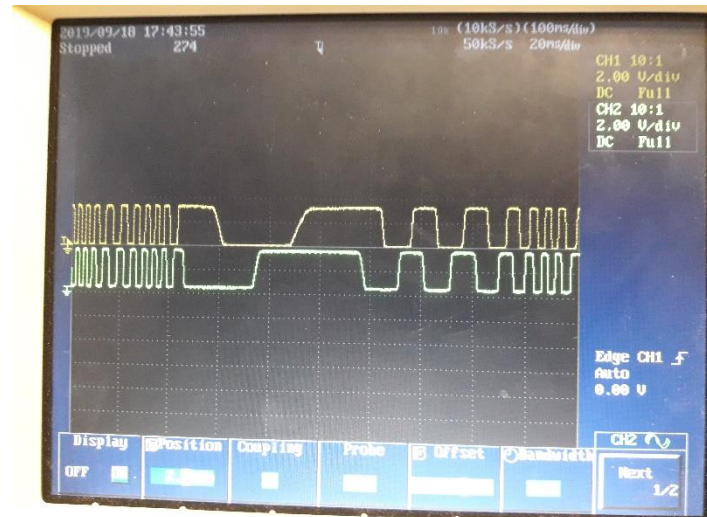


Figura 6.9: Forma de la señal del encoder vista con el osciloscopio

Puede verse que la señal ya estaba acondicionada para que la salida fuera digital.

Por último, queda saber cuál es la tensión con la que se debe alimentar el encoder. La alimentación de estos tipos de encoder suele ser de 5V por lo que se prueba y se ve que con ese voltaje es capaz de leer las marcas de la rueda. Con Arduino no hay problema para conectar los cables del encoder, pero la FPGA funciona a 3,3V, por lo que sí que puede haber problemas.

Hay componentes que trabajan a 5V mientras que otros a 3,3 V. Estos no pueden conectarse de forma general entre ellos porque el dispositivo de 3,3V podría dañarse.

En principio, no habría problemas en la conexión de un componente de salida de 3,3V a una entrada de un dispositivo de 5V, no siendo necesario un adaptador. La lógica de 5V de muchos dispositivos considera el nivel alto (un 1 digital) a partir de 2V. Aunque, no sería el caso de los transistores CMOS, siendo a partir de 3,5V, en muchos componentes sí es así. Esto permitiría que los datos provenientes de componentes de 3,3V se puedan interpretar correctamente. Siempre será necesario comprobar la hoja de características (o datasheet) del componente con el que trabajes para ver que funcione correctamente [45].

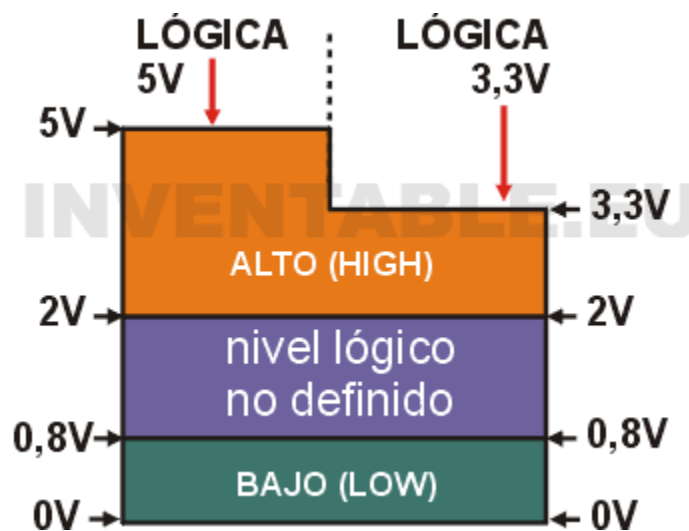


Figura 6.10: Lógica de componentes de 5 V y 3,3 V [45]

Pero, en este proyecto se tiene como componente de salida un encoder (de 5V) que se conecta a una FPGA (dispositivo de 3,3V). El voltaje del encoder supera el de funcionamiento de la FPGA, pudiendo quemarla. Por ello, es necesario el uso de un adaptador de niveles, o como se determina en inglés, logic level converter.

Antes de ver si es necesario este componente, se comprueba con Arduino si el encoder es capaz de funcionar cuando se le alimenta con 3,3 V. Se observa que sí funciona por lo que se puede conectar directamente a 3,3 V y no se necesita el conversor de nivel.

Anejo VI Unidades de velocidad

Como ya se ha comentado, en un control con lazo cerrado lo que se busca es conseguir que tu salida coincida con tu entrada. Para ello se usa un sensor que mida la salida y así se pueda comparar con la entrada pudiendo actuar en función de ese dato. En este proyecto la entrada es la velocidad deseada y la salida es la velocidad real. Para medir la velocidad real que se tiene se usa un encoder. Ya se ha visto que el encoder detecta líneas que se pueden contar y, con ello, se puede hacer un cálculo de la velocidad: la cantidad de líneas que se cuentan por unidad de tiempo (espacio entre tiempo).

El problema surge en que la velocidad deseada se da en lo que se ha llamado unidades de tensión. Como la velocidad del motor depende de la tensión en sus bornes, que se varía con el PWM, la velocidad deseada se pasa con un valor de 0 a 255, ya que el ciclo de trabajo del PWM que se tiene consta de 8 bits. Dado que la velocidad real se mide en líneas, no se puede comparar la velocidad deseada con la velocidad real porque se tienen en distintas unidades.

Por tanto, parece obvio que se necesita realizar un cambio de unidades para tener ambas variables en las mismas unidades y así poder compararlas.

Para ello, se hace una prueba en la que se mide el tiempo que tarda en dar 10 vueltas, es decir, en contar 18000 líneas a la velocidad máxima. Se decide que sean 10 vueltas para que el tiempo sea suficiente para poder tomar una medida fiable. Con este dato se pueden obtener las líneas que debe leer para cualquier velocidad. Sabiendo que tarda 2,5 segundos en dar 10 vueltas, se tiene lo siguiente:

$$2,5 \text{ s} \rightarrow 18000 \text{ barras}$$

$$16 \text{ ms} \rightarrow x \text{ barras}$$

$$x = \frac{16 \cdot 18000}{2500} = 115,2 \text{ barras}$$

Se obtiene que a la máxima velocidad es capaz de leer 115 barras (no se pueden contar partes de barras) en un tiempo de 16 ms. La velocidad es espacio dividido entre tiempo. Por tanto la muestra de número de barras que lee debe ser en un periodo de tiempo. Se escoge siempre el mismo para poder hablar de velocidad en número de barras y debe ser un tiempo lo suficientemente grande para que la medida sea fiable y lo suficientemente pequeño para que tome varias muestras y así poder hacer un control exhaustivo. Además se elige 16 ms porque es una potencia de 2, por lo que es cómodo trabajar con este número en VHDL.

Con la relación entre el número de líneas leídas y la tensión, se puede hacer el cambio de unidades para armonizar las unidades de la velocidad deseada y la velocidad real.

Se elige cambiar las unidades de la velocidad real para tenerlas en tensión, ya que estas unidades son las que se usan para conseguir el movimiento del motor.

Con el dato obtenido a partir del ensayo anterior, se tiene que la relación entre el número de barras y la tensión es:

$$255 \text{ tensión} \rightarrow 115,2 \text{ barras}$$

$$\text{y tensión} \rightarrow x \text{ barras}$$

$$\text{y tensión} = \frac{255}{115,2} \cdot x \text{ barras} = 2,21 \cdot x \text{ barras} \cong 2 \cdot x \text{ barras}$$

En VHDL no se puede operar con números decimales, por lo que se aproxima la relación a 2. Si se aproximara a 3 se le pediría que a la máxima potencia leyera más líneas de las que puede, por lo que se redondea a la baja. Además, la multiplicación por 2 en VHDL es muy sencilla ya que simplemente es un desplazamiento de bits a la izquierda.

Anejo VII Análisis del tiempo de lectura del encoder en Arduino

En el subapartado 3.4.3. *Arduino* se comprobó experimentalmente que Arduino era demasiado lento para leer el encoder, pero la prueba no cuantificaba cuánto de lejos estaba de ser capaz de leerlo.

Para cuantificar cuánto es su retraso, se realizó otra prueba. Dado que la FPGA sí que es capaz de leer el encoder, se puede calcular cuánto tarda en leer una cada una de las líneas, que sería el tiempo que debería tardar Arduino en ejecutar todo el programa.

Para conocer el tiempo que debería tardar en leerlo, se utilizan los datos obtenidos del ensayo del *Anejo VI*, donde se midió el tiempo que tardaba en dar 10 vueltas:

$$2,5 \text{ s} \rightarrow 18000 \text{ barras}$$

$$1 \text{ ms} \rightarrow x \text{ barras}$$

$$x = \frac{1 \cdot 18000}{2500} = 7,2 \frac{\text{barras}}{\text{ms}}$$

Con esta relación se puede calcular el tiempo que se tarda en leer una barra:

$$7,2 \text{ barras} \rightarrow 1 \text{ ms}$$

$$1 \text{ barra} \rightarrow x$$

$$x = \frac{1}{7,2} = 138 \mu\text{s}$$

Por tanto, para que Arduino fuera capaz de leer el encoder, debería ejecutar todo el código en este tiempo.

Para ello se realiza un programa en el que mida el tiempo en que tarda en ejecutarlo y lo muestre por pantalla en microsegundos:

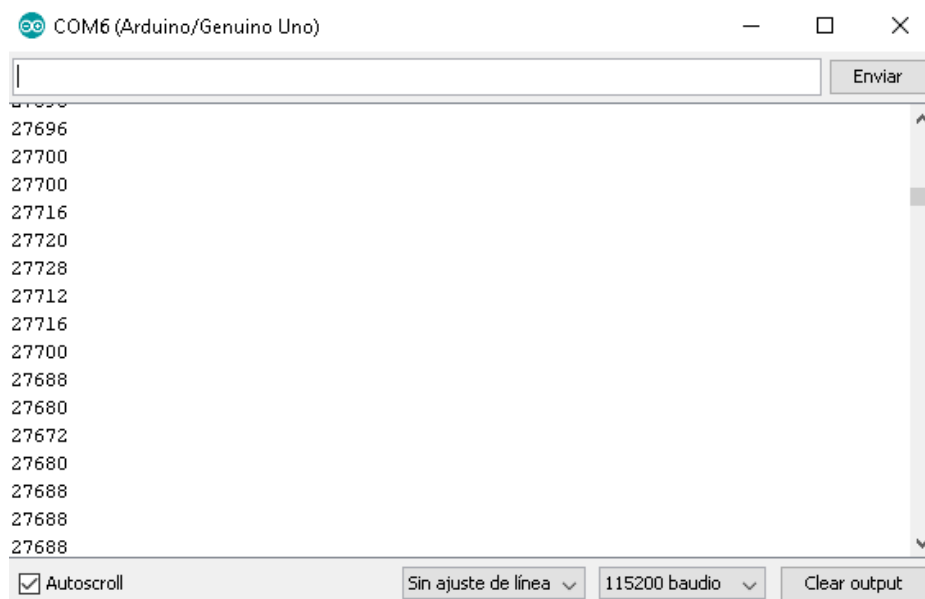


Figura 6.11: Tiempo que tarda Arduino en ejecutar el código en milisegundos

Puede verse que el tiempo que tarda en ejecutar el código es aproximadamente de 27.700 μ s frente a los 138 μ s que debería tardar para ser capaz de leer todas las líneas del encoder.

Por tanto, a la máxima velocidad sólo es capaz de leer una línea cuando debería leer 200 líneas. Por ello, no es fiable el uso de Arduino con este encoder.

Cabe mencionar que la FPGA es más rápida por varias razones:

- Su forma de trabajo concurrente. De esta manera, mientras que Arduino debe ejecutar todo el código para poder llegar a las líneas de código donde se cuentan las barras, la FPGA al hacerlo de forma paralela, está siempre disponible para contar las líneas.
- Además, la FPGA tiene un ciclo de reloj de 100 MHz frente al de Arduino que es de 16 MHz, mucho más lento.

Anejo VIII Diseño de bloques en VHDL

En este anejo se explica el diseño del PWM y del display.

- **PWM**

En el apartado de 3.3.3. *Control de velocidad: Pulse Width Modulation (PWM)* se explica qué es el PWM. Partiendo de eso, es obvio que se debe crear la señal del ciclo de trabajo (duty cycle). Como el ciclo de trabajo es la velocidad, se implementa lo explicado en el apartado 3.4.4. *FPGA*. Se puede incrementar o disminuir la velocidad con dos botones, pudiendo elegir si modificar las unidades, decenas o centenas con el interruptor.

También es necesario crear la señal de periodo del ciclo de trabajo. Esto se hace con varios contadores. Recordando que el periodo debe ser bajo para que el ojo humano no se dé cuenta, se elige que el periodo sea de 2 ms. El contador es la división del periodo entre el tamaño utilizado para el ciclo de trabajo, que en este caso es de 256. Por tanto se debe hacer un contador de 8 μ s. Con otro contador se cuenta desde 0 hasta 255 a un ritmo de 8 μ s. Con esto se tienen las 256 variaciones en el periodo del ciclo de trabajo.

Por último, es necesario un comparador para crear la señal del ciclo de trabajo. Hasta ahora lo que se tiene es un número en el ciclo de trabajo y una señal que cuenta 255 en el periodo del ciclo de trabajo que se desea tener. Ahora lo que se necesita es crear una señal que valga 1 durante el ciclo de trabajo y 0 el resto del tiempo. Con el comparador lo que se hace es comparar el número del ciclo de trabajo con la cuenta de la variación de 0 a 255 en el periodo deseado. Mientras la cuenta que se lleva sea menor que el ciclo de trabajo, la señal que se crea valdrá 1, y cuando sea mayor, valdrá 0. Con esto se consigue crear la señal PWM deseada.

- **Display**

La FPGA dispone de 4 displays de 7 segmentos más 1 punto por cada display. Esto hace que se necesiten $4 \cdot 8 = 32$ pines para encender cada segmento. Pero esto es un desperdicio de recursos, por lo que las conexiones están hechas de tal manera que todos los segmentos de la misma posición de cada display estén conectados. Esto hace que si se enciende el segmento A se encienda en todos los displays. Con esto se consigue que se necesiten únicamente 4 pines para encender cada display y 8 pines para los segmentos (7 segmentos + 1 punto). El problema que da esto es que encender 2 displays llevaría a que se muestre lo mismo por ambos. Para mostrar distintas cosas en cada display, es necesario multiplexar en el tiempo. Esto consiste en dividir un periodo de tiempo en 4 partes, una por cada display. De esta manera, en cada parte de tiempo se activa un único display por el que se muestra

el número o letra que se quiera mostrar en este display. Para seleccionar lo que se quiere mostrar por el display se utiliza un multiplexor y para encender el display se usa un decodificador. El periodo de tiempo utilizado para multiplexar debe ser lo suficientemente pequeño para que el ojo humano no se dé cuenta de que sólo hay un display encendido a la vez. De esta manera, el cerebro interpreta que todos los displays están encendidos a la vez.

Para dividir el tiempo se usan contadores. Con un contador se crea la señal de tiempo que se decide tener encendidos los displays. Con otro se cuenta la cantidad de displays que se tienen, 4 en este caso. De esta manera, pasándole esta última señal al multiplexor se elige el número que se quiere mostrar en el display, cambiándolo según la señal de tiempo del primer contador. La señal que cuenta 4 con la señal de periodo creada anteriormente además se decodifica para encender el display deseado. De esta forma, está sincronizado el número que se quiere mostrar con el display en el que se quiere mostrar para que se enciendan a la vez.

En la Figura 6.12 se muestra el diseño digital que se debe implementar el VHDL:

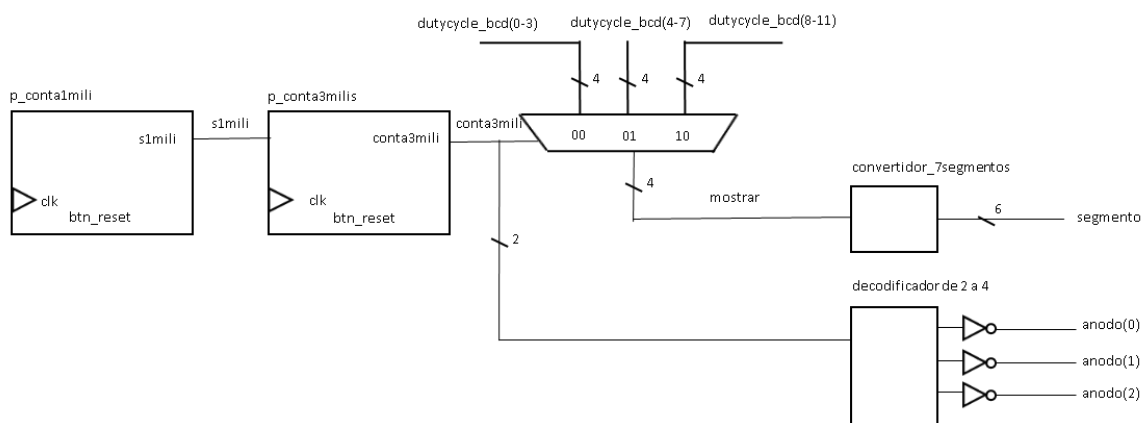


Figura 6.12: Esquema del diseño digital para hacer lucir el display de la FPGA

En este trabajo, dado que sólo se necesita usar 3 displays, se multiplexa en 3 partes. Además, se añadió un multiplexor para la selección de cada uno de los dígitos. Su señal de selección es el vector del interruptor, de manera que si el interruptor está activado, muestre durante medio segundo el dígito y medio segundo no muestre nada. Con esto se consigue que mientras el interruptor correspondiente a su dígito esté activo, ese dígito parpadeará para saber que se puede cambiar. Para ello fue necesario además incluir otro contador de 0,5 segundos.