



Universidad
Carlos III de Madrid

*Titulación: Grado en Ingeniería Electrónica y
Automática*

PROYECTO DE FIN DE GRADO

Departamento de Ingeniería Automática y Sistemas

*“DISEÑO DE UNA MANO
ROBÓTICA PARA USO DOCENTE”*

Autor: Jonathan Pazmiño Espinoza

Tutor: Álvaro Villoslada Peciña

Director: Luis Enrique Moreno Lorente

MADRID – LEGANÉS 2014

Dedicatoria

A mis padres y hermana.

Agradecimientos

A mis padres, por su apoyo incondicional durante toda la carrera así como en todas mis metas, este trabajo va dedicado en especial para ellos.

A mi hermana que ha creído en mí, aun cuando ni yo lo hacía.

A mis amigos Wilmer Sergio y Arlin con los que hemos compartido grandes momentos tanto dentro como fuera de la universidad.

A mi novia Jessica, por estar a mi lado apoyándome y animándome en todas mis decisiones.

A mi tutor Álvaro Villoslada, por su dedicación e inigualable ayuda en la realización de este proyecto.

Y por último al Dr. Luis Moreno Llorente por brindarme la oportunidad de trabajar en este proyecto.

Resumen.

El trabajo que se desarrolla a continuación es para uso docente, en el cual se presenta el diseño y la construcción de una mano robótica sencilla de tres dedos antropomórficos.

El objetivo del presente trabajo es el de obtener un primer prototipo de este manipulador robótico sencillo y de bajo coste para que los alumnos aprendan a programar dicho dispositivo de una forma sencilla mediante *Simulink*.

Otro de los propósitos es enseñar al alumno que mediante sencillos métodos de fabricación se pueden diseñar dispositivos funcionales, capaces de emular alguno de los movimientos de la mano humana, sin necesidad de utilizar tecnologías complejas.

La mano robótica que se ha construido consta de tres dedos antropomórficos, cada uno de los cuales está constituido de tres falanges, cuyos movimientos están acoplados mediante un sistema mecánico rígido de transmisión. El movimiento de cada uno de los dedos se controla a través de dos tendones en configuración agonista-antagonista conectados a un servomotor.

Los dedos del manipulador robótico están pensados y diseñados con medidas reales y formas similares a los de la mano humana, para así conseguir movimientos verosímiles a los de una mano real.

El diseño, tanto de los dedos como de la palma de la mano y del resto de componentes mecánicos que conforman el diseño final se ha realizado mediante el software *Solidworks*.

La fabricación de cada parte del dedo, palma de la mano, soportes varios se han realizado mediante la impresión 3D. El principal objetivo de utilizar este método de fabricación ha sido disminuir los costes para el desarrollo del prototipo, ya que para su uso docente se necesitan varias unidades.

Para concluir se puede decir que este diseño presenta una alternativa de bajo coste y permite la actuación y control de una mano artificial con un número pequeño de grados de libertad.

Abstract

The work being done below is for teaching purposes, in which the design and construction of a simple three anthropomorphic robotic hand fingers is presented.

The aim of this study is to obtain a first prototype of this robot manipulator (simple and low cost) for students to learn how to program the device in a simple way by Simulink.

Another purpose is to teach them that by simple manufacturing methods, they can design functional devices, capable of emulating any of the movements of the human hand, without the use of complex technologies.

The robot hand has been constructed with three anthropomorphic fingers; each one consists of three phalanges, whose movements are mechanically coupled by a rigid drive system. The movement of each finger is controlled by two tendons in agonist-antagonist configuration connected to a servomotor.

The fingers of the robotic manipulator are conceived and designed with real measures and similar to those of the human hand, so it could be possible to get real hand movements.

The design of both, the fingers and the palm of the hand, and the other mechanical components that make up the final design was performed using the Solidworks software.

The manufacture of each of the finger, palm, and various supports was performed using 3D printing. The main objective of using this method of manufacture has been to reduce prototype's costs.

In conclusion it can be said that this design is a low cost alternative and allows actuation and control in an artificial hand with a small number of degrees of freedom.

Índice

Resumen.....	i
Abstract	ii
CAPÍTULO 1	1
INTRODUCCIÓN.....	1
1.1. Motivación.....	2
1.2. Objetivos	3
1.3. Planteamiento del problema	4
1.4. Restricciones del diseño.....	5
1.5. Estructura del documento.....	6
CAPÍTULO 2	7
2. Estado del Arte	7
2.1 Manipulación de objetos y aplicaciones.	7
2.1.1 Mano Utah/MIT.....	8
2.1.2 BarrettHand.....	9
2.1.3 Mano Robonaut	10
2.1.4 Shark Hand	11
2.1.5 Shadow Hand	12
2.1.6 Bebionic Hand	14
2.2 Docencia y Robótica	14
2.2.1 Lego Wedo.....	15
2.2.2 Lego Mindstorms.....	16
2.2.3 Ollo Inventor.....	17
2.2.4 Arduino.....	17
2.2.5 BIOLOID	18
2.2.6 DARWIN-OP.....	19
2.2.7 Robots industriales ABB	20
CAPÍTULO 3	23
3. Diseño Mecánico	23
3.1. Biomecánica	24
3.2. Diseño mecánico de los dedos	28

3.3.	Diseño mecánico de la palma de la mano y soportes.....	33
3.4.	Construcción de la mano robótica	37
3.5.	Método de fabricación.....	38
3.6.	Actuadores	39
3.7	Placa de interfaz.....	41
CAPÍTULO 4	44
4.	Sistema de control.....	44
4.1.	Placa de control STM32 DISCOVERY	45
4.2.	Sistema de control implementado en Simulink	50
CAPÍTULO 5	55
5.	Aplicación Docente.....	55
5.1.	Práctica ejemplo: Configuración del modelo de control en Simulink.....	55
CAPÍTULO 6	78
6.	Conclusiones finales y trabajos futuros	78
6.1.	Sugerencias para trabajos futuros	80
Bibliografía	81
ANEXOS	84

Índice de Figuras

<i>Figura 2.1 1 a) Mano Barret b) Agarre de la mano Barret</i>	9
<i>Figura 2.1 2 Representación de la transmisión del movimiento de la mano Barret</i>	10
<i>Figura 2.1 3 Mano Robonaut</i>	11
<i>Figura 2.1 4 Mano Shark Hand</i>	12
<i>Figura 2.1 5 Mano Shadow Hand</i>	13
<i>Figura 2.1 6 Mano Bebionic Hand</i>	14
<i>Figura 2.2. 1 Lego Wedo</i>	16
<i>Figura 2.2. 2 Lego Mindstorms</i>	16
<i>Figura 2.2. 3 Ollio Inventor</i>	17
<i>Figura 2.2. 4 Arduino</i>	18
<i>Figura 2.2. 5 Robots Bioloid</i>	19
<i>Figura 2.2. 6 Robots Darwin-OP</i>	20
<i>Figura 2.2. 7 Robots industriales ABB</i>	21
<i>Figura 3.1. 1 Huesos de mano vista dorsal</i>	25
<i>Figura 3.1. 2 Articulaciones involucradas en la cadena cinemática del dedo</i>	25
<i>Figura 3.1. 3 Flexión y Extensión de un dedo</i>	26
<i>Figura 3.1. 4 Tendón flexor y extensor del dedo</i>	28
<i>Figura 3.2 1 Articulaciones del dedo mecánico y medidas</i>	30
<i>Figura 3.2 2 Vista del enrutamiento de los tendones</i>	31
<i>Figura 3.2 3 Cara exterior tendonal</i>	32
<i>Figura 3.2 4 Cara exterior de agarre</i>	32
<i>Figura 3.2 5 Articulaciones del dedo mecánico</i>	33
<i>Figura 3.3. 1 Palma de la mano</i>	34
<i>Figura 3.3. 2 Palma de la mano vista de la planta</i>	35
<i>Figura 3.3. 3 Base servo-motores</i>	36
<i>Figura 3.3. 4 Base servo-motores dividida</i>	36
<i>Figura 3.3. 5 Carcasa exterior</i>	37
<i>Figura 3.5. 1 Modelado por deposición fundida</i>	38
<i>Figura 3.6 1 Servomotor Futaba S3003</i>	40
<i>Figura 3.7. 1 Regulador de tensión fija tri-terminal</i>	41
<i>Figura 3.7. 2 Circuito regulador de tensión fija</i>	42
<i>Figura 3.7. 3 Diseño de la placa de interfaz</i>	43
<i>Figura 4.1. 1 Placa STM32F4 Discovery (ST Microelectronics)</i>	45
<i>Figura 4.1. 2 Descripción de componentes y pines</i>	47
<i>Figura 4.2. 1 Bloque Target Setup</i>	51
<i>Figura 4.2. 2 Bloque USB VCP Receiver STM32F4</i>	52
<i>Figura 4.2. 3 Bloque Fcn</i>	52
<i>Figura 4.2. 4 Bloque UC3M Basic PWM</i>	53
<i>Figura 4.2. 5 Bloque Saturation</i>	53
<i>Figura 4.2. 6 Bloque Slider Gain</i>	54

<i>Figura 5. 1 Simulink</i>	56
<i>Figura 5. 2 Bibliotecas Simulink</i>	57
<i>Figura 5. 3 Ventana de modelado</i>	57
<i>Figura 5. 4 Comandos de la ventana de modelado</i>	58
<i>Figura 5. 5 Target Setup</i>	59
<i>Figura 5. 6 Modelo target setup</i>	60
<i>Figura 5. 7 Parámetros Target Setup</i>	60
<i>Figura 5. 8 Target Setup</i>	62
<i>Figura 5. 9 USB VCP Receiver STM32F4</i>	63
<i>Figura 5. 10 Parámetros USB VCP Receiver STM32F4</i>	64
<i>Figura 5. 11 Bloque Memory</i>	65
<i>Figura 5. 12 Parámetros del bloque Memory</i>	66
<i>Figura 5. 13 Modelo Control_PWM</i>	66
<i>Figura 5. 14 Bloque Fcn</i>	67
<i>Figura 5. 15 Parámetros de la función</i>	68
<i>Figura 5. 16 Bloque Saturation</i>	68
<i>Figura 5. 17 Parámetros del bloque de Saturation</i>	69
<i>Figura 5. 18 UC3M Basic PWM</i>	70
<i>Figura 5. 19 Parámetros del bloque UC3M Basic PWM</i>	71
<i>Figura 5. 20 Modelado de señales PWM</i>	72
<i>Figura 5. 21 Puerto COM del microcontrolador</i>	73
<i>Figura 5. 22 Bloque Host Serial Setup</i>	73
<i>Figura 5. 23 Bloque Slider Gain</i>	74
<i>Figura 5. 24 Degrees</i>	74
<i>Figura 5. 25 Bloque Host Serial Tx</i>	75
<i>Figura 5. 26 Configuración Host Serial Tx</i>	76
<i>Figura 5. 27 Modelo Host de control</i>	77

Índice de tablas

<i>Tabla 2.1. 1 Característica de la mano Utah</i>	9
<i>Tabla 2.1. 2 Características de la mano Barret</i>	10
<i>Tabla 2.1. 3 Características de la mano Robonaut</i>	11
<i>Tabla 2.1. 4 Características de la mano Shark Hand</i>	12
<i>Tabla 2.1. 5 Características de la mano Shadow Hand</i>	13
<i>Tabla 3.1. 1 Flexión y Extensión de un dedo</i>	27
<i>Tabla 3.2. 1 Característica del tensor para la mano robótica</i>	31
<i>Tabla 3.7. 1 Características Regulador de tensión</i>	42
<i>Tabla 4.1 1 Comparación entre STM32F4 Discovery, Arduino Uno y Raspberry Pi</i>	49
<i>Tabla A. 1 Desglose por fases</i>	86
<i>Tabla A. 2 Tabla de Gantt</i>	87
<i>Tabla A. 3 Consumo de Plástico ABS</i>	88
<i>Tabla A. 4 Costes del Plástico ABS</i>	89
<i>Tabla A. 5 Costes componentes y dispositivos Electrónicos</i>	90
<i>Tabla A. 6 Costes totales de materiales</i>	90
<i>Tabla A. 7 Costes de fabricación en impresora 3D</i>	91
<i>Tabla A. 8 Costes totales de fabricación</i>	91
<i>Tabla A. 9 Costes totales de personal</i>	91



CAPÍTULO 1

INTRODUCCIÓN

El ser humano siempre ha buscado mejorar su forma de vida y protegerlo. Actualmente existen actividades que el hombre no puede desarrollar, ya sea por limitaciones físicas o por ambientes hostiles que ponen en riesgo su integridad.

Las aplicaciones robóticas se han expandido rápidamente desde aplicaciones industriales clásicas con tareas repetitivas a aplicaciones con una estrecha interacción humano-robot. En las últimas décadas se ha invertido mucho tiempo en investigar en el ámbito de la robótica humanoide, en particular, la robótica asistencial ha ganado una creciente atención. De hecho, la adaptación de los servicios de salud a las necesidades de la población dependiente tendrá un gran impacto en el desarrollo de los dispositivos robóticos que tratan de emular el comportamiento de las diferentes extremidades del hombre.

La robótica no tiene únicamente aplicación en la industria o en trabajos en ambientes hostiles para el ser humano. Una de las variantes de la robótica que más se está



desarrollando en los últimos años es la llamada *wearable robotics*. Estos son robots que se "ponen" en el usuario, como los exoesqueletos o las prótesis robóticas.

En este mismo ámbito se dirige la ingeniería biomédica, disciplina la cual aplica los principios eléctricos, electrónicos, mecánicos, químicos, o cualquier otro principio de la ingeniería, para comprender, modificar o controlar los sistemas biológicos, así como para diseñar y fabricar productos capaces de monitorizar funciones fisiológicas [1].

A pesar de los grandes avances que se han logrado, no se ha conseguido construir un sistema que logre emular la destreza de la mano al 100%, debido a que la mano está constituida por una compleja estructura mecánica y un complejo sistema sensorial que le da características únicas y difíciles de emular por la robótica.

1.1. Motivación

El diseño de la mano mecánica antropomórfica se desarrolla con el objetivo de ser utilizada para las prácticas de Aplicaciones de la Automática en la Biomédica, para que el alumno pueda realizar el control mediante *Simulink* y realizar sencillos movimientos.

Los alumnos que cursan dicha asignatura, tienen como objetivo esencial conocer las diferentes aplicaciones de la automática en distintos campos, pero sobre todo en el campo médico (prótesis y ortesis). Por otra parte también se abordaran los aspectos relacionados con el control de estos sistemas robotizados que interactúan con el cuerpo humano, así como los diferentes tipos de sensores y actuadores [2].

Con estas necesidades, se desarrolla el proyecto que se presenta en este documento, con el objetivo de que el alumno pueda comprender mejor el funcionamiento y el control con un ejemplo real de una sencilla mano robótica, y que puedan ver las limitaciones y ventajas que presentan en un marco más práctico.

Para el control de la mano robótica diseñada se ha optado por la utilización del *Software Simulink*, que es una plataforma proporcionada por Matlab. Simulink es un



entorno de diagramas de bloque para la simulación multidominio y el diseño basado en modelos. Admite el diseño y la simulación a nivel de sistema, la generación automática de código y la prueba y verificación continuas de los sistemas embebidos. La utilización de Simulink fue elegida porque proporciona un entorno gráfico interactivo y un conjunto de librerías de bloques personalizados que permiten diseñar, simular, implementar y probar una gran variedad de sistemas con variación temporal, entre los que incluyen sistemas de control, procesamiento de señal, vídeo e imagen. Simulink ofrece acceso a numerosas herramientas que permiten desarrollar algoritmos, analizar y visualizar simulaciones, personalizar el entorno de modelaje y definir señales, parámetros y datos de pruebas [3].

La finalidad de la utilización de este software es que el alumno pueda controlar un sistema robótico real, y que gracias a las librerías diseñadas por el Sr Antonio Flores Caballero del laboratorio de Robótica y Automática de la Universidad Carlos III de Madrid, puede programarse un microcontrolador muy potente, como es el microcontrolador STM32F4 Discovery, de forma gráfica y sencilla, sin necesidad de usar un lenguaje de programación textual.

1.2. Objetivos

Primero se realizará un estudio de las manos robóticas, su desarrollo a lo largo del tiempo y las manos existentes hoy en día, para extraer sus características y de esta manera saber cuáles han de ser las especificaciones que se deben tomar en cuenta a la hora del diseño del prototipo.

También se analizarán los diferentes métodos e instrumentos que se utilizan para la enseñanza de la robótica en la actualidad para distintas edades y etapas educativas, y así estudiar el papel que cumpliría nuestra prótesis en las prácticas de la asignatura “*Aplicaciones de la Automática en la Biomédica*”.

El objetivo del presente proyecto es el desarrollo de un manipulador robótico antropomórfico de bajo coste, que se desarrolla con el objetivo de ser utilizada para



las prácticas de la asignatura “Aplicaciones de la Automática en la Biomédica”, para que el alumno pueda realizar el control mediante el *software Simulink* y realizar sencillos movimientos.

Para el propósito planteado se han seguido las siguientes fases:

- Diseño mecánico de la mano
- Construcción física de la mano
- Diseño de la etapa de potencia para alimentar los actuadores.
- Desarrollo de las estrategias de control mediante simulink para mover las articulaciones de los dedos.

1.3. Planteamiento del problema

El problema que se nos plantea en este proyecto es la necesidad de diseñar una mano robótica para la utilización en las prácticas de la asignatura *Aplicaciones de la Automática en la Biomédica*.

Se ha de proponer un diseño factible para su fabricación y construcción. Que dicho manipulador robótico pueda simular movimientos reales de los dedos. Proponer un sistema de control cognoscible e intuitivo para el estudiante, un sistema de control que pueda estimular al alumno a manifestar o rebatir diversas alternativas de control. Por último utilizar un software primordial y asequible que aporte importantes conocimientos en la programación de la mano robótica. Optar por una enseñanza más práctica que involucre más al alumno y utilizar la *robótica docente* como medio para ello.



1.4. Restricciones del diseño

El diseño que se va a implementar presenta una serie de restricciones tanto en movimientos como el coste de la fabricación del prototipo de la mano robótica, lo que nos condiciona el material que se ha de utilizar y el proceso de fabricación. Todas estas restricciones se deben tomar muy cuenta para el buen desarrollo del proyecto.

La mano humana es una de las extremidades más complejas del hombre lo cual hace muy complicado emular todos los movimientos que puede realizar. Para simplificar, y debido a que el objetivo principal del proyecto es diseñar un dispositivo con el cual el alumno tenga su primer contacto con simulink para programar y controlar el movimiento de una pequeña prótesis, la mano robótica diseñada contará únicamente con tres dedos y los movimientos que podrá realizar serán únicamente la flexión y extensión de los mismos, movimientos suficientes para poder realizar un correcto agarre de objetos.

Al ser un dispositivo que está dedicado a la realización de prácticas, es necesario fabricar suficientes unidades para los alumnos de dichas prácticas, por lo que el coste de cada dispositivo tiene que ser lo más reducido posible. Con este propósito se ha optado por la fabricación con una impresora 3D, que utiliza la metodología de fabricación por deposición fundida. Dicha impresora es capaz de crear piezas o maquetas volumétricas, por superposición de capas sucesivas de material, a partir de un diseño previo realizado por ordenador [4].

La fabricación mediante la impresora 3D es idónea para la realización del proyecto por su velocidad de fabricación y por el material que utiliza, que es económicamente asequible. El material comúnmente utilizado para este tipo de impresiones, es el plástico crilonitrilo butadieno estireno “ABS”. Es un termoplástico, un plástico muy resistente al impacto lo que lo hace muy útil para el proyecto. Entre otras aplicaciones es muy utilizado en automoción y otros usos tanto industriales como domésticos.

1.5. Estructura del documento

La organización del trabajo de fin de grado se detallará a continuación.

- ✚ *Capítulo 2 Estado del Arte.* Se realiza un repaso de las manos robóticas existentes, estudiando sus componentes, partes que las componen, grados de libertad entre otras características, cuyo estudio resultará muy útil para el diseño de la mano que se va a desarrollar en el presente proyecto.
- ✚ *Capítulo 3 Diseño Mecánico.* En este capítulo se estudia la anatomía de la extremidad superior, concretamente la mano, con el objetivo de establecer las especificaciones para el diseño. Por otra parte se detalla la construcción de la mano robótica, sus actuadores y el método de fabricación empleado para cumplir los objetivos del proyecto.
- ✚ *Capítulo 4 Sistema de control implementado en simulink.* El objetivo de este capítulo es explicar el control de los actuadores de la mano que se ha desarrollado para el microcontrolador STM32F4 Discovery e implementado con simulink.
- ✚ *Capítulo 5 Práctica de ejemplo.* Se muestra una práctica “modelo” para que el alumno pueda desarrollar los conocimientos teóricos adquiridos en la asignatura y una guía útil para el manejo de simulink para el movimiento de la prótesis.
- ✚ *Capítulo 6 Conclusiones finales y trabajos futuros*
- ✚ *Capítulo 7 Anexo.* Se aportan las datasheets de cada uno de los componentes que utilizamos, así como planos del dedo por completo.



CAPÍTULO 2

2. Estado del Arte

2.1 Manipulación de objetos y aplicaciones.

Sobre habilidades de manipulación con manos robóticas hay un extenso desarrollo desde hace décadas, pero en estos últimos años ha habido un gran avance en la investigaciones en este ámbito. Este avance ha sido debido, entre otras cosas, porque las aplicaciones de la robótica son inmensamente amplias como sucede en la actualidad, con la robótica industrial, y en un futuro no muy lejano, se verá reflejado en la vida cotidiana o en la medicina.

A continuación se detallarán cronológicamente las diferentes manos robóticas que se han desarrollado a lo largo de los años y se han construido a lo largo del mundo.

Debido al gran número de manos robóticas existentes, este apartado se limitará a un estudio de los dispositivos más significativos.

2.1.1 Mano Utah/MIT

La mano Utah fue desarrollada por el Instituto de Tecnología de Massachusetts en 1985. Esta mano consta de cuatro dedos con cuatro grados de libertad cada uno. Cuenta con un antebrazo donde se colocan los actuadores neumáticos que son los encargados de mover cada una de las articulaciones mediante unos tendones conducidos por poleas. Para medir la posición de la mano se emplean sensores de efecto hall montados en las diferentes articulaciones de la mano.

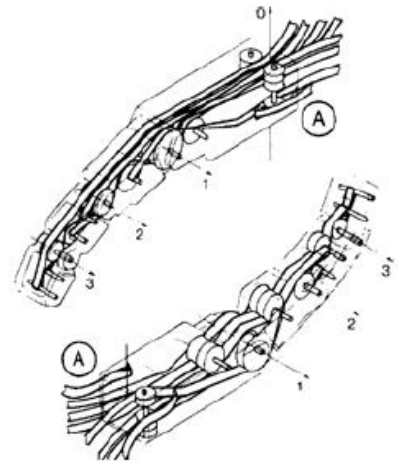


Figura 2.1.1 a) Fotografía mano UTAH b) Representación de la transmisión mediante poleas

El diseño contiene grandes complicaciones en el sistema de control de la mano, ya que para mover la mano con cierta regularidad, había que crear un complicado plan para mover las 288 poleas del sistema, por lo que los límites y confiabilidad del sistema son impredecibles [5].

Tabla 2.1. 1 Característica de la mano Utah

Número de dedos	4
Grados de libertad	16
Tipo de actuadores	Actuadores neumáticos
Tipo de transmisión	Tendones

2.1.2 BarrettHand

La barretthand fue diseñada y construida por la empresa Barrett Technology en 1988. Fue una de las primeras manos que tuvieron un gran éxito comercial [5]. Una de las razones por la que consiguieron tal éxito fue que estaba compuesto por ocho ejes altamente flexibles para el agarre, los cuales pueden ser reconfigurados en tiempo real para dar seguridad en el agarre de la amplia variedad de formas de objetos que puede manipular, como se puede ver en la siguiente figura.

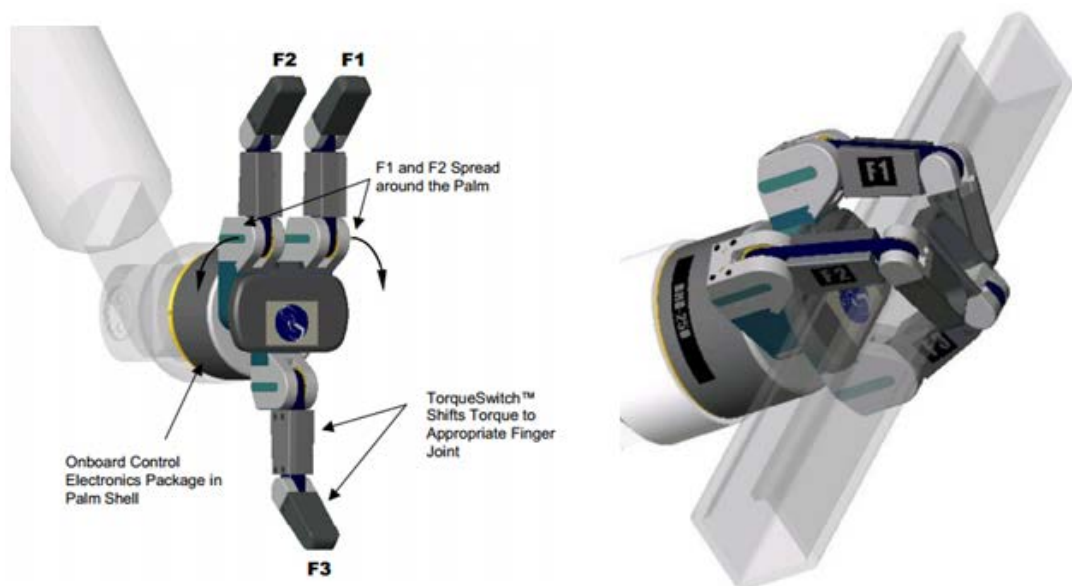


Figura 2.1 1 a) Mano Barret b) Agarre de la mano Barret

El prototipo consta de tres dedos con nueve grados de libertad, tres en cada uno de ellos. Los actuadores eran una serie de motores brushless que transmitían su movimiento por medio de tornillos sin fin y engranajes como se muestra en la Figura 2.1 2

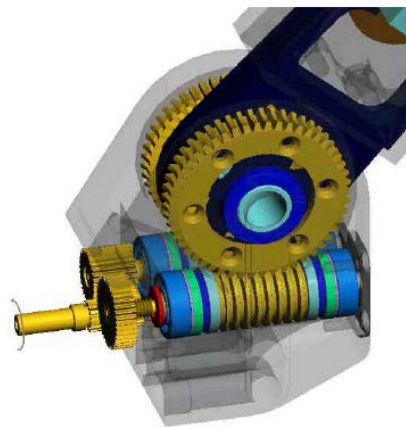


Figura 2.1 2 Representación de la transmisión del movimiento de la mano Barret

Tabla 2.1. 2 Características de la mano Barret

Número de dedos	3
Grados de libertad	9
Tipo de actuadores	Motores eléctricos
Tipo de transmisión	Transmisión por engranajes

2.1.3 Mano Robonaut

La mano Robonaut fue desarrollada en 1999 en los laboratorios de la NASA en Estados Unidos. Esta mano consta de cinco dedos con 19 grados de libertad, los cuales son manejados por motores brushless que se encuentran colocados en un antebrazo para transmitir la potencia a cada uno de los dedos.



Figura 2.1 3 Mano Robonaut

Las señales de realimentación para el control de posición y fuerza son proporcionadas por encoders incrementales, sensores táctiles y celdas de carga [5]. Esta mano ha ido evolucionando con los años y su versión actual está en uso en el Robonaut de la Estación Espacial Internacional.

Tabla 2.1. 3 Características de la mano Robonaut

Número de dedos	5
Grados de libertad	14
Tipo de actuadores	Motores eléctricos
Tipo de transmisión	Ejes rígidos

2.1.4 Shark Hand

Esta mano fabricada por Intel está basada en el diseño de la mano Barretthand de la cual hablamos en el apartado 2.1.2. Al igual que esta, cuenta con tres dedos reconfigurables en tiempo real. Lo que diferencia a esta mano de las demás es su sistema táctil que puede sentir la forma de los objetos. Las puntas de sus dedos

envían un débil impulso eléctrico que interfiere con objetos, dando la mano una idea aproximada de lo que se trata de captar [5].

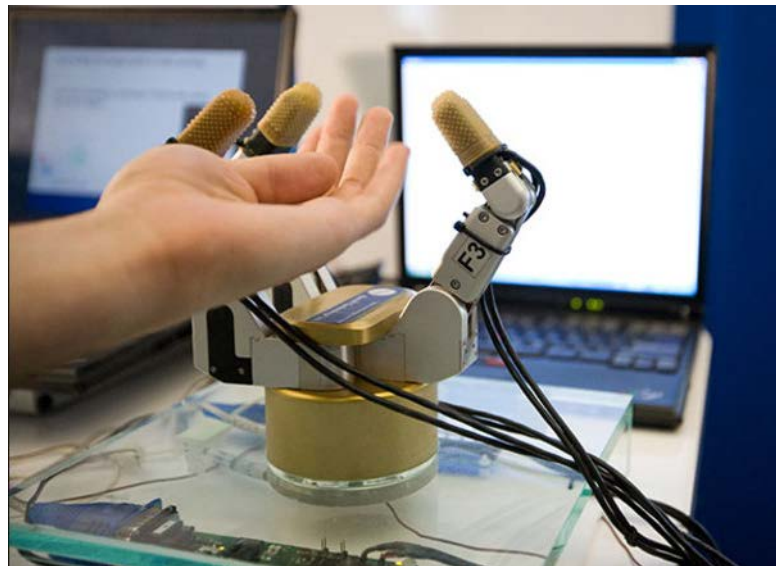


Figura 2.1 4 Mano Shark Hand

Tabla 2.1. 4 Características de la mano Shark Hand

Número de dedos	3
Grados de libertad	9
Tipo de actuadores	Motores eléctricos
Tipo de transmisión	Arrastre de los engranajes

2.1.5 Shadow Hand

Shadow Robot Company en el Reino Unido en el año 2002 desarrolló una de las manos robóticas más avanzadas del mercado, la Shadow Hand. El sistema constituido por cinco dedos y 24 grados de libertad, utiliza músculos neumáticos

y una red tendonal con lo que busca reproducir lo más fielmente posible los movimientos de la mano humana. La información acerca de la posición de los dedos es proporcionada por una serie de sensores de efecto-hall integrados en las articulaciones. También cuenta con sensores de presión en cada músculo. En 2009, Shadow lanzó una versión de la Shadow Hand que utiliza como actuadores 20 motores DC [5].



Figura 2.1 5 Mano Shadow Hand

Tabla 2.1. 5 Características de la mano Shadow Hand

Número de dedos	5
Grados de libertad	24
Tipo de actuadores	Neumáticos
Tipo de transmisión	Tendones

2.1.6 Bebionic Hand

La mano Bebionic es una mano protésica de las más desarrolladas en el mercado. Una de las características que destaca esta mano sobre las demás es que se puede configurar de forma inalámbrica, indicándole la velocidad de actuación, la fuerza o la manera en que debe agarrar un objeto esto le da una gran versatilidad y funcionalidad. Utiliza motores individuales en cada dedo lo que le permite mover la mano y el agarre de una manera natural y coordinada. Otro de los avances que presenta esta mano es la utilización de microprocesadores potentes que monitorean continuamente la posición de cada dedo, que le permite un control preciso y fiable sobre los movimientos de la mano [6].



Figura 2.1 6 Mano Bebionic Hand

2.2 Docencia y Robótica

La robótica ha tenido un fuerte desarrollo en estas últimas décadas, la robótica personal y de servicio pronto será cotidiana en nuestros hogares. Por este motivo es esencial familiarizar a las nuevas generaciones con este tipo de dispositivos desde

muy temprana edad. Encontramos en la Robótica Educativa, una herramienta muy importante, pensada para que niños y jóvenes desarrollen su capacidad en distintas áreas tales como [7]:

- ✚ Intelectual
- ✚ Biopsicomotora
- ✚ Lógico-matemáticas
- ✚ Comunicación integral

Mejorando además:

- ✚ Su autoestima y confianza en sí mismo
- ✚ Habilidades para solucionar problemas
- ✚ Capacidad de atención y memoria

Para instruir la práctica de la robótica se han desarrollado diferentes metodologías y herramientas, en las que intenta mezclar el juego con la robótica, sobre todo a tempranas edades. Un ejemplo de las herramientas docentes que se utilizan en la enseñanza de la robótica se ilustra a continuación:

2.2.1 Lego Wedo

Una herramienta extremadamente fácil y divertida para iniciarse en la robótica, construyendo modelos con sensores simples y un motor que se conecta a un ordenador, y programando comportamientos. Ideal para desarrollar un aprendizaje colaborativo y participativo.



Figura 2.2. 1 *Lego Wedo*

2.2.2 **Lego Mindstorms**

Solución completa de aprendizaje a partir de 8 años. Permite a los estudiantes descubrir la programación controlando dispositivos reales de entrada y salida. Su lenguaje de programación visual posibilita una funcionalidad muy avanzada a la vez que intuitiva.



Figura 2.2. 2 *Lego Mindstorms*

2.2.3 Olo Inventor

Nuevo sistema de iniciación a la robótica, flexible, escalable, programable y educativo para diseñar y construir robots, jugando y despertando el interés por la ciencia y la tecnología.



Figura 2.2. 3 Olo Inventor

Para continuar el aprendizaje de la robótica en niveles más avanzados, como por ejemplo niveles universitarios, existen métodos y herramientas para ampliar los conocimientos dentro de la robótica, como por ejemplo placas con microcontroladores, que sirven para para construir y programar tus propios robots, o plataformas humanoides escalables y reconfigurables que permiten al alumno aprender lo básico de estructuras y principios de las articulaciones de robots. Por ejemplo tenemos:

2.2.4 Arduino

Es una plataforma electrónica abierta para la creación de prototipos basada en software y hardware fáciles de usar. Para construir tus propios robots y realizar

tus proyectos puedes combinar la plataforma arduino con sistemas mecánicos escalables y reconfigurables [8].

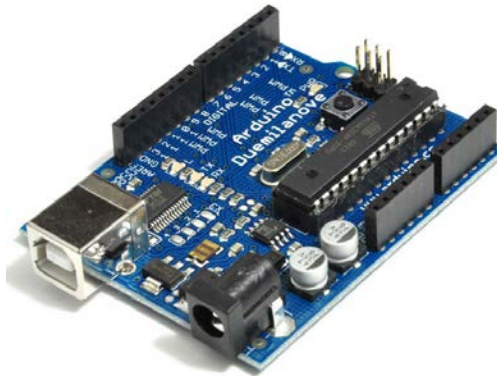


Figura 2.2. 4 Arduino

2.2.5 BIOLOID

Es una plataforma robótica educativa concebida de forma flexible, modular y escalable para construir tus propias creaciones robóticas. Ideal para educación y hobby, así como para competición e investigación. Con este sistema se aprende lo básico de estructuras y principios de las articulaciones de robots, programar comportamientos, trayectorias y ampliar su aplicación a la ingeniería creativa, cinemática inversa, y la cinética [9].



Figura 2.2. 5 Robots Bioloid

2.2.6 DARWIN-OP

Es una plataforma humanoide open source (software y hardware). Es uno de los modelos de humanoides a pequeña escala más avanzados. Relaciona distintos campos muy importantes en el aprendizaje del alumno, como la utilización de sensores para aplicaciones de visión artificial, inteligencia artificial, interacción y comunicación hombre-máquina, competiciones entre otras características [10].



Figura 2.2. 6 Robots Darwin-OP

2.2.7 Robots industriales ABB

Son los más utilizados en las grandes industrias, principalmente en la automoción, por la gran flexibilidad que aporta a la cadena de montaje [11]. Las universidades utilizan robots industriales de pequeño tamaño para enseñar a los alumnos los fundamentos básicos de la programación y control de este tipo de robots. También se emplean en áreas de fundición y pintura entre otras actividades.



Figura 2.2. 7 Robots industriales ABB

Como conclusión podemos decir que la robótica educativa intenta que la robótica sea una herramienta motivadora que permita la integración del uso de la tecnología en cualquiera de las áreas curriculares para el desarrollo de aprendizaje significativo en el alumno.

Cabe destacar que en la robótica educativa se puede utilizar una metodología docente cada vez más común llamado *Aprendizaje Basado en Proyectos*.

El Aprendizaje Basado en Proyectos es un método docente basado en el estudiante como protagonista de su propio aprendizaje. De los objetivos principales que persigue este método es mantener a los estudiantes comprometidos, motivados y estimular, el trabajo cooperativo, así como una mayor participación y dejar de lado la enseñanza mecánica y memorística. Utilizar la realización de proyectos como parte de las asignaturas no es un concepto nuevo y los docentes lo incorporan con frecuencia. Pero la enseñanza basada en proyectos es diferente: es una estrategia educativa integral (holística), en lugar de ser un complemento.

Este enfoque motiva a los estudiantes a aprender porque les permite seleccionar temas que les interesan, muestran un mayor compromiso y la motivación posibilita el alcance de logros importantes.



Los principales beneficios del aprendizaje basado en proyectos incluyen: preparar a los estudiantes para los puestos de trabajo, aumentar la motivación, hacer la conexión entre el aprendizaje en la escuela y la realidad, colaboración para construir conocimiento, aumentar las habilidades sociales y de comunicación, acrecentar las habilidades para la solución de problemas, permitir a los estudiantes tanto hacer como ver las conexiones existentes entre diferentes disciplinas, aumentar la autoestima, permitir que los estudiantes hagan uso de sus fortalezas individuales de aprendizaje y de sus diferentes enfoques hacia este, entre otros beneficios [12]



CAPÍTULO 3

3. Diseño Mecánico

La prótesis diseñada consta de tres dedos, dos de ellos iguales y un tercero simétrico y en oposición, pero los tres con similares propiedades para un correcto funcionamiento que detallaremos más adelante.

Aparte de los dedos se ha diseñado una base donde se colocan los dedos, la cual se denomina *palma de la mano*, la cual unimos con un soporte diseñado para la sujeción de los servos-motores, y por último una carcasa exterior para cubrir la parte de los servos-motores. Todas las características de cada parte del proyecto se detallarán con más precisión en apartados siguientes.

El principal objetivo en esta primera etapa es conseguir que la mano sea antropomórfica, sobre todo a la hora de diseñar los dedos. Para conseguir este objetivo es imprescindible considerar aspectos tales como el trabajo que va a realizar el prototipo, las condiciones a las cuales va a realizar el trabajo, tipos de actuadores que vamos a utilizar, velocidad a la que queremos que se realice, entre otros aspectos.

Los aspectos antes mencionados se deben analizar con detenimiento para sacar conclusiones necesarias y poder desarrollar un prototipo que cumpla con los planteamientos trazados; además se debe considerar el costo debido a que un diseño complejo y con mayor número de piezas requiere una mayor inversión económica.

En este capítulo se abordarán los detalles correspondientes al diseño, construcción y ensamblaje de la mano, así como pequeños detalles de la biomecánica de la mano humana útiles para la realización de un diseño bioinspirado.

3.1. Biomecánica

A continuación se realizará un breve estudio de la anatomía de la mano humana, poniendo especial énfasis en los dedos que es la principal parte que desarrollamos en este proyecto. Estos datos se tomarán en cuenta a la hora de definir y acotar los distintos parámetros del diseño mecánico de la mano robótica.

3.1.1. *Huesos y uniones de la mano*

La mano humana está compuesta por veintisiete huesos, de los cuales catorce huesos son falanges en los dedos. Los dedos están compuestos por tres tipos de falanges: falange distal, falange medial y falange proximal. Estos tres tipos de falanges son las encargadas de realizar los movimientos básicos para cada una de las actividades cotidianas [13].



Figura 3.1. 1 Huesos de mano vista dorsal

Para el diseño de la prótesis, se consideran estas tres falanges. La primera articulación corresponde a la unión metacarpal-falangeal (MCP) que va unida a la base del diseño, la segunda a la interfalangeal-proximal (PIP), y la tercera a la interfalangealdistal (DIP) del dedo ordenado como se muestra en la figura 3.1.2.



Figura 3.1. 2 Articulaciones involucradas en la cadena cinemática del dedo

3.1.2. Movilidad de los dedos

La Flexión es la curvatura o reducción del ángulo que forman partes del cuerpo y la Extensión, se define, por lo general, como el retorno de la flexión. Cuando la extensión excede el estado neutro, se denomina hiperextensión. El rango de movimiento de la flexión y extensión de los dedos, varía de unos dedos a otros. Los movimientos del dedo corazón son más amplios por lo que se tomarán esos valores como referencia a la hora de diseñar los dedos de la mano robótica [14].

En dicho dedo la flexión de la articulación metacarpal-falangeal (MCP) es de aproximadamente 90° , la interfalangeal-proximal (PIP) desde 100 a 110° , así como la interfalangealdistal (DIP) va desde los 60 a 70° .

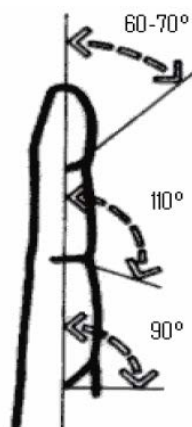


Figura 3.1. 3 Flexión y Extensión de un dedo

Todos los dedos de la mano humana tienen un aspecto similar por lo que los rangos de movimientos de cada articulación son parecidos, más no iguales. En este proyecto se han tomado valores iguales para cada dedo, los cuales se engloban en la Tabla 3.1. 1

Tabla 3.1. 1 Flexión y Extensión de un dedo

<i>Articulación</i>	<i>Movimiento</i>	<i>Rango de movimiento</i>
DIP	Flexión/Extensión	70°
PIP	Flexión/Extensión	110°
MCP	Flexión/Extensión	90°

3.1.3. Tendones de los dedos

El sistema de tendones que actúan en los dedos de la mano se clasifican en: tendones flexores que son los encargados de la flexión de los dedos, y los extensores que son los encargados de la extensión de los dedos (ver Figura 3.1. 4) Estos dos tipos de tendones se originan en el antebrazo. Nos centraremos en estos dos tipos de tendones, para el diseño de los movimientos del manipulador robótico.

Los flexores principales de las falanges y de la mano son el flexor profundo digitorum (FPD) y el flexor superficial digitorum (FSD). FPD se utiliza para doblar los dedos sin carga, mientras que el FSD se utiliza cuando se requiere de fuerza adicional. Los tendones de FPD están conectados con la falange distal, mientras que los tendones FSD están conectados con la falange medial [15]

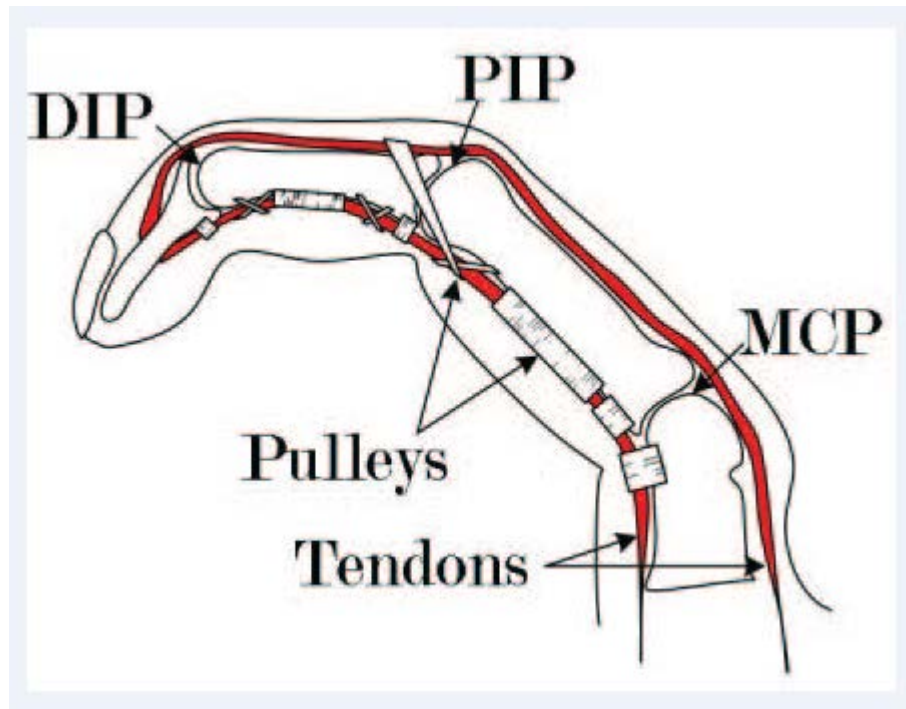


Figura 3.1. 4 Tendón flexor y extensor del dedo

Reproducir el sistema de tendones de la mano humana que hace posible el movimiento de los dedos, puede llegar a tener una configuración muy compleja, pero como los movimientos que se ha propuesto que realice el manipulador robótico diseñado (extensión y flexión) son menos complejos que los de una mano real se ha dispuesto una configuración agonista-antagonista de los tendones, que veremos más adelante con más detalles, para que los dedos diseñado tengan la movilidad deseada.

3.2. Diseño mecánico de los dedos

El diseño empleado para la fabricación de dos de los tres dedos que se han fabricado es idéntico y un tercero simétrico a los otros dos. Pero el diseño de los detalles de los tres dedos es semejante, por esta razón a continuación se presentan los detalles de construcción correspondientes a uno de ellos.



Las tres falanges de cada uno de los dedos forman una cadena cinemática de 3 grados de libertad, los cuales le permiten realizar diversos movimientos. Los movimientos de los actuadores son transmitidos a las falanges por medio de dos tendones en configuración agonista-antagonista.

El diseño de cada parte que compone el dedo se ha realizado en el *software Solidworks*. Solidworks es un programa de diseño asistido por ordenador para el modelado mecánico desarrollado en la actualidad por Solid-Works Corp. Es un modelador de sólidos paramétricos. El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la producción. Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en trasvasar la idea mental del diseñador al sistema CAD, construyendo virtualmente la pieza o conjunto [16]. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada.

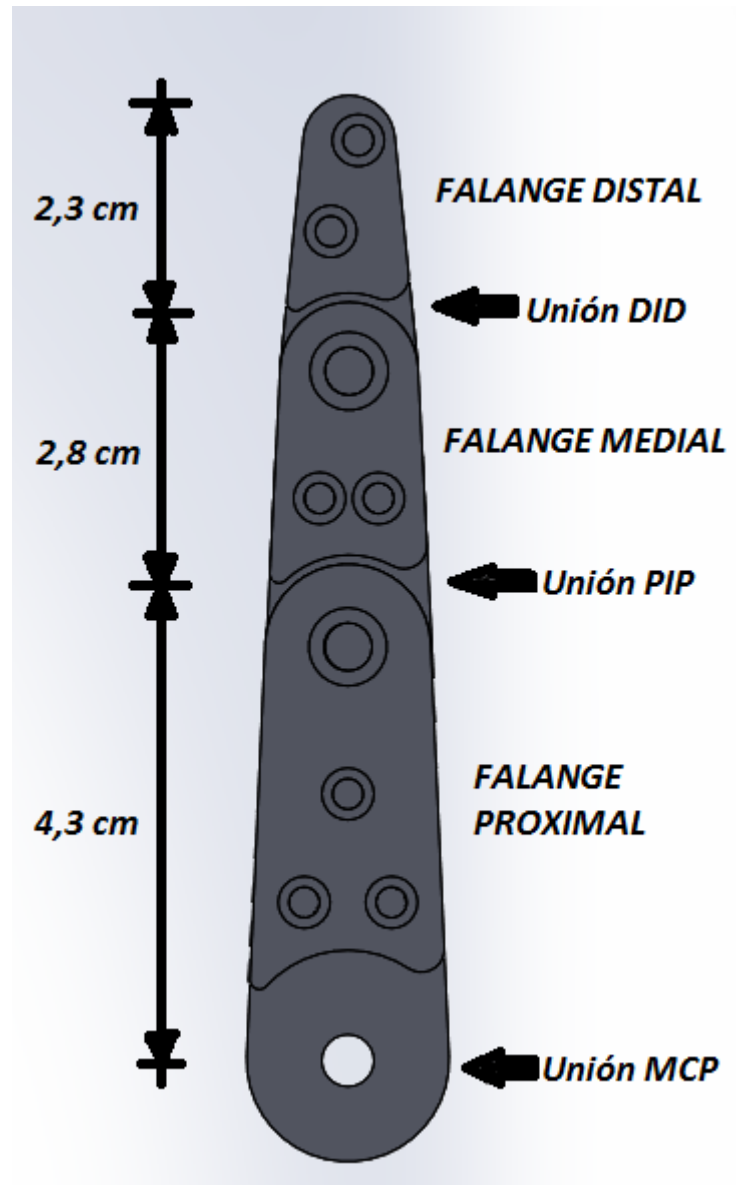


Figura 3.2 1 Articulaciones del dedo mecánico y medidas

3.2.1. Sistema de transmisión de movimiento de los dedos

El sistema de transmisión mecánico que se ha escogido para el movimiento de los dedos de la mano robótica es un enrutamiento de tendones colocados en una configuración *agonista-antagonista*, como se puede ver en la Figura 3.2 2

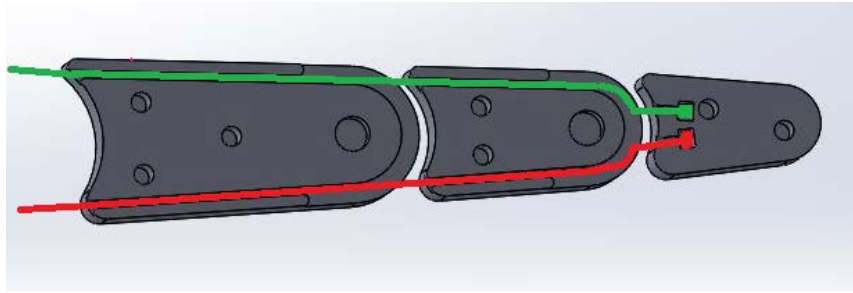


Figura 3.2 2 Vista del enrutamiento de los tendones

Este sistema de transmisión escogido es capaz de transmitir el movimiento a través de las falange hasta los servo-motores, al mismo tiempo proporcionar la potencia suficiente para realizar los movimientos. Para este fin, el material más apropiado por sus propiedades de tensión, flexibilidad y diámetro, es *el hilo de nylon*. En la Tabla 3.2. 1 se muestran las principales características del tensor que moverá las articulaciones de los dedos [17].

Tabla 3.2. 1 Característica del tensor para la mano robótica

<i>Diámetro</i>	0,45 mm
<i>Carga máxima de trabajo</i>	18 Kg
<i>Material</i>	Nylon

3.2.2. Descripción esquemática y funcional

Cada uno de los dedos diseñados consta de nueve piezas, se ha dividido cada falange del dedo en tres partes:

- ✚ ***Cara exterior tendonal:*** Es la cara por donde pasan los dos tendones, que son los encargados de transmitir el movimiento a cada una de las falanges.

Se han integrado dos pequeños enrutamientos en cada una de las falanges del dedo que sirven de guía para los tendones.

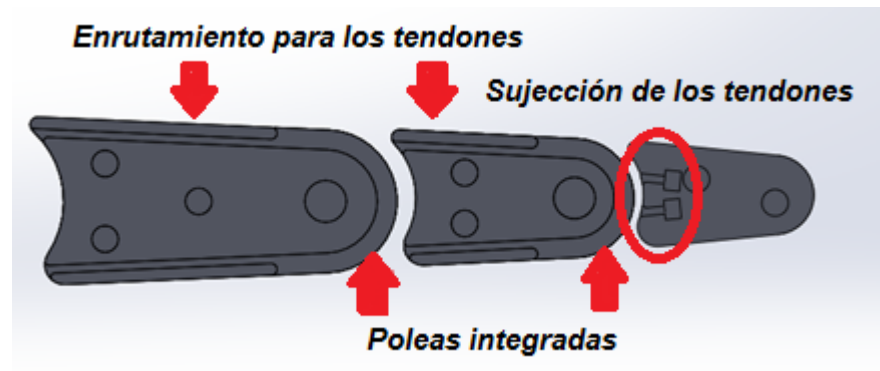


Figura 3.2 3 Cara exterior tendonal

Como se puede observar en la Figura 3.2 3, la *falange distal* tiene dos pequeños agujeros que hacen de punto de sujeción de los dos tendones. En las *falanges medial y proximal*, se diseñan semicírculos a modo de poleas integradas en la pieza para facilitar el movimiento de los tendones y disminuir lo máximo posible el rozamiento entre la pieza y el tendón.

- ✚ *Cara exterior de agarre*: Hace de sujeción para la cara exterior tendonal y el cuerpo medio.

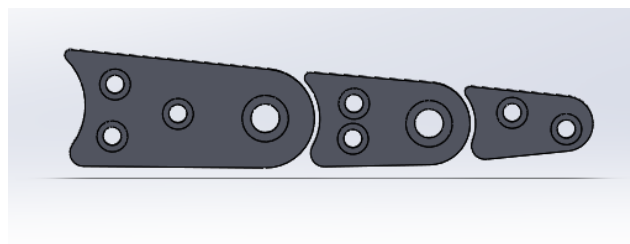


Figura 3.2 4 Cara exterior de agarre

✚ *Cuerpo medio:* Es la estructura central del dedo. Como se puede ver en la Figura 3.2 5, la falange distal y la falange medial cuentan con unos pequeños salientes cuyo cometido es el de restringir el rango articular en la extensión del dedo, bloqueando el movimiento.

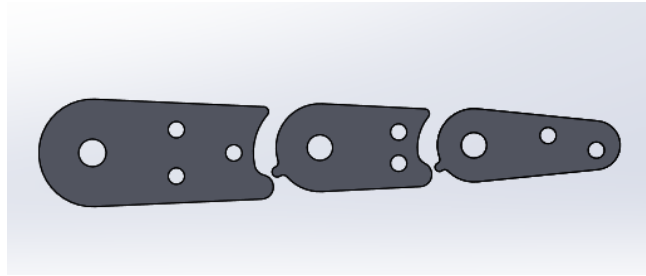


Figura 3.2 5 *Articulaciones del dedo mecánico*

3.3. Diseño mecánico de la palma de la mano y soportes.

3.3.1. Palma de la mano.

Para el diseño de la palma de mano, se ha optado por el diseño de una plataforma rectangular que pueda aportar estabilidad, mediante el agarre del dedo y los topes de giro, para cada uno de los movimientos.

Esta plataforma consta de tres articulaciones correspondientes a la articulación MCP de los tres dedos y por donde se guían los tendones hasta el agarre con los servomotores.

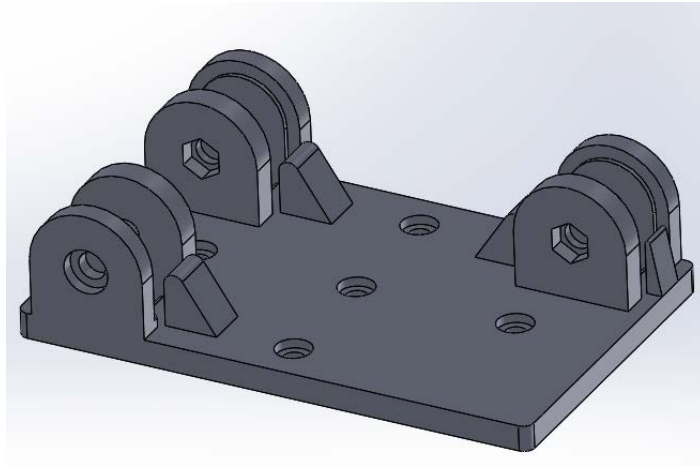


Figura 3.3. 1 Palma de la mano

Como se puede observar en la figura anterior, cada articulación cuenta con dos topes mecánicos para limitar el movimiento de los dedos, de tal manera que los movimientos de los dedos robóticos sean semejantes a los de una mano humana respetando los rangos de movimientos que vimos anteriormente en la Tabla 3.1.

1

En la siguiente figura se puede apreciar una vista de la planta de la palma, donde se observan detalles del diseño como los orificios por donde se enrutan los tendones para el posterior agarre con los servomotores o los huecos en la parte central de las articulaciones MCP para evitar el rozamiento de los dedos con la base.

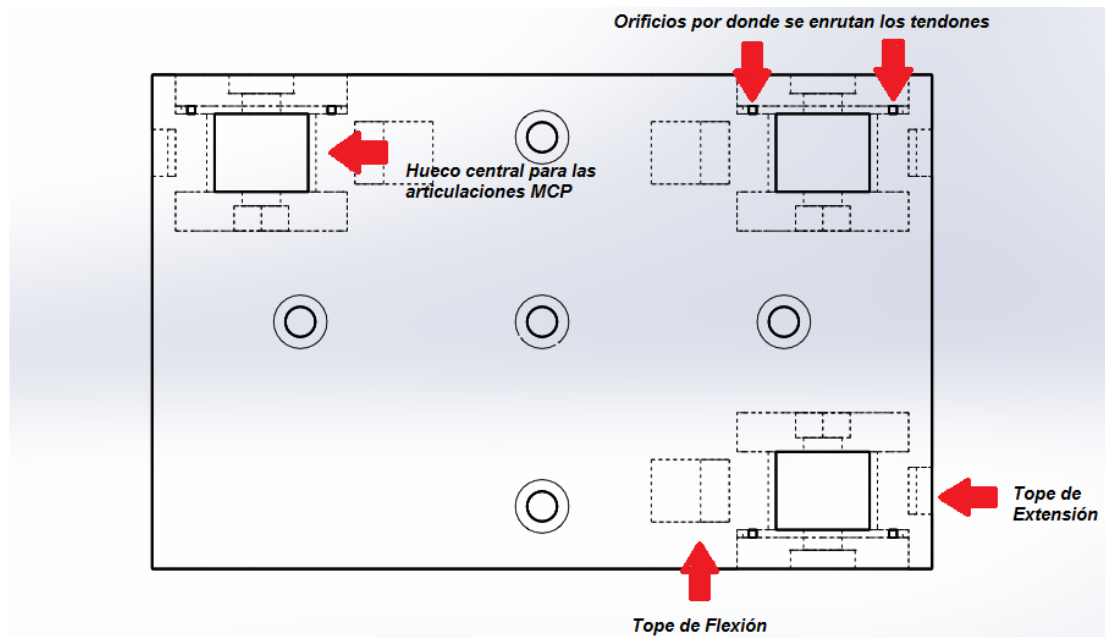


Figura 3.3. 2 Palma de la mano vista de la planta

3.3.2. Soporte de los servo-motores y carcasa.

Se ha diseñado un soporte donde se instalan los servo-motores. Dicha base para los servo-motores tiene que soportar pequeñas vibraciones, por lo que tienen que estar bien sujetos para evitar pérdidas de movimientos o un deterioro de los componentes.

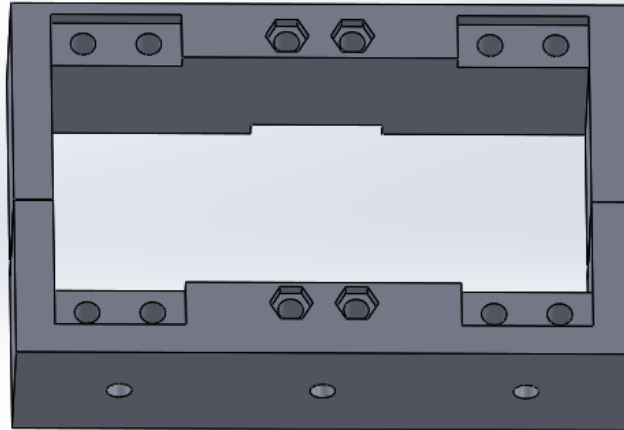


Figura 3.3. 3 Base servo-motores

Una de las limitaciones que se plantea en este diseño, es que al ser una estructura cerrada y ajustada a las medidas de los servomotores (para eliminar posibles vibraciones), la instalación de los servo-motores en sus respectivos huecos presenta algunas dificultades. Otra de las limitaciones, es debida a las dimensiones de la pieza que incrementan el tiempo de impresión, provocando algunos errores de fabricación. Por estas limitaciones se ha optado por dividir el soporte en dos partes, una parte con cuatro salientes redondeados, dos en cada lado, y la otra con cuatro agujeros del mismo diámetro para conseguir un encaje correcto de las dos piezas al ensamblar, como se muestra en la figura 3.3.4.

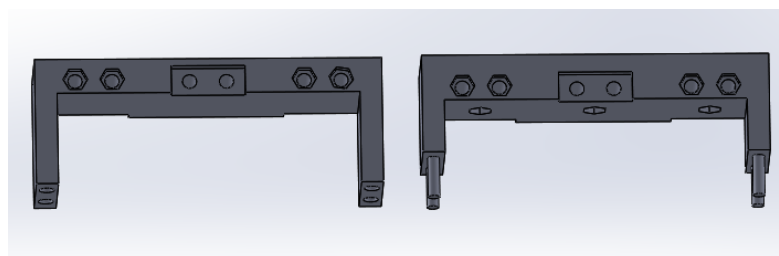


Figura 3.3. 4 Base servo-motores dividida

Por último se ha diseñado una carcasa exterior, para proteger toda la parte de los servos y el agarre de los tendones para evitar el deterioro de los servos o que puedan modificar el agarre de los tendones y por extra conseguir que el proyecto sea más estético. La carcasa tiene tres orificios en la parte trasera para permitir la salida de los cables de los servo-motores a la placa controladora, y en la parte superior dos orificios para atornillar la carcasa a la palma de la mano robótica.

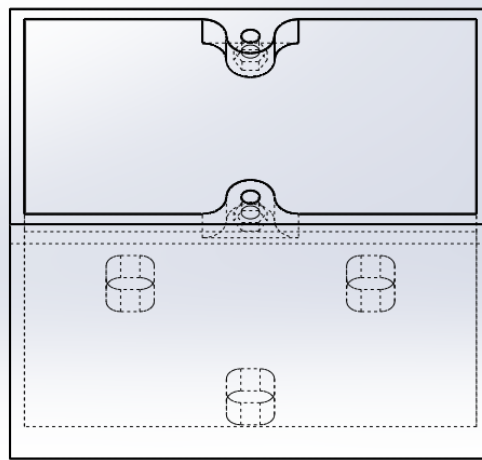


Figura 3.3. 5 Carcasa exterior

3.4. Construcción de la mano robótica

El software Solidworks aporta un entorno muy preciso a la hora del diseño, acabados casi perfectos y la simulación de propiedades físicas como el contacto mecánico entre distintas piezas de un ensamblaje.

Pero aun así para la construcción del prototipo se tuvieron que tener en cuenta los mecanismos de transmisión de movimientos, rozamientos entre las articulaciones, el acabado de las piezas, entre otros aspectos.

3.4.1. Planos SOLIDWORDS

En el anexo B se aportan los planos detallados de cada una de las partes del dedo así como las bases de los dedos y de los servomotores, para poder llevar a cabo su fabricación y posterior ensamblaje. Como se ha mencionado anteriormente el diseño de los tres dedos es similar por lo que solo se muestra los planos de uno de ellos.

3.5. Método de fabricación

El método empleado para la fabricación de cada una de las piezas del proyecto es la utilización de la *Máquina de prototipado rápido*, más conocida como “*impresora 3D*” que utiliza *el proceso de modelado por deposición fundida*. Este proceso utiliza una técnica aditiva de fabricación que consiste en depositar el material en capas, como se puede observar en la Figura 3.5. 1, para conformar la pieza [18]. Lo que le convierte en un proceso idóneo para el modelado de prototipos y la fabricación a pequeña escala.

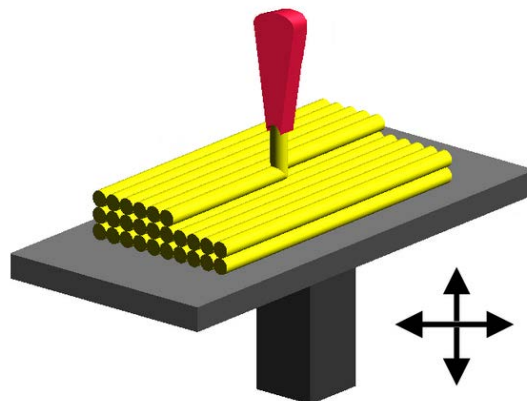


Figura 3.5. 1 Modelado por deposición fundida

La impresora 3D es una máquina que es capaz de realizar “impresiones” de piezas volumétricas a partir de un diseño realizado por ordenador. Una de las características por las que se ha optado por este método es porque permite pasar de un modelo virtual imprimible a una pieza física en cuestión de horas o incluso menos según el tamaño de la pieza a fabricar. Otra diferencia importante respecto a otras técnicas empleadas para la fabricación por inyección de plástico, es que este método de impresión 3D no necesita moldes, que pueden llegar a ser muy caros. Por último cabe destacar que la impresora 3D es capaz de imprimir con una precisión superior a una décima de milímetro, cuando se encuentra calibrada correctamente [4].

3.5.1. Propiedades Plástico ABS

Todos los componentes están elaborados por plástico acrilonitrilo butadieno estireno más conocido por plástico ABS, que es el utilizado por las impresoras 3D. El plástico ABS destaca por los bloques de acrilonitrilo que le proporciona rigidez y una gran estabilidad a altas temperaturas así como dureza. Los bloques de butadieno, que es un elastómero, proporcionan tenacidad a cualquier temperatura [19], por lo que la sinergia de estos dos materiales hace que este plástico sea idóneo para la realización de nuestro proyecto.

3.6. Actuadores

Para el movimiento de cada uno de los dedos se ha utilizado servomotores, ya que son muy sencillos controlar y tienen un consumo de energía reducido. La mano se diseña para la sujeción de pequeños objetos, pero sobre todo para que los alumnos tengan una primera toma de contacto con un dispositivo robótico relacionado con la asignatura, y para que adquieran habilidades programándolo y controlándolo con Simulink, por lo que el par de torsión que se necesita es pequeño. Con estas especificaciones se ha seleccionado el servomotor Futaba S3003 cuyas características esenciales [20] se presentan en la Tabla 3.6. 1:

Tabla 3.6. 1 Características servomotor Futaba S3003

Servomotor	Futaba S3003
Tensión nominal	6 V
Par máximo (6V)	4,1 kgm
Ángulo de funcionamiento	45 grados del punto inicial
Consumo de corriente (6V)	8 mA en reposo

El servomotor Futaba S3003 está conformado principalmente por un motor de corriente continua, una caja reductora y un circuito de control, como lo vemos reflejado en la siguiente imagen:

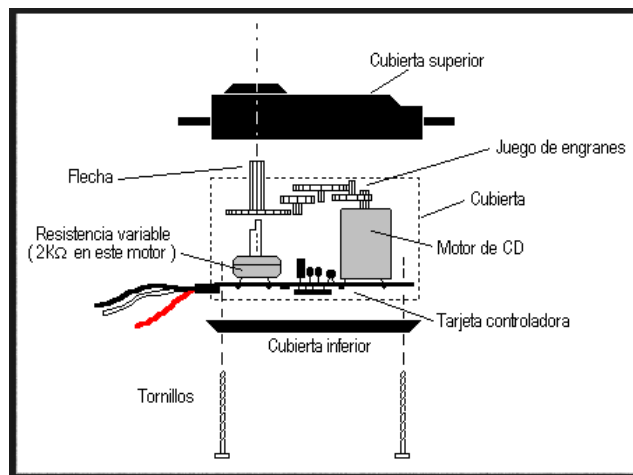


Figura 3.6 1 Servomotor Futaba S3003

Los servomotores hacen uso de la *modulación por ancho de pulsos (PWM)*, para controlar la dirección o llevar a los motores de corriente continua a posiciones angulares específicas.

3.7 Placa de interfaz

Como se ha visto en el apartado anterior, para alimentar los actuadores que se ha escogido para el movimiento de los dedos diseñados en este proyecto se necesita una tensión constante de 6V, independientemente de las variaciones de la corriente que circula por la carga y que no presente ningún tipo de rizado. Por este motivo se optó por utilizar un regulador de tensión.

La función del regulador de tensión utilizado es proporcionar una tensión específica y estable para alimentar a los actuadores a partir de una pila de 12V de entrada y aportar una mayor potencia a los servo-motores. Se optó por utilizar un *regulador de tensión fija tri-terminal*. Los reguladores de tensión típicos de tres terminales tienen un terminal para la entrada no regulada (IN), la salida regulada (OUT) y tierra (GND) y están ajustados para proporcionar una tensión de salida constante tal como $\pm 6V$ o $\pm 15V$.

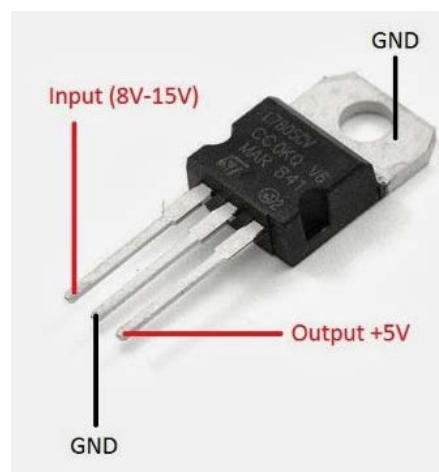


Figura 3.7. 1 Regulador de tensión fija tri-terminal

Dentro de esta categoría se encuentran la serie μA 78XX (positivos) o μA 79XX (negativos) de Fairchild. Los dos últimos dígitos indicados por XX indican la tensión de salida y pueden ser 05, 06, 08, 12, 15 V. El modelo que satisface todas las necesidades

del diseño propuesto es el μA 7806 el la Tabla 3.7. 1 se muestran las principales características.

Tabla 3.7. 1 Características Regulador de tensión

<i>V_{out}</i>	6V
<i>Temperatura de operación</i>	0 - 125 °C
<i>I_{max} de salida</i>	1A

La Figura 3.7. 2 describe el circuito regulador de tensión fija que se ha utilizado, con el regulador μA 7806, para obtener la tensión fija de +6V. Los condensadores C_1 y C_2 , mejoran la respuesta transitoria del regulador.

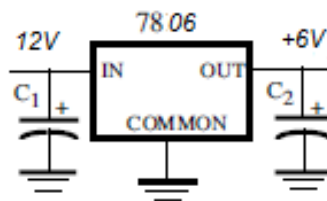


Figura 3.7. 2 Circuito regulador de tensión fija

Para la implementación del regulador se ha optado por el diseño de una placa de interfaz que se conectará al microprocesador como una “shield” de la placa STM32F4 en donde, aparte del regulador, se incluyen los conectores de los servo-motores y la conexión a alimentación.

Para el diseño de la placa de interfaz se eligió el *software* ORCAD. Este software es especializado para el diseño de circuitos electrónicos. Este software está dividido en cuatro aplicaciones principales que permiten realizar las siguientes operaciones [21]:

CAPTURE: Descripción del diseño, que puede ser realizada mediante esquema eléctrico o con el lenguaje de descripción hardware VHDL.

PSPICE: Simulación de circuitos analógicos, digitales y mixtos (analógico + digital).

LAYOUT: Realización de placas de circuito impreso.

EXPRESS/CAPTURE: Diseño de circuitos digitales con dispositivos lógicos programables y memorias.

La aplicación que nos interesa para la implementación de la placa de interfaz es *Layout* que permite el diseño de la PCB (Printer Circuit Board). En la Figura 3.7. 3 se puede ver el diseño final de la placa de interfaz.

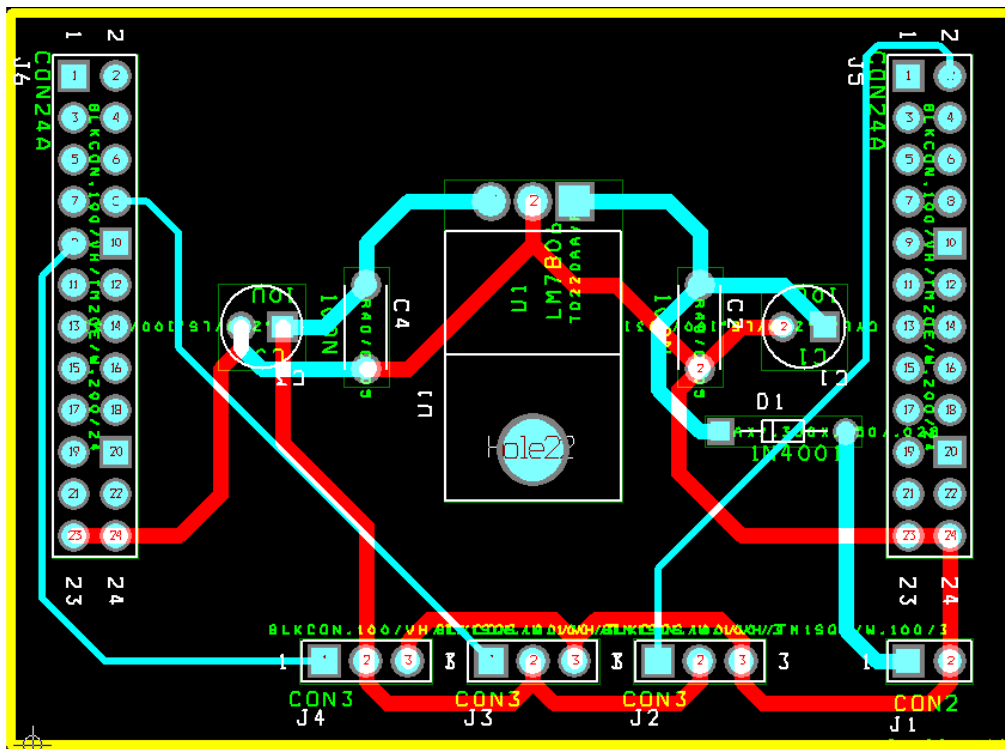


Figura 3.7. 3 Diseño de la placa de interfaz



CAPÍTULO 4

4. Sistema de control

En este capítulo se hablará de cómo se realiza el control implementado en Simulink del Manipulador Robótico que se ha diseñado en este proyecto. Se abordarán las principales características del microcontrolador elegido para dicha función, y como se genera la señal PWM (Pulse Width Modulation), necesaria para el correcto funcionamiento de los servo-motores.

El microcontrolador se encarga de dos tareas. Por un lado ha de comunicarse con el PC y por tanto debe ser capaz de recibir los datos que este le envía. Su otro cometido es generar las ondas PWM que accionan los servo-motores en función de los comandos de control recibidos del PC.

Los comandos que recibe el microcontrolador son los programados en Simulink. La programación en este entorno se ejecuta de forma gráfica mediante un diagrama de bloques. El software de Simulink brinda ventanas de elementos para su programación, y componentes que van enlazados para la lectura y recolección de datos. Mediante este software se pueden desarrollar diagramas de bloques para diferentes tipos de aplicaciones. Cada bloque posee su debida programación interna [22].

El microcontrolador utilizado es el *STM32F4 Discovery*. Se ha escogido este controlador por su alta gama de prestaciones y su bajo coste. En el siguiente apartado se hablará con más detalle de este microcontrolador.

4.1. Placa de control STM32 DISCOVERY

La serie STM32 está desarrollada por STMicroelectronics. Se trata de una familia de microcontroladores, basados en los núcleos ARM RISC de 32 bits de ARM Holdings. Algunos de estos núcleos son Cortex-M4F, Cortex-M3, Cortex-M0+ y Cortex-M0. La serie STM32F4 es versátil, fácil de utilizar, de bajo coste, y es la primera basada en ARM Cortex-M4F.

La arquitectura ARM son las siglas de *Advanced RISC Machine*. La arquitectura RISC (*Reduced Instruction Set Computer*) es un tipo de diseño de CPU que se utiliza generalmente en microcontroladores. A continuación se presentan algunas de sus características [23]: instrucciones de tamaño fijo, mostradas en un reducido número de formatos, disponen de muchos registros de propósito general y ejecutan instrucciones en paralelo y reducen los accesos a memoria

4.1.1. Características de la plataforma STM32 F4 Discovery

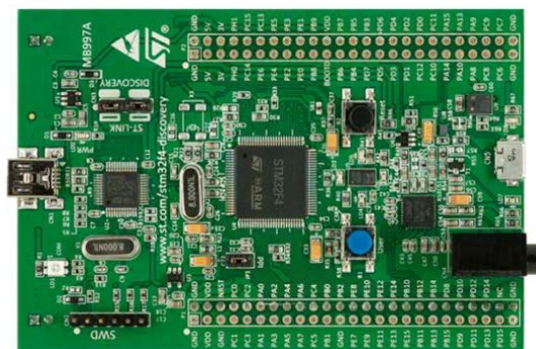


Figura 4.1. 1 Placa STM32F4 Discovery (ST Microelectronics)

A continuación se citarán las principales características de la placa STM32F4 *Discovery* [23]:

- *Core* ARM Cortex-M4F a una velocidad máxima de reloj de 180 MHz;
- 192 KB de RAM estática, 64 KB de CCM (del inglés *Core Coupled Memory*), 4 KB memoria de respaldo, 80 bytes de memoria de respaldo con detección de manipulación de borrado;
- Microcontrolador STM32F407VGT6 con 1 MB de memoria flash, que consta de 512/1024/2048 KB de uso general, 30 KB sistema de arranque, 512 bytes programables una vez (*One Time Programmable*), 16 bytes de opciones. Encapsulado LQFP100;
- Fuente de alimentación: a través del bus USB o desde una fuente de alimentación externa de 5 V.
- Sensor de movimiento ST MEMS LIS302DL, acelerómetro con salida digital de 3 ejes;
- Sensor de audio ST MEMS MP45DT02, micrófono digital omnidireccional;
- Ocho LEDs;
- Dos pulsadores (usuario y reset);
- USB OTG FS con conector micro-AB;

4.1.2. Descripción de componentes y pines

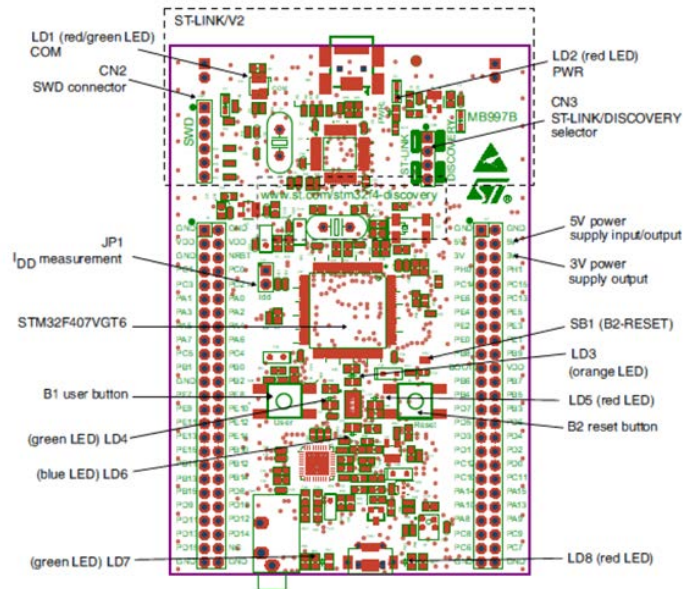


Figura 4.1. 2 Descripción de componentes y pines

- 16 DMA: canales de acceso directo a memoria
- 6 USART: Transmisor/receptor asíncrono síncrono universal
- 3 SPI: protocolo de datos en serie utilizado por microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas, o bien para la comunicación entre dos microcontroladores
- 3 I2C: es un bus de comunicaciones en serie. Su nombre viene de *Inter-Integrated Circuit* (Inter-Circuitos Integrados). Utilizado en la industria, principalmente para comunicar microcontroladores y sus periféricos integrados (*embedded systems*) y para comunicar entre sí circuitos integrados alojados en el mismo circuito impreso.
- 3 ADC: convertidor analógico- digital y 2 DAC: convertidor digital-analógico

- RTC: reloj en tiempo real, varios temporizadores
- Unidad CRC: inspector cíclico de redundancia
- Ethernet
- 2 *watchdogs* independientes: temporizador electrónico, cuyo funcionamiento es detectar y recuperarse de fallos.

4.1.3. Comparación con las plataformas Arduino Uno y Raspberry Pi

Es interesante realizar un estudio de las distintas plataformas de uso cotidiano para poder apreciar las distintas ventajas de la STM32F4 Discovery. A continuación se realiza una comparativa de las características más significativas

Arduino UNO

El Arduino UNO es un microcontrolador equipado con la placa ATmega328, posee 14 pines digitales de Entrada/Salida (seis de los cuales pueden ser usados como salidas PWM), seis entradas analógicas, un cristal que brinda una frecuencia de operación de 16 MHz, posee un puerto de conexión USB, cabecera ICSP y un botón de *reset*. Cuenta con el ATmega16U2 programado como un convertidor *USB-to-serial* [8].

Raspberry Pi

El *Raspberry Pi* es un sistema incrustado (del inglés *embedded*) que incluye un procesador central ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) y

512 MB de memoria RAM (modelo B). No posee disco duro ni memoria flash, puesto que utiliza una tarjeta SD como método de almacenamiento [24].

A continuación, a manera de comparación, se presenta un cuadro que resume los datos más importantes de cada una de las plataformas.

Tabla 4.1 1 Comparación entre STM32F4 Discovery, Arduino Uno y Raspberry Pi

<i>Características</i>	<i>STM32F4 Discovery</i>	<i>Arduino Uno</i>	<i>Raspberry Pi</i>
<i>Microcontrolador</i>	STM32F407VGT6	ATMega328	ARM1176JZF-S
<i>Memoria Flash</i>	1 MB	32 KB	SD exterior
<i>Memoria SRAM</i>	192 KB	1536 KB	512 MB
<i>Pines</i>	80 de propósito general	14 (6 PWM)	17 de propósito general
<i>Alimentación</i>	5 V (USB)	5 V (USB)	5 V (USB)
<i>Tensión de operación</i>	1,8 V- 3,6 V	5 V	3,3 V- 5 V
<i>Frecuencia de operación</i>	168 MHz	16 MHz	700 MHz
<i>Costo</i>	13,76 €	23,45 €	26,05 €

Como podemos ver el microcontrolador STM32F4 DISCOVERY es claramente superior en prestaciones al microcontrolador Arduino UNO. Sin embargo, la Raspberry Pi, pese a duplicar el coste, es mucho más potente tanto por velocidad

del microprocesador, como por memoria RAM o la inclusión de una GPU. Sin embargo, es un dispositivo más pensado como un ordenador de baja potencia que como una tarjeta controladora como es la Discovery.

Por las necesidades del proyecto, el microcontrolador STM32F4 DISCOVERY es superior debido a su mayor número de entradas/salidas y sobretodo porque algunas de estas salidas pueden programarse como salidas PWM, a diferencia del Raspberry Pi.

4.2. Sistema de control implementado en Simulink

Para poder realizar el correcto control de los servo-motores con el microcontrolador STM32F4 Discovery se realizaron dos diseños basado en *modelos* [25] complementarios en Simulink. Un modelo *host*, que es el ejecutable en el ordenador, y un segundo modelo que se programa en el microcontrolador. Cabe resaltar que una de las ventajas del sistema de control empleado es que el microcontrolador se programa mediante diagrama de bloques en Simulink, que luego se cargan en dicho microcontrolador, y no con lenguajes textuales habitualmente utilizados tipo C o C++.

Para realizar los dos modelados de control se necesitó añadir las librerías *WAIJUNG BLOCKSET* desarrollada por *Aimagin* [26] y complementada con la biblioteca *UC3M ADDONS STM32F4*, creada por el Sr Antonio Flores Caballero del departamento de Ingeniería de Sistemas y Automática, de la Universidad Carlos III de Madrid a las librerías ya existentes de Simulink [27]. Estas dos librerías son necesarias para el desarrollo del modelo de control.

A continuación se realizará una breve descripción de los principales bloques que se han utilizado para dicho propósito, en el apartado 5.1 se hablará con mayor particularidad la programación de cada uno de los bloques utilizados.

El primero y principal bloque que utilizamos es el “*Target Setup*”, que está dentro de la biblioteca *Waijung Blockset*, biblioteca desarrollada por *Aimagin*, para este tipo de microcontroladores. Este bloque se utiliza para configurar el STM32F4 en el modelo Simulink que se ha desarrollado. En la Figura 4.2. 1, se pueden observar diversas características, que se deben definir con anterioridad, como el compilador que utiliza (GNU ARM), el modelo del microcontrolador (STM32F407VGT6), el tiempo de muestreo, entre otras características.

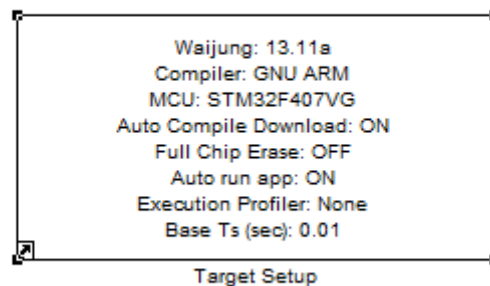


Figura 4.2. 1 Bloque Target Setup

Otro de los bloques importantes que se ha utilizado es el encargado de recibir los datos que se envían desde el ordenador al controlador. El bloque que se ha utilizado para este cometido es el *USB VCP Receiver STM32F4*.

Dentro de este bloque se ha de programar, el tamaño de máximo de datos, el tipo señales que se desea utilizar, cuál es la cabecera y la terminación de los mensajes que se envían, número de señales que se van a manejar a la salida del bloque y el tiempo de muestreo.

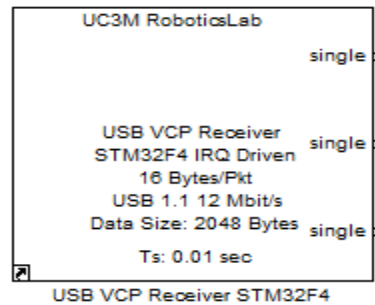


Figura 4.2. 2 Bloque USB VCP Receiver STM32F4

Como se observa en la Figura 4.2. 2; **Error! No se encuentra el origen de la referencia.** se programaron tres señales *single*, (una señal para el control de cada uno de los servo-motores). Las tres señales salientes corresponden a las posiciones angulares a las que se quiere girar los servo-motores. Mediante un algoritmo se convierten estas señales angulares en unos valores de ancho de pulso. Del algoritmo se hablará con más detalle en el apartado 5.1.1 (Paso 4).

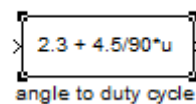


Figura 4.2. 3 Bloque Fcn

Por último se deben transformar estas señales en señales PWM. Para este cometido se utiliza el bloque *UC3M Basic PWM*, que se encuentra dentro de la biblioteca *UC3M ADDONS STM32F4*.

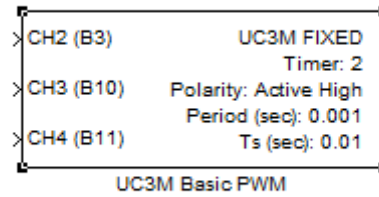


Figura 4.2. 4 Bloque UC3M Basic PWM

En este bloque es importante seleccionar los puertos del microcontrolador por los que se van a transmitir las señales PWM, y el periodo de la señal. El periodo que se introduce es 20 ms, este dato es aportado por el datasheet del servo-motor.

Las entradas que llegan a este bloque no deben superar los valores máximo y mínimo de ancho de pulso que aceptan los servomotores. Por este motivo se utilizan los bloques de *Saturation*, que limitan la señal de entrada en dichos valores, y así evitar un mal funcionamiento de los actuadores. Los valores que limitan la señal se han sacado de forma experimental.



Figura 4.2. 5 Bloque Saturation

En el modelo que se ha diseñado para controlar el ángulo de giro de los servo-motores, y que se ejecuta en el PC, cabe destacar el bloque *Slider Gain*, donde una de las propiedades que se impone es que varíe entre los valores angulares de 0 y 180, como se puede ver en la Figura 4.2. 6.

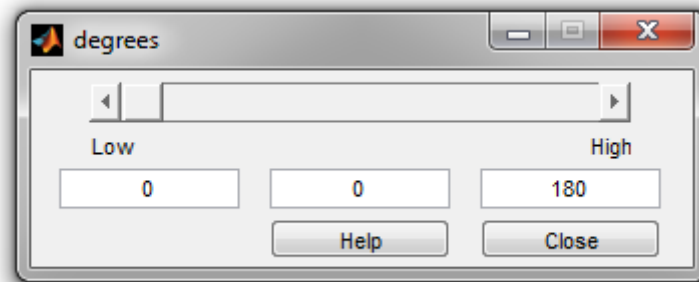


Figura 4.2. 6 Bloque Slider Gain



CAPÍTULO 5

5. Aplicación Docente

A continuación, se aportará un modelo sencillo de práctica docente para la asignatura *Aplicaciones de la Automática en la Biomédica*. El objetivo de esta práctica consistirá en el desarrollo de un *modelo de control del manipulador robótico* diseñado en este documento.

Antes de empezar, como expusimos en el apartado 4.2, para la utilización de los bloques que se han de utilizar para los diseños de los modelo de control, se requieren la instalación de las bibliotecas *WAIJUNG BLOCKSET* y *UC3M ADDONS STM32F4*

5.1. Práctica ejemplo: Configuración del modelo de control en Simulink

Para realizar el control del manipulador robótico se necesita implementar dos diseños de control basado en *modelos* en simulink [25]: un modelo que se ejecuta en el microcontrolador y otro que realiza el control desde el ordenador (host).

5.1.1. Programación del microcontrolador

Como mencionamos con anterioridad la programación del microcontrolador se realiza mediante diagramas de bloque implementado en Simulink. Para hacer un modelo con esta herramienta, el primer paso es acceder a Matlab y seleccionar en la ventana de comandos en el icono `>>Simulink` como se puede ver en la siguiente figura.

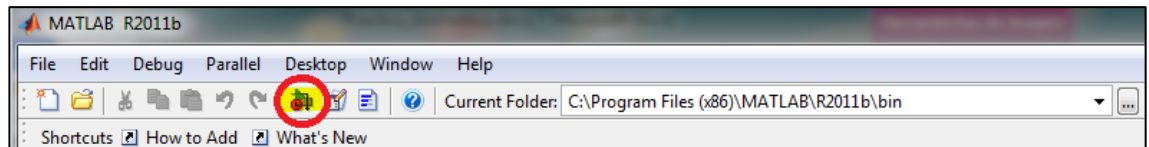


Figura 5. 1 Simulink

Aparecerá una ventana que contiene todas las librerías de bloques en el entorno de Simulink. En cada estante de la biblioteca se puede encontrar una serie de bloques del mismo tipo (ver Figura 5. 2).

Esta ventana está dividida en dos partes. La de la izquierda es la correspondiente a las librerías, y la de la derecha es el contenido de la librería seleccionada.

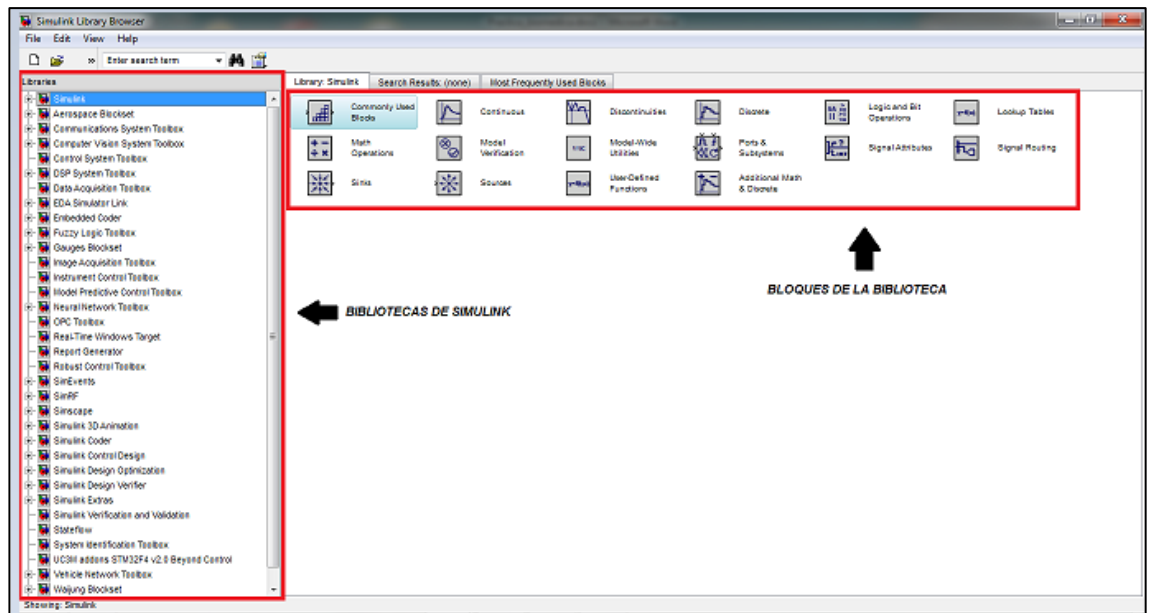


Figura 5. 2 Bibliotecas Simulink

Antes de crear el modelo con los bloques proporcionadas por Simulink, es necesario para crear una hoja de trabajo. Esto se logra siguiendo la secuencia:

FILE >> NEW >> MODEL

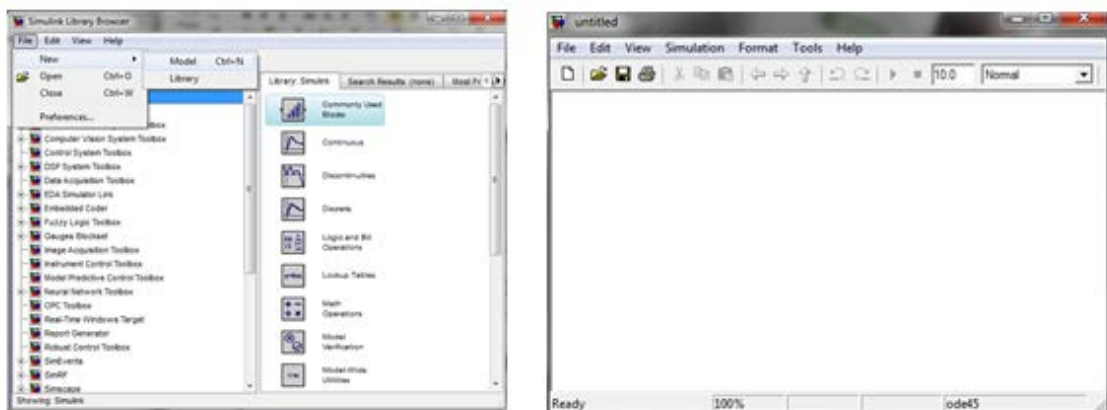


Figura 5. 3 Ventana de modelado

Se abrirá una ventana en blanco como se ve en la Figura 5. 3, en esta nueva ventana que aún no hemos dado nombre (untitled) se desarrollara el modelo de control.

En primer lugar se guardará el modelo con un nombre adecuado. Para ello, en el menú *File* de la nueva ventana elegimos la secuencia: **FILE >> SAVE**. Nos situamos en el directorio en el que se desea guardar a través del menú desplegable, ponemos un nombre al archivo, por ejemplo “*CONTROL_PWM*”. Los principales comando de nuestro archivo *CONTROL_PWM.mdl*, son el tiempo de simulación del modelo, los parámetros de simulación y el inicio de la simulación.

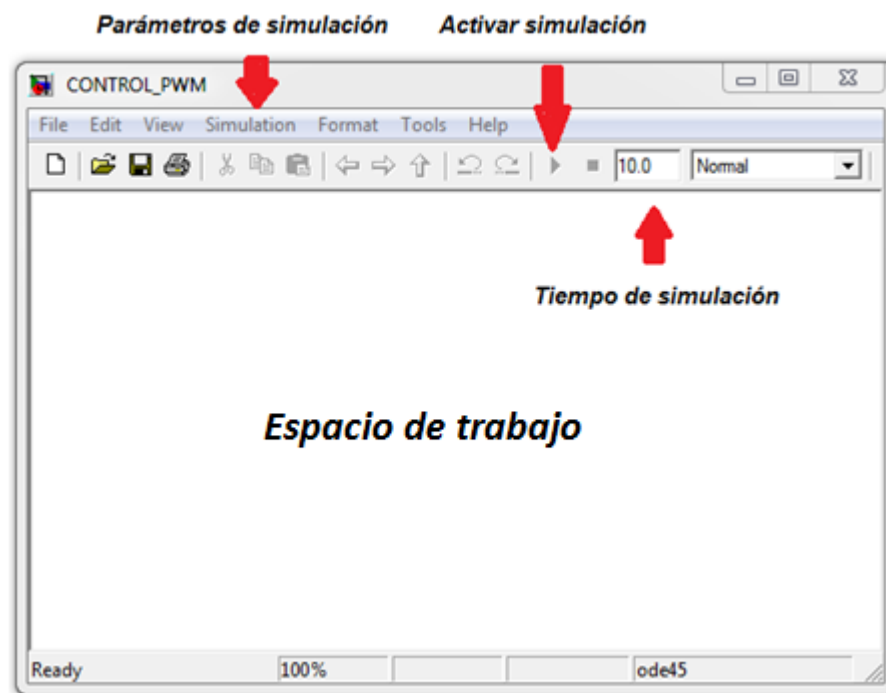


Figura 5. 4 Comandos de la ventana de modelado

En el espacio de trabajo se colocarán los diagramas de bloque del modelo. Para añadir un bloque, se hace “click” sobre la librería para desplegar los bloques, y se arrastran hasta nuestro espacio de trabajo.

Después de una breve introducción de la creación de un modelo en Simulink, se procederá a desarrollar un modelo para programar el microcontrolador.

Paso 1

Lo primero es configurar el Simulink que tipo de controlador se va a utilizar, así como el compilador que debe utilizar. Para ello se utiliza el bloque *Target Setup*. Este bloque se encuentra dentro de la biblioteca *Waijung Blockset*, siguiendo la secuencia: *Waijung Blockset* >> *STM32F4 Target* >> *Device Configuration* >> *Target Setup* (ver Figura 5. 5).

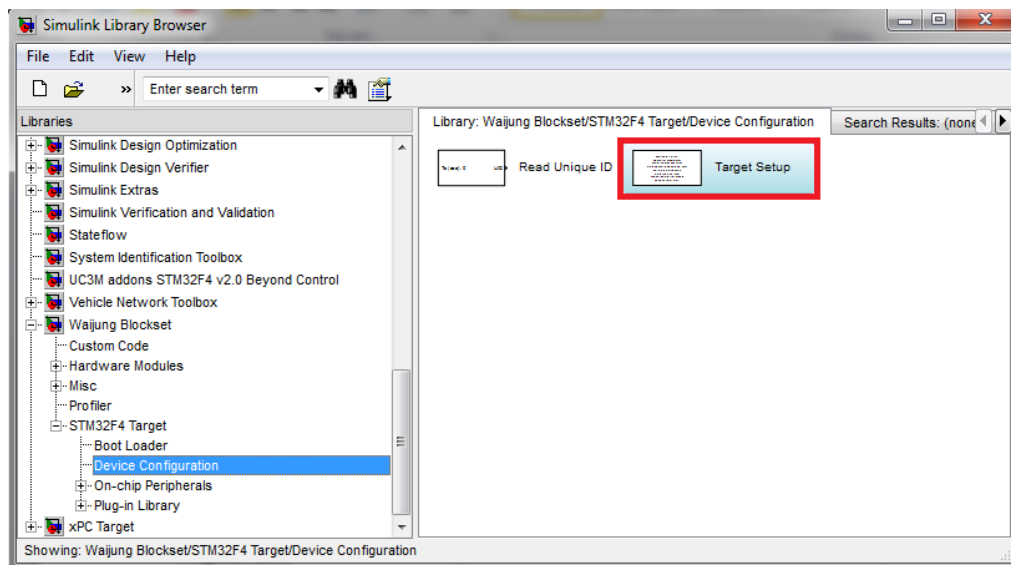


Figura 5. 5 Target Setup

Una vez se ha localizado el bloque, se arrastra hacia la ventana del modelo CONTROL_PWM.

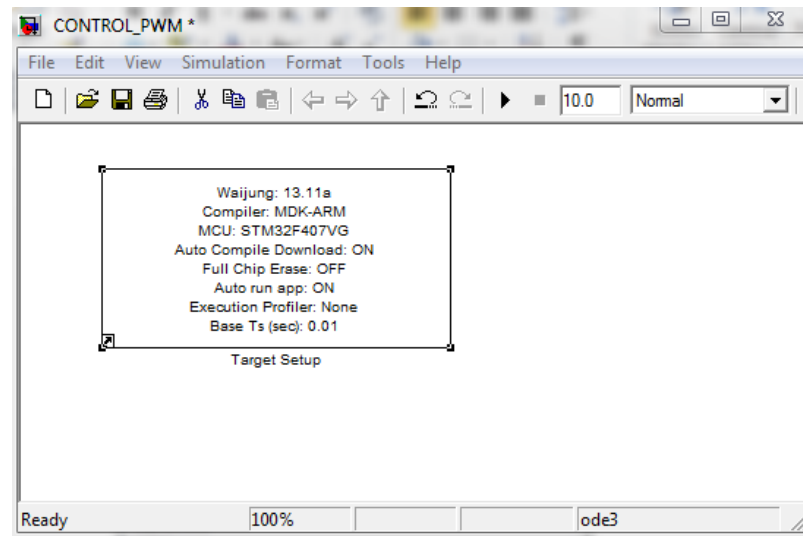


Figura 5. 6 Modelo target setup

Al hacer "doble click" en el bloque de configuración, se abre la ventana donde se seleccionan las distintas opciones de configuración del microcontrolador. (STM32F4 DISCOVERY). Para que el controlador funcione correctamente, el dispositivo debe configurarse como se muestra en las siguientes imágenes.

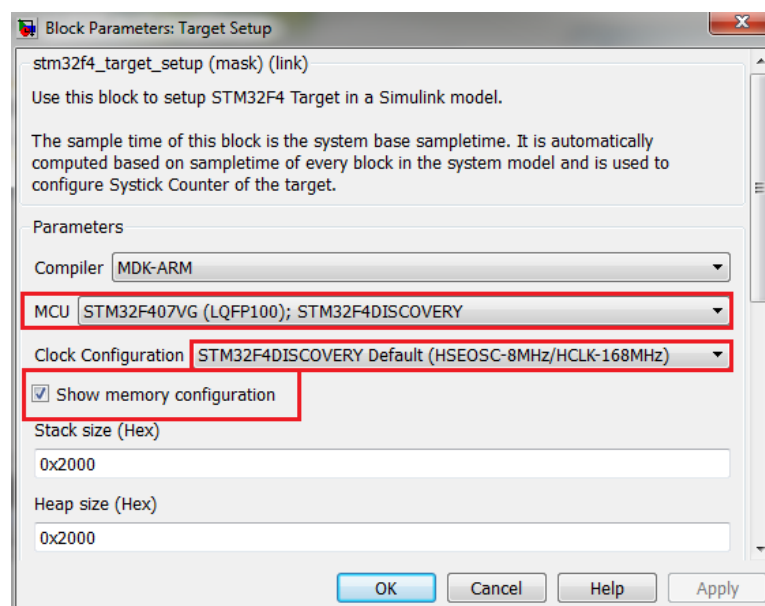


Figura 5. 7 Parámetros Target Setup

Como se puede observar en la Figura 5. 7, configuramos el compilador que se debe utilizar para el microcontrolador STM32F4 DISCOVERY (MDK-ARM) y en la pestaña inferior seleccionamos el microcontrolador.

Otro parámetro que se debe seleccionar en este bloque, es marcar la casilla “*show/ edit control string*”, nos muestra la cadena de control que utiliza el compilador, el ensamblador y el enlazador.

Compiler control string: *-c --cpu Cortex-M4.fp -g -O3 --apcs=interwork --split_sections -Otime \$(INCLUDES)*

Assembler control string: *--cpu Cortex-M4.fp -g --apcs=interwork --xref \$(INCLUDES)*

Linker control string: *--cpu Cortex-M4.fp --strict --scatter "\$(MODELNAME).sct" --summary_stderr --info summarysizes --map --xref --callgraph --symbols --info sizes --info totals --info unused --info veneers --list ".\$(MODELNAME).map" -o \$(MODELNAME).axf*

Y marcar la casilla *manually set base sample time*, para establecer el tiempo de la muestra (ver Figura 5. 8).

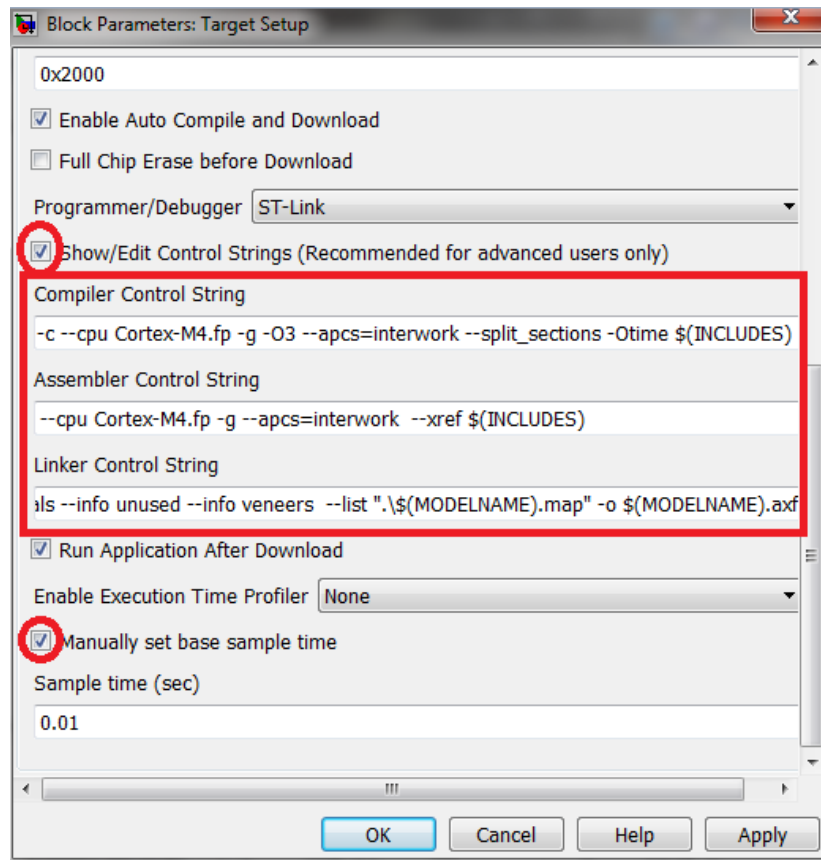


Figura 5. 8 Target Setup

Paso 2.

El siguiente bloque que se utiliza es el USB VCP Receiver STM32F4. Con este bloque vamos a predefinir el tipo de datos que se va a utilizar y el número de señales que se necesitan para controlar los dedos. Este bloque está dentro de la biblioteca UC3M ADDONS STM32F4 (ver Figura 5. 9).

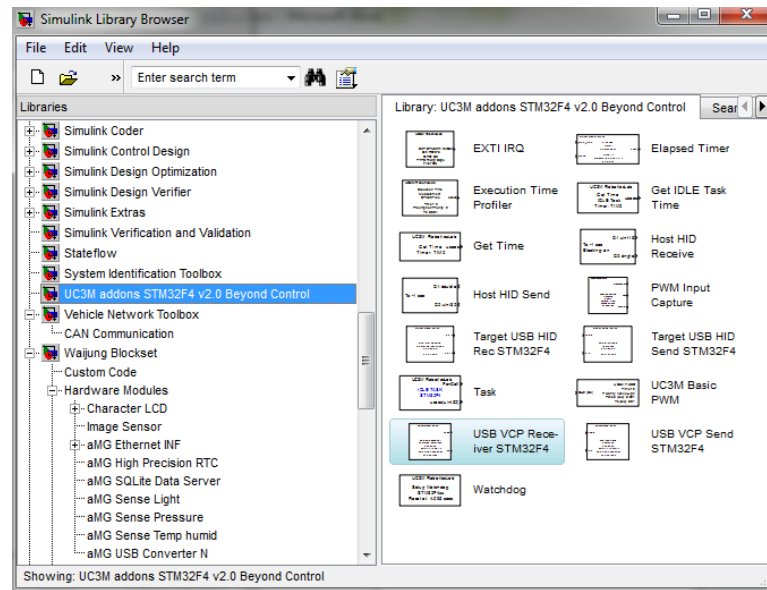


Figura 5. 9 USB VCP Receiver STM32F4

Como vemos en la Figura 5. 10 se definen las tres señales de tipo *single*, el tamaño, el tiempo de muestreo y la cabecera y terminación de los mensajes que va a recibir el microcontrolador, algo muy importante para que la comunicación entre el ordenador y el microcontrolador funcione.

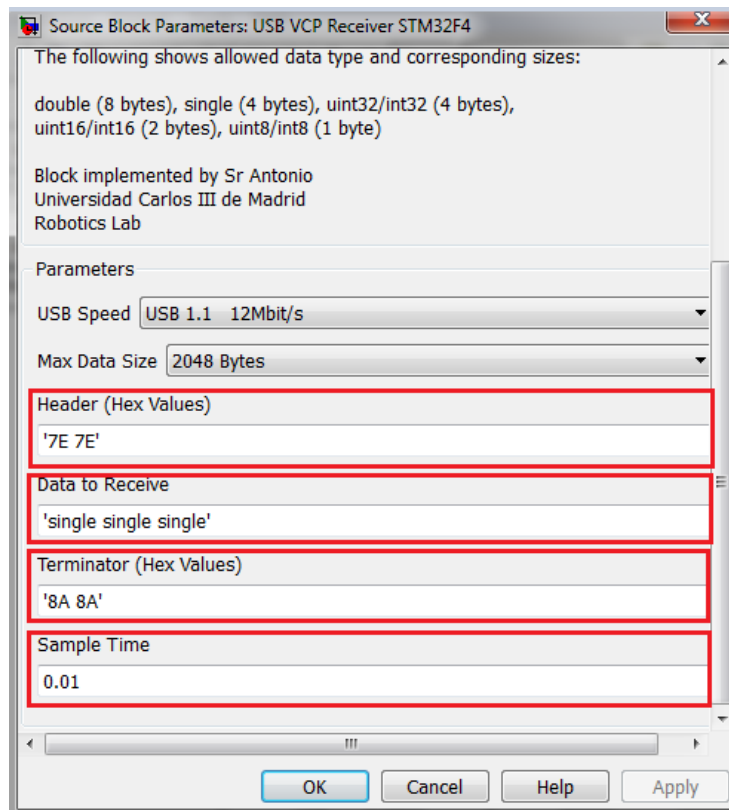


Figura 5. 10 Parámetros USB VCP Receiver STM32F4

Paso 3.

Ahora se procede a configurar cada una de las tres señales de salida del anterior bloque. El siguiente bloque que se utilizará está dentro de la biblioteca simulink, siguiendo la siguiente secuencia: **SIMULINK >> DISCRETE >> MEMORY**. Este bloque se utiliza para inicializar los valores de las señales de control de los servomotores.

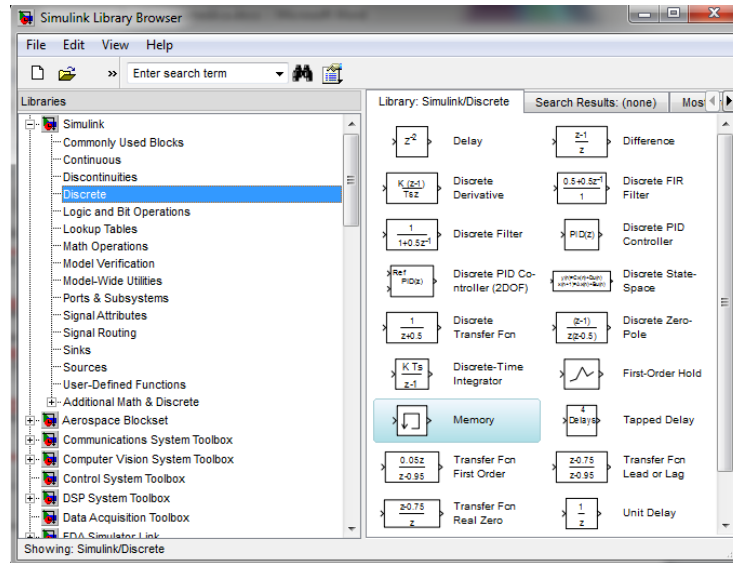


Figura 5. 11 Bloque Memory

En este bloque es importante definir la posición inicial de los servo-motores, las dos primeras señales de salida del bloque anterior (USB VCP Receiver STM32F4), empezaran en una posición angular de 180° y la tercera señal cuya posición inicial será 0° , por el posicionamiento de los dedos enfrentados y el giro en sentido contrario de los servo-motores. Otro aspecto a configurar es marcar la casilla de heredar tiempo de muestra (ver Figura 5. 12).

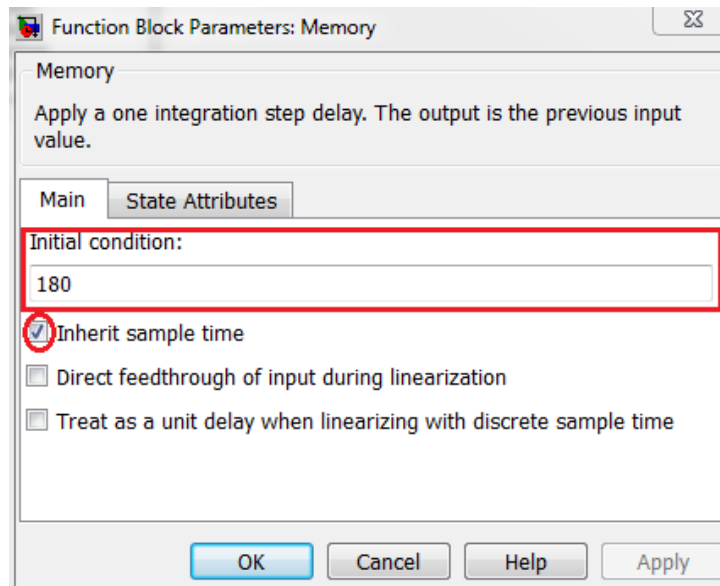


Figura 5. 12 *Parámetros del bloque Memory*

Una vez configurado y llevado el bloque al modelado de control, se conecta cada señal de salida del bloque USB VCP Receiver STM32F con cada bloque Memory parametrizado.

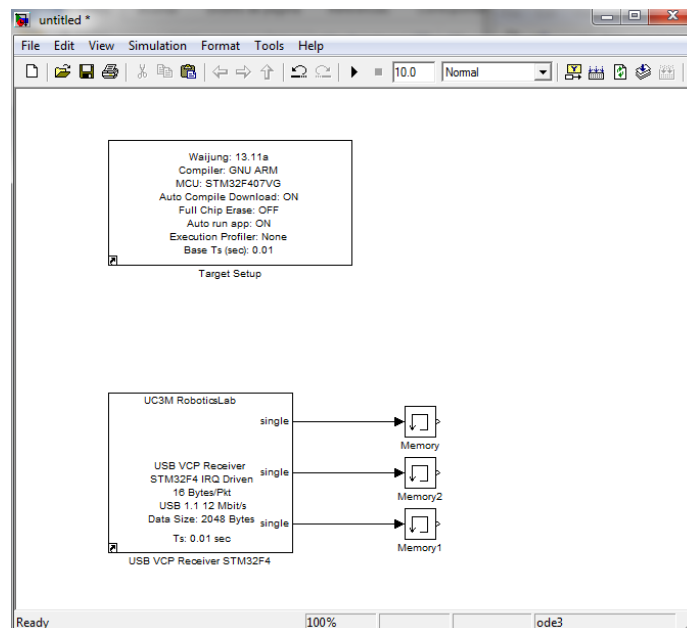


Figura 5. 13 *Modelo Control_PWM*

Paso 4

Una vez llegado a este punto, se necesita saber los rangos del valor del ancho de pulso de la señal de control PWM, a los cuales el servo-motor gira de 0° a 180° . Se introduce una posición angular y a continuación una función mapea los valores angulares que recibe el controlador en valores de ancho de pulso para la señal de control PWM de los servomotores. De esta manera se logra un control más intuitivo del giro de los servo-motores. Para introducir una ecuación se debe seleccionar la biblioteca USER-DEFINED FUNCTIONS y seleccionar el bloque Fcn.

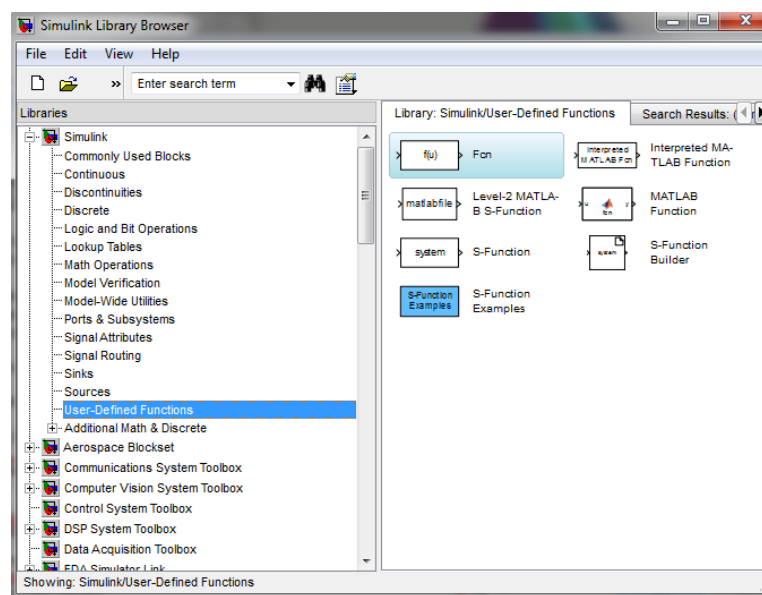


Figura 5. 14 Bloque Fcn

Los valores que se introducen en la función (ver Figura 5. 15) se han calculado de forma experimental. En esta ecuación el parámetro u es la señal de entrada, es decir, la posición angular a la que se quiere girar el motor. De forma experimental se ha obtenido un valor de ancho de pulso de 2.3 para la posición de 0° . De forma similar, se ha calculado un valor de ancho de pulso de 11.3 correspondiente a la posición angular de 180° . Con estos valores se ha obtenido la expresión que transforma las posiciones angulares en los valores de ancho de pulso correspondientes.

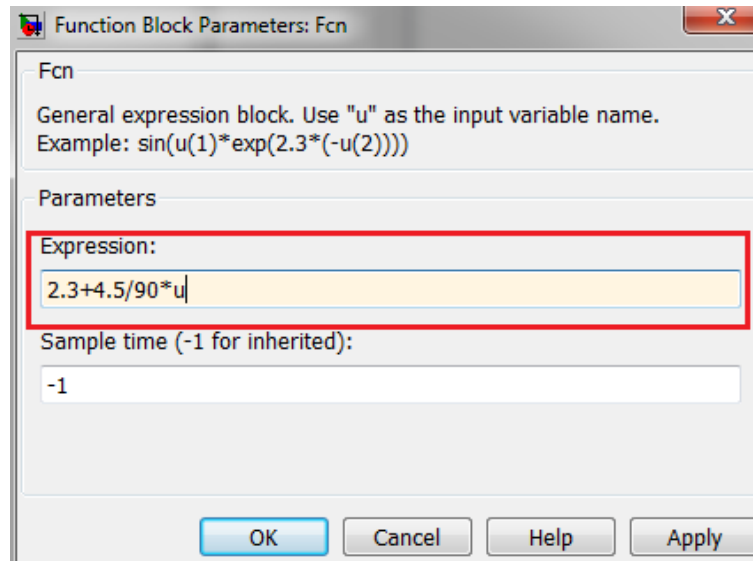


Figura 5. 15 Parámetros de la función

Paso 5

El siguiente bloque que se utilizará es *Saturation*, que se encuentra dentro de la biblioteca Simulink, siguiendo la secuencia: **SIMULINK >> DISCONTINUITIES >> SATURATION.**

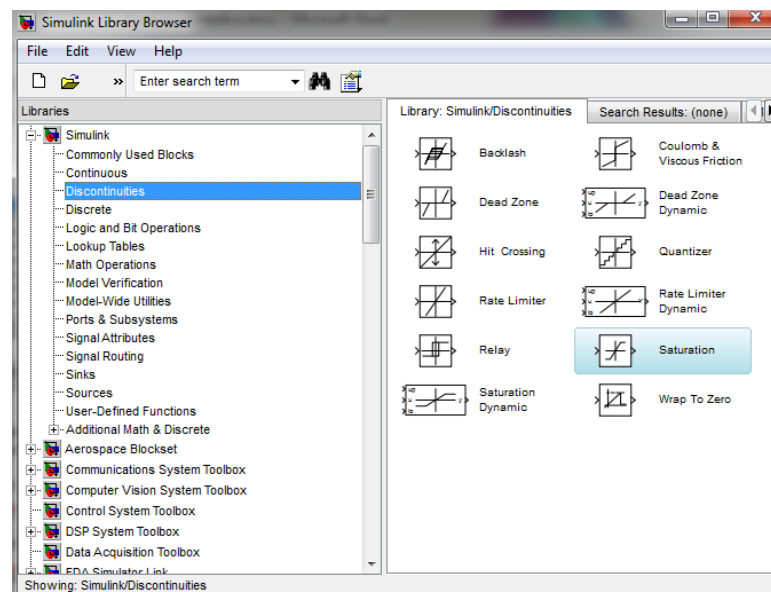


Figura 5. 16 Bloque Saturation

Este bloque limita la señal de entrada a los valores de saturación superior e inferior para evitar que la señal que entra en el bloque PWM supere los valores máximo y mínimo de ancho de pulso que aceptan los servomotores y de esta forma evitar un mal funcionamiento de los mismos. Estos valores se introducen en los parámetros del bloque, (ver Figura 5. 17)

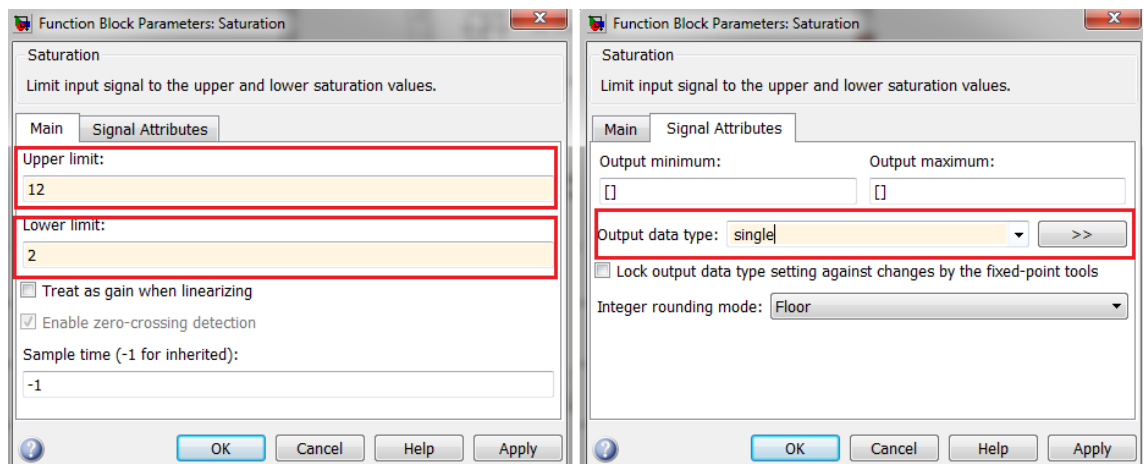


Figura 5. 17 Parámetros del bloque de Saturation.

Paso 6

El último bloque que se va a utilizar es el encargado de generar las tres señales PWM de control, con los valores de ancho de pulso que recibe en sus entradas. El bloque que se utiliza es UC3M Basic PWM.

UC3M ADDONS STM32F4 >> UC3M Basic PWM

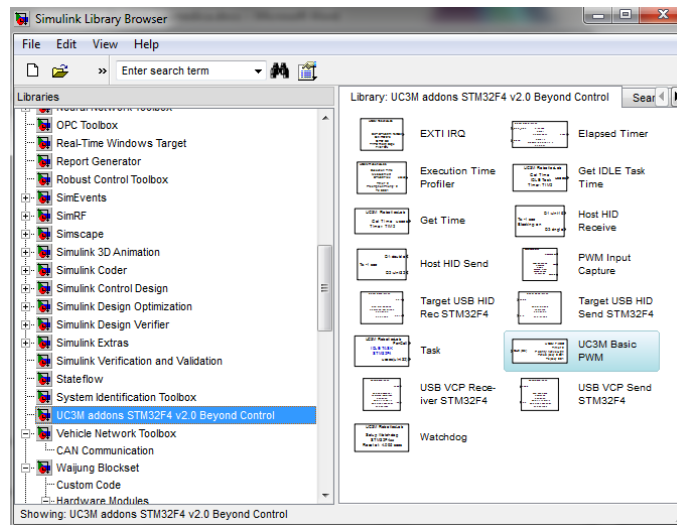


Figura 5. 18 UC3M Basic PWM.

Los parámetros que se deben de definir en este bloque son el periodo de las señales PWM que va a generar el bloque, así como el tiempo de muestreo del bloque y los puertos por donde se generan las señales.

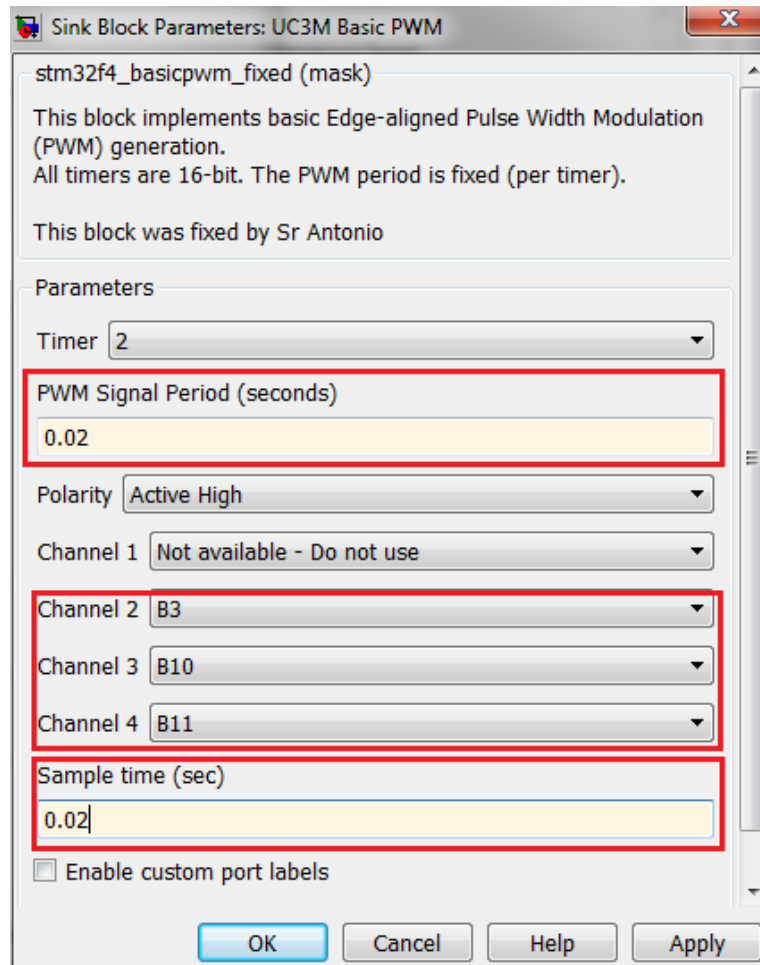


Figura 5. 19 Parámetros del bloque UC3M Basic PWM

El modelo del sistema de control implementado, genera tres señales PWM a partir de los valores enviados al microcontrolador desde un PC, conectando los distintos bloques que se han detallado en este apartado, puede verse en la Figura 5. 20.

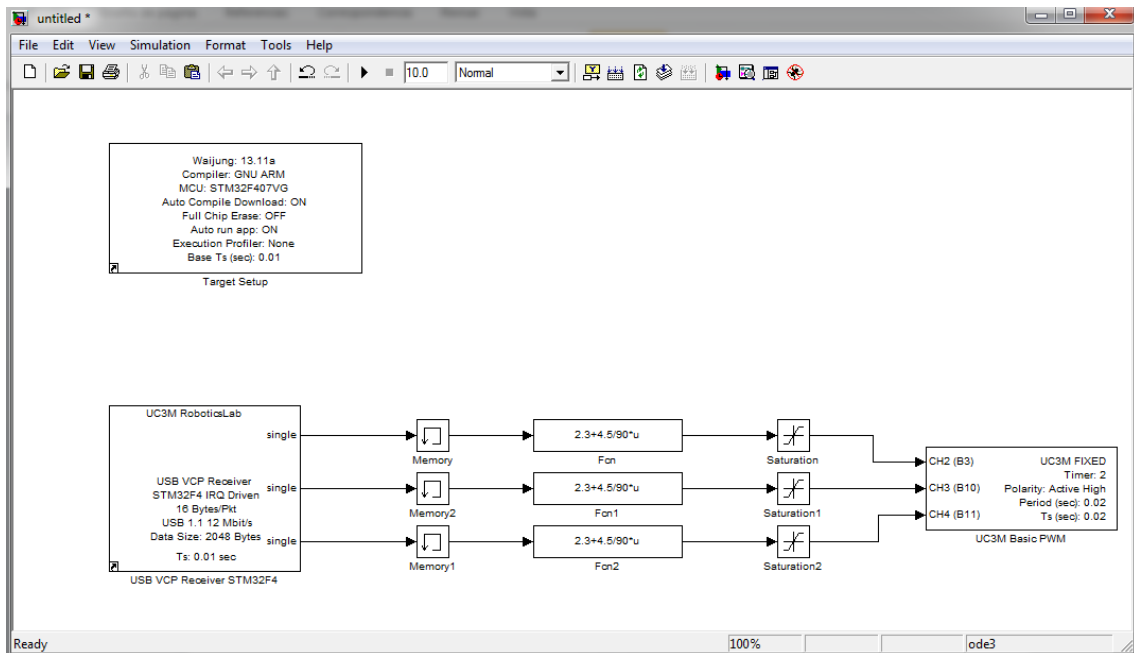


Figura 5. 20 Modelado de señales PWM

5.1.2. Implementación del Modelo Host

El modelo de control que se va a diseñar en este apartado es el encargado del control desde el ordenador. Como en el apartado anterior, el primer paso es abrir una nueva ventana de modelado.

El primer bloque se debe seleccionar es el *Host Serial Setup*, este bloque se utiliza para seleccionar el puerto COM del ordenador por donde se transmiten los datos al microcontrolador y la velocidad de comunicación. Se debe tener en cuenta que la velocidad máxima que permite el microcontrolador son 12 Mbps [26]. El bloque se encuentra en:

WAIJUNG BLOCKSET >> HARDWARE MODULES >> aMG USB CONVERTER N >> HOST SERIAL SETUP.

Para consultar el puerto COM al que se ha conectado al microcontrolador solo basta con entrar en: *inicio Windows >> dispositivos e impresoras >> botón derecho >> propiedades*

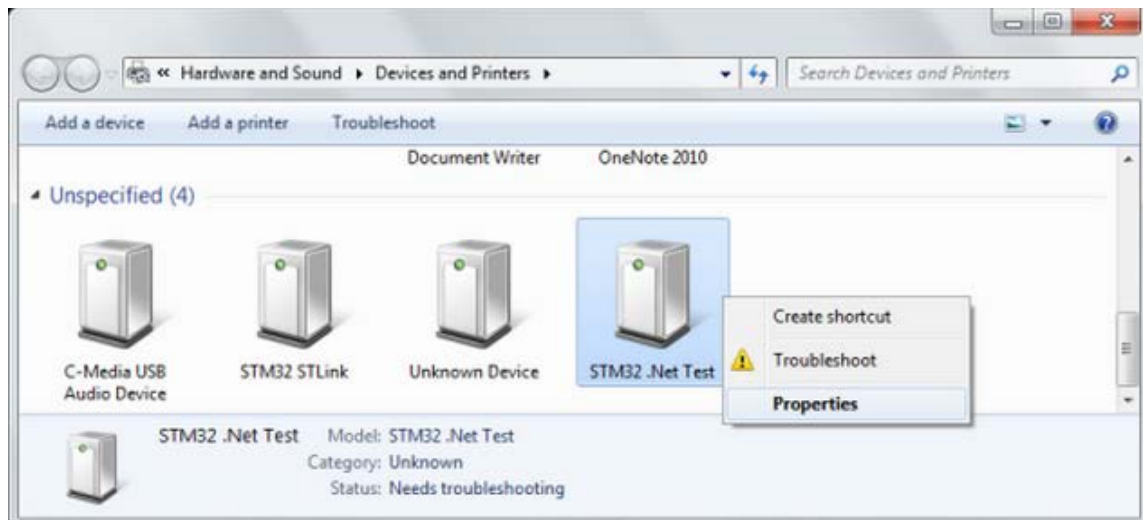


Figura 5. 21 Puerto COM del microcontrolador

El bloque queda parametrizado como se muestra en la Figura 5. 22

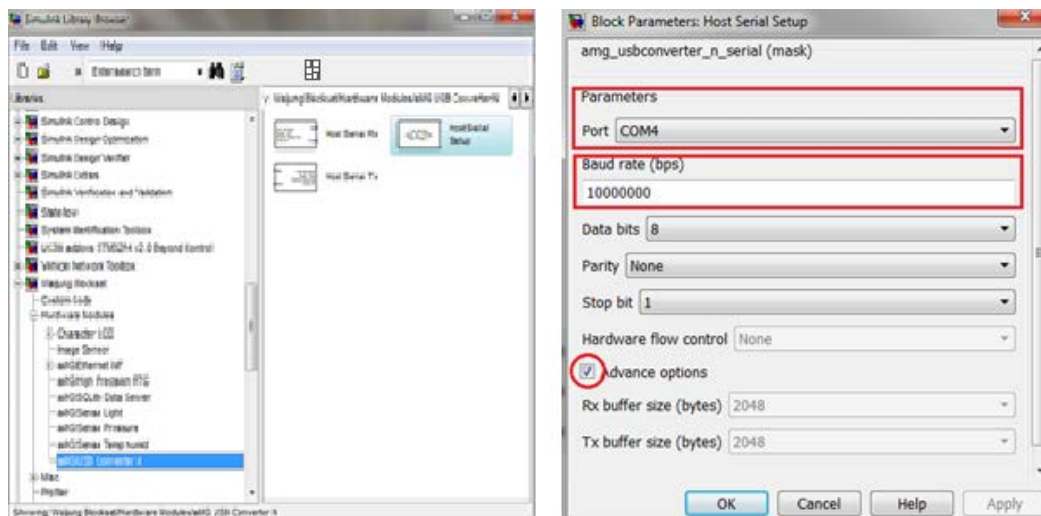


Figura 5. 22 Bloque Host Serial Setup

Para controlar el ángulo de giro de los servo-motores, se ha optado por utilizar un *Slider Gain*. Este bloque es un deslizador de ganancia, que multiplica por un escalar, cuyo valor se puede variar con un control deslizante. En el modelo diseñado, el bloque de entrada tendrá un valor de 1 que se multiplicará por una ganancia variable entre 0 y 180 (como se puede ver en la Figura 5. 23), que será la señal de control angular de los servomotores de la mano.

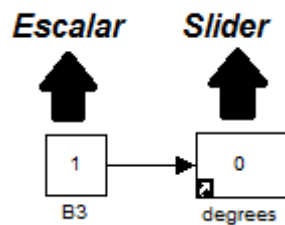


Figura 5. 23 Bloque Slider Gain

Como se muestra en la Figura 5. 24 se debe especificar el límite inferior y superior del rango de control deslizante, que en esta práctica corresponde 0 y 180 respectivamente [28].

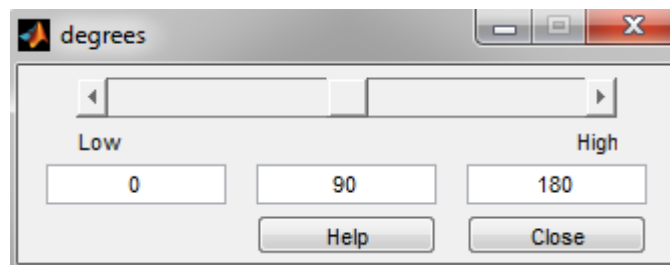


Figura 5. 24 Degrees

Por último se necesita transmitir los datos a través del puerto COM host del ordenador al microcontrolador. Para dicho propósito se utiliza el bloque *Host*

serial Tx, que se encuentra en la biblioteca *Waijung Blockset*, siguiendo la siguiente secuencia: **WAIJUNG BLOCKSET >> HARDWARE MODULES >> aMG USB CONVERTER N >> HOST SERIAL Tx**.

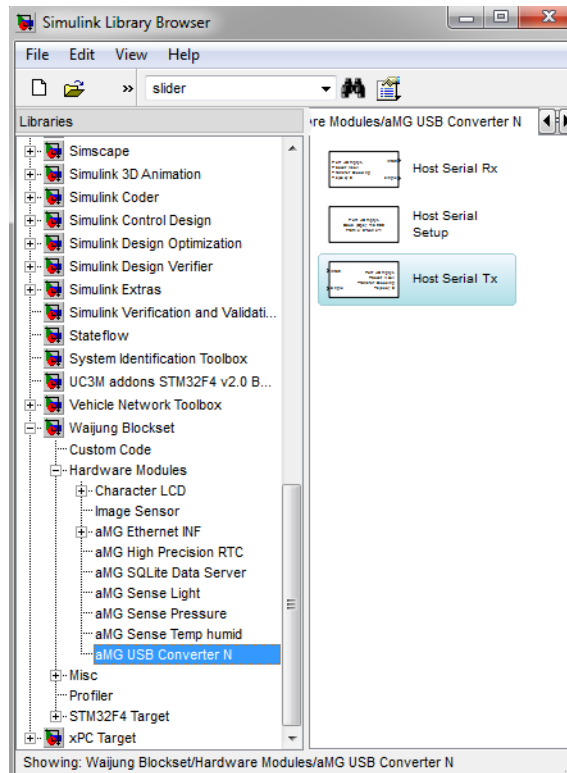


Figura 5. 25 Bloque Host Serial Tx

Las principales configuraciones que se debe realizar en este bloque son los que se muestran en la Figura 5. 26. Primero seleccionar el puerto COM, el puerto debe coincidir con el puerto COM especificado en el bloque de configuración de la comunicación USB, (ver Figura 5. 22), por el cual se va a transmitir los datos, el código por el cual se va a transmitir la información (binary, ascii). En este modelo de control se utiliza una transmisión mediante código binario, por lo cual también se debe definir cuál es la cabecera y la terminación de la información que se envía. Dicha cabecera y terminación tienen que coincidir con la configuración del bloque USB VCP Receiver STM32F4, que se ha parametrizado en el modelo de programación del microcontrolador (ver Figura 5. 10). Para terminar se define el número y el tipo de dato que se transmite, que

una vez más tiene que coincidir con el número y tipo de datos que recibe el bloque USB VCP Receiver STM32F4 del microcontrolador.

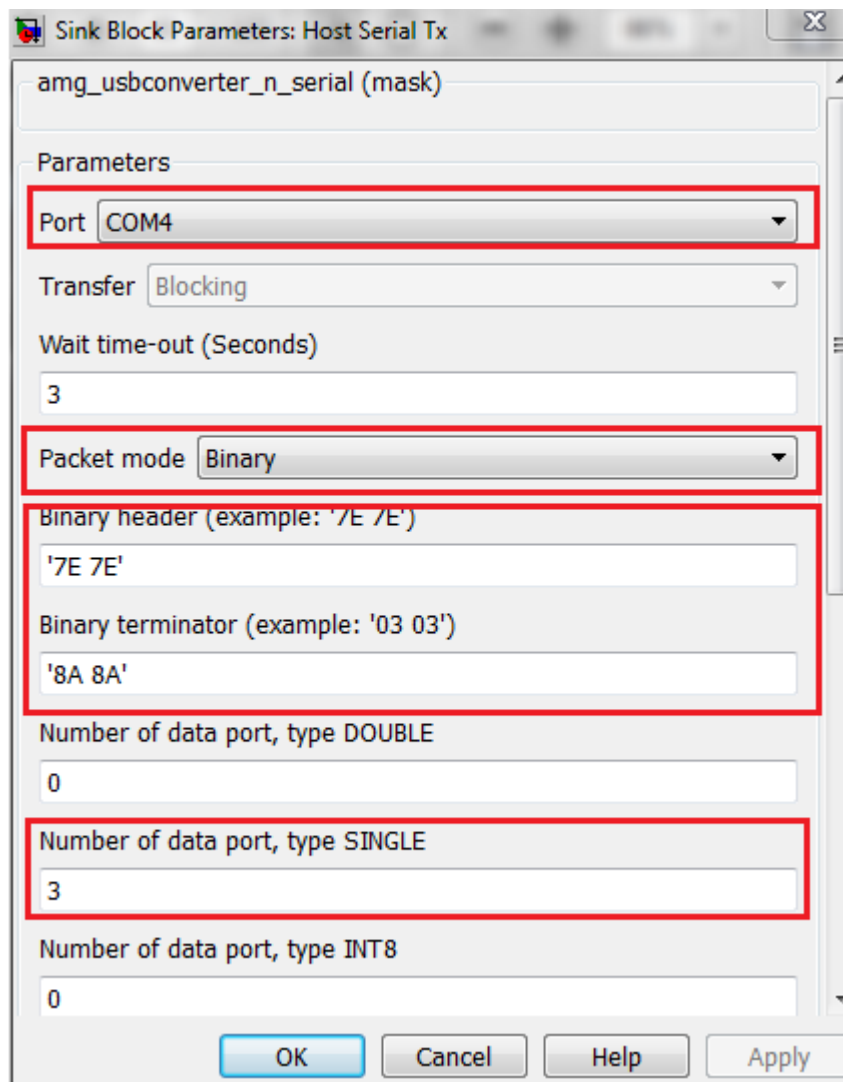


Figura 5. 26 Configuración Host Serial Tx

El tiempo de ejecución de este modelo de host debe ser infinito (inf), para que se ejecute durante un tiempo indefinido, hasta que la ejecución se detenga de forma manual. Conectando todos los bloques respectivamente se obtiene el modelo de control *Host* que enviará los datos al microprocesador el cual convierte esos datos en señales PWM idóneas para el control de los servo-motores.

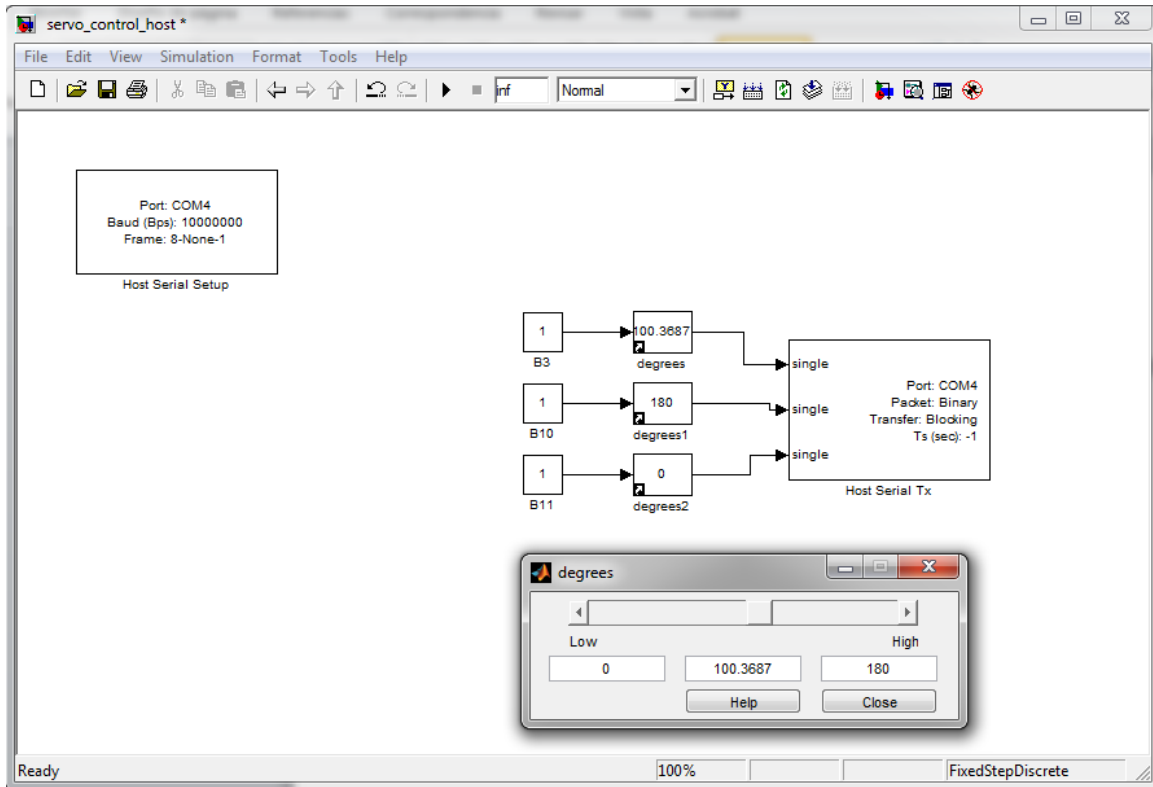


Figura 5. 27 Modelo Host de control



CAPÍTULO 6

6. Conclusiones finales y trabajos futuros

Durante los cinco capítulos anteriores se han presentado los aspectos más relevantes de este proyecto. A continuación se detallarán las conclusiones a las que se han llegado en las distintas etapas del mismo.

Como se ha mencionado se han superado distintas fases en el desarrollo del proyecto. En primer lugar se realizó el *diseño y la construcción física de la mano*.

Los factores evaluados y las conclusiones obtenidas en esta etapa fueron las siguientes:

➤ **Funcionalidad del prototipo completo**

Se han evaluado diferentes aspectos, por ejemplo si el prototipo es ensamblable y si los ajustes mecánicos del mismo permiten la movilidad deseada. Se determinó después de múltiples pruebas que así es.

➤ Dinámica de los actuadores

En esta fase se dedujo que los parámetros que se han tomado en cuenta en el control de las articulaciones eran los apropiados ya que se ha conseguido que los dedos del manipulador robótico se muevan con facilidad, y el rozamiento sea mínimo (lo que se asegura una gran durabilidad del prototipo).

➤ Funcionamiento de los tendones

Se ha comprobado que el sistema de transmisión que se ha optado para la movilidad de las falanges en diferentes condiciones de trabajo es apto para la aplicación.

➤ Funcionamiento de la etapa eléctrica de potencia

Las pruebas eléctricas realizadas han demostrado que la electrónica de potencia usada es lo suficientemente capaz de soportar la cantidad de corriente que demandan los actuadores, y por tanto es válida para el proyecto.

Como se puede apreciar en esta etapa de construcción se ha trabajado tanto en la construcción del sistema completo de acondicionamiento de señales eléctricas como en la fabricación de las partes complementarias del prototipo.

Tras analizar los factores que se acaban de detallar, puede concluirse que en esta etapa de fabricación y testeo de la mano robótica se han cumplido con los objetivos marcados al inicio del proyecto.

Tras haber analizado el cumplimiento de los objetivos en la fase de diseño y fabricación del prototipo, se va a analizar la siguiente fase del proyecto, de acuerdo con los objetivos marcados, realizar *estrategias de control*. Se ha implementado un modelo de control de forma sencilla gracias a un entorno de programación gráfico, y de cómo este modelo básico puede servir como punto de partida para la implementación de modelos más complejos.

Por último, cabe señalar que se han alcanzado todos los objetivos planteados en este proyecto, y que como consecuencia, actualmente se cuenta con un prototipo de una mano de tres dedos con tres GDL cada uno y con la capacidad de desarrollar movimientos de flexión y extensión para su uso docente.

6.1. Sugerencias para trabajos futuros

Es importante destacar que los resultados que se han obtenido con esta primera versión de la *mano para uso docente* son susceptibles a mejoras. No obstante, cumple con las especificaciones requeridas y el conocimiento que se ha generado ha sido muy importante para la realización de futuras versiones de la mano. Es importante considerar las siguientes recomendaciones:

En el área de instrumentación:

Incorporar distintos tipos de sensores, para obtener una mayor información del movimiento de los dedos, por ejemplo: incorporar sensores de tacto, para obtener información de la fuerza que aplica el manipulador robótico al sustentar e interactuar con diferentes objetos sin romperlos o deformarlos, incorporar sensores de fuerza para monitorear la tensión de los tendones, incorporar sensores de posición en cada falange del dedo para controlar la posición exacta y que los movimientos correspondan a la posición que se desea.

En el área mecánica:

Disminuir el volumen de la mano completa buscando componentes de menor dimensión. Integrar los diferentes componentes electrónicos en el diseño del manipulador robótico e incluir el diseño de la muñeca, para obtener una mayor destreza en los movimientos de la mano.

En el área electrónica:

Disminuir el tamaño de las tarjetas electrónicas, hacer un diseño electrónico más compacto con el fin de poder incorporarla dentro del prototipo diseñado para aumentar el antropomorfismo evitando así tener cables externos al diseño final.



Bibliografía

- [1] Instituto de investigación en ingeniería de Aragón, «Ingeniería Biomédica,» [En línea]. Available: <http://www.monografias.com/trabajos-pdf5/bioingenieria-disciplinas/bioingenieria-disciplinas.shtml>. [Último acceso: 2014].
- [2] Universidad Carlos III de Madrid, «Descripción de la asignatura Aplicaciones de la automática en biomedicina,» [En línea]. Available: http://www3.uc3m.es/reina/Fichas/Idioma_1/223.14061.html. [Último acceso: Julio 2014].
- [3] Mathworks, «Aplicaciones de Simulink,» [En línea]. Available: www.mathworks.es/products/simulink/. [Último acceso: Julio 2014].
- [4] Wikipedia, «Impresora 3D,» 6 Agosto 2014. [En línea]. Available: http://es.wikipedia.org/wiki/Impresi%C3%B3n_3D. [Último acceso: 2014].
- [5] A. V. Peciña, «Diseño y aplicación de un actuador SMA en el control de manos robóticas,» 2010-2011. [En línea]. Available: <http://e-archivo.uc3m.es/handle/10016/10484>. [Último acceso: 25 Julio 2014].
- [6] Mindtrans, «Manos Roboticas,» [En línea]. Available: <http://mindtrans.narod.ru/hands/hands.htm>. [Último acceso: Julio 2014].
- [7] RO-BOTICA Global S.L, «Robótica Educativa y Personal en Europa,» [En línea]. Available: <http://ro-botica.com/Educacion.aspx>. [Último acceso: Julio 2014].
- [8] Arduino S.A, «Arduino,» [En línea]. Available: <http://arduino.cc/en/pmwiki.php?n=>. [Último acceso: Julio 2014].
- [9] Robotis, «Bioloid Live,» [En línea]. Available: http://www.robotis.com/x/bioloid_en. [Último acceso: Julio 2014].
- [10] Robotis, «DARWIN-OP,» [En línea]. Available: http://www.robotis.com/x/darwin_ko. [Último acceso: Julio 2014].
- [11] ABB Corporations, «Robots Industriales,» [En línea]. Available: <http://www.abb.es/product/seitp327/b0168cc2e1ea9fd5c125717900223403.aspx>.



- [Último acceso: Julio 2014].
- [12] Northwest Regional Educational Laboratory, «Eduteka,» 11 Marzo 2006. [En línea]. Available: <http://www.eduteka.org/AprendizajePorProyectos.php>. [Último acceso: Julio 2014].
- [13] R. Soames, D. Field y N. Palastanga, «ANATOMÍA Y MOVIMIENTO HUMANO. ESTRUCTURA Y FUNCIONAMIENTO,» vol. Musculos de la mano, Barcelona, Paidotribo, 2000.
- [14] Cenidet, H. Cimadevilla Lajud y J. G. Herrera Perez, «Diseño de un sistema articulado emulando el movimiento de una mano,» Junio 2006. [En línea]. Available: <https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCIQFjAA&url=http%3A%2F%2Fwww.cenidet.edu.mx%2Fsubaca%2Fweb-mktro%2Fsubmenus%2Finvestigacion%2Ftesis%2F23-24%2520Hernando%2520Cimadevilla%2520Lajud%2520-%2520Jesus%2520Gust>. [Último acceso: Julio 2014].
- [15] E. Persson y M. De Lourdes, Anatomía Humana, Segunda ed., Vols. %1 de %2Unidad 1, Fascículo 1, México D.F.: Buena Onda S.A., 2002.
- [16] Solidworks, «Software Solidworks,» [En línea]. Available: www.solidworks.es/. [Último acceso: Julio 2014].
- [17] Textos Científicos .com, «Propiedades del Nylon,» [En línea]. Available: <http://www.textoscientificos.com/polimeros/nylon/propiedades>. [Último acceso: 2014].
- [18] Todo3D, «Modelado por Deposición Fundida,» [En línea]. Available: <http://todo3d.cl/que-es-fdm/>. [Último acceso: 2014].
- [19] Wikipedia, «Características del plástico ABS,» [En línea]. Available: http://es.wikipedia.org/wiki/Acrilonitrilo_butadieno_estireno. [Último acceso: 2014].
- [20] Futaba, «Características Servomotores Futaba,» [En línea]. Available: <http://www.futaba-rc.com/servos/analog.html>. [Último acceso: 2014].
- [21] Cadence, «OrCAD Official Website,» [En línea]. Available: <http://www.orcad.com/>. [Último acceso: 2014].
- [22] Wikispace, «Simulink,» [En línea]. Available: simulink-aplicacion-scd.wikispaces.com. [Último acceso: Julio 2014].



- [23] STMicroelectronics group of companies, «STM32F4,» [En línea]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf. [Último acceso: Julio 2014].
- [24] Raspberry PI, «Características Raspberry PI,» [En línea]. Available: <http://www.raspberrypi.org/>. [Último acceso: 2014].
- [25] MathWorks, «Diseño Basado en modelos,» [En línea]. Available: <http://www.mathworks.es/model-based-design/>. [Último acceso: 2014].
- [26] Aimagin, «Waijung Blockset,» [En línea]. Available: <http://waijung.aimagin.com/>. [Último acceso: 2014].
- [27] Aimagin Forum, «Descarga de bibliotecas,» [En línea]. Available: <https://www.aimagin.com/forum/viewtopic.php?f=2&t=269>. [Último acceso: 2014].
- [28] MathWorks, «Slider Gain,» [En línea]. Available: <http://www.mathworks.es/es/help/simulink/slref/slidergain.html>. [Último acceso: 2014].



ANEXOS

Anexo A: Fases de Trabajo y Presupuesto

El tiempo que se ha invertido tanto en el diseño, como la implementación de este proyecto se desglosarán a continuación. Este resumen ayudará a cuantificar el coste aproximado de su realización.



A. Fases de trabajo

Para un mejor entendimiento del desarrollo del proyecto se ha dividido por fases toda la planificación del manipulador robótico.

Fase 1: Documentación inicial

- a) Planteamiento del problema y necesidades docentes: 60 horas
- b) Documentación acerca del entorno de trabajo: 60 horas
- c) Búsqueda de dispositivos y componentes: 20 horas
- d) Soluciones propuestas: 20 horas

Fase 2: Diseño

- a) Búsqueda de alternativas de diseño y restricciones: 20 horas
- b) Generar esquemáticos de piezas: 150 horas

Fase 3: Implementación del prototipo

- a) Fabricación: 170 horas
- b) Montaje: 50 horas

Fase 4: Ensayos realizados

- a) Pruebas y ensayos: 50
- b) Mejoras: 20

Fase 5: Escritura de memoria

- a) Redacción de la memoria: 150 horas
- b) Corrección y estructuración: 70 horas



Tabla A. 1 Desglose por fases

<i>FASES</i>	<i>HORAS</i>
<i>Documentación inicial</i>	<i>160</i>
<i>Diseño</i>	<i>170</i>
<i>Implementación del prototipo</i>	<i>220</i>
<i>Ensayos realizados</i>	<i>70</i>
<i>Escritura de memoria</i>	<i>220</i>
<i>TOTAL</i>	<i>840</i>

B. Presupuesto del Trabajo Fin de Grado

En este apartado se va a tratar de aproximar el coste económico del desarrollo de este trabajo de fin de grado. Estos costes se van a dividir en: costes de material, costes de fabricación, y coste de personal.

Costes de materiales

Los costes de materiales se han dividido en tres grandes grupos: un primer grupo en el que se cuantifica el gasto de plástico ABS que se ha utilizado para la fabricación de las piezas que componen el manipulador robótico; un segundo grupo de componentes electrónicos y actuadores que se han utilizado para el correcto funcionamiento del proyecto; y un último grupo de material informático que se ha utilizado.

En esta primera tabla se muestra el consumo de plástico ABS que se ha consumido para la fabricación de un prototipo.

Tabla A. 3 Consumo de Plástico ABS

<i>Pieza</i>	<i>Volumen Solidworks (cm³)</i>	<i>Relleno</i>	<i>Volumen total pieza (cm³)</i>	<i>Densidad plástico ABS g/cm³</i>	<i>Masa total en gramos</i>
<i>Distal 1</i>	1,28	0,3	0,39	1,05	0,40
<i>Distal 2</i>	1,21	0,3	0,36	1,05	0,38
<i>Distal 3</i>	3,24	0,3	0,97	1,05	1,02
<i>Medial 1</i>	2,18	0,3	0,66	1,05	0,68
<i>Medial 2</i>	2,00	0,3	0,60	1,05	0,63
<i>Medial 3</i>	3,6	0,3	1,09	1,05	1,14
<i>Proximal 1</i>	4,35	0,3	1,31	1,05	1,375
<i>Proximal 2</i>	4,15	0,3	1,25	1,05	1,30
<i>Proximal 3</i>	6,79	0,3	2,04	1,05	2,13
<i>Dedo entero</i>	28,86		8,66		9,09
<i>Palma de la mano</i>	42,74	0,30	12,82	1,05	13,46
<i>Base servomotores macho</i>	26,97	0,30	8,09	1,05	8,50



<i>Base servomotores hembra</i>	23,28	0,30	6,98	1,05	7,33
<i>Cubierta</i>	99,47	0,30	29,84	1,05	31,33

En esta segunda tabla se relaciona el consumo del material con su valor. El precio de un kilogramo de plástico ABS es de veinte euros, valor facilitado por el proveedor.

Tabla A. 4 Costes del Plástico ABS

<i>Inventario</i>	<i>Unidades</i>	<i>Pieza</i>	<i>Masa (gramos)</i>	<i>Coste (€/unidad) 1 kg de plástico ABS = 20 €</i>	<i>Coste real (€)</i>
1	3	<i>Dedo entero</i>	9,09	0,18	0,54
2	1	<i>Palma de la mano</i>	13,46	0,27	0,27
3	1	<i>Base servomotores macho</i>	8,50	0,17	0,17
4	1	<i>Base servomotores hembra</i>	7,33	0,15	0,15
5	1	<i>Cubierta</i>	31,33	0,62	0,62
TOTAL			87,89	1,39	1,76

Aparte del consumo de plástico para este proyecto hicieron falta componentes y dispositivos electrónicos para su funcionamiento final. En la Tabla A. 5 se muestra un listado del material que ha sido necesario para su montaje.

Tabla A. 5 Costes componentes y dispositivos Electrónicos

<i>Inventario</i>	<i>Unidades</i>	<i>Descripción</i>	<i>Coste unidad</i>	<i>Coste total</i>
1	1	Regulador LM7806	0,41 €	0,41 €
2	1	Condensador cerámico 0,1 μ F	0,05 €	0,05 €
3	1	Condensador cerámico 1 μ F	0,05 €	0,05 €
4	3	Servo-motores Futaba S3003	5,69 €	17,07 €
5	1	Microcontrolador STM32F4	13,76 €	13,76 €
TOTAL			19,96 €	31,34 €

Por último se deben cuantificar los costes de las licencias de los programas que se han utilizado en el transcurso del proyecto (Matlab, Orcad y Solidworks), pero al ser licencias para estudiantes facilitadas por la Universidad Carlos III de Madrid son totalmente gratuitas. Los costes totales de los materiales se muestran en la Tabla A. 6.

Tabla A. 6 Costes totales de materiales

<i>Concepto</i>	<i>Precio</i>
<i>Componentes y dispositivos Electrónicos</i>	31,34 €
<i>Consumo Plástico ABS</i>	1,76 €
TOTAL	33,10 €

Costes de fabricación

Por una parte se tienen los costes de fabricación de la placa PCB que se realizó en las instalaciones de la Oficina Técnica de la Universidad Carlos III de Madrid dedicadas a la fabricación de prototipos para proyectos docentes e investigación. El presupuesto ofertado por el departamento se muestra en la Tabla A. 8 donde únicamente aparece el coste total, ya que los costes de material, personal y desgaste de maquinaria no fueron facilitados por el técnico. Por otro lado se debe considerar el coste de la fabricación

realizada por la impresora 3D del laboratorio de Robótica y Automática de la Universidad Carlos III de Madrid de cada una de las partes del dedo que se ha diseñado así, como su base y demás componentes.

Tabla A. 7 Costes de fabricación en impresora 3D

<i>Concepto</i>	<i>Coste unitario</i>	<i>Vida útil</i>	<i>Uso</i>	<i>Coste</i>
<i>Impresora 3D</i>	<i>425,00€</i>	<i>60 meses</i>	<i>6 meses</i>	<i>42,50€</i>
			<i>TOTAL</i>	<i>42,50€</i>

Tabla A. 8 Costes totales de fabricación

<i>Concepto</i>	<i>Precio</i>
<i>Fabricación PCB</i>	<i>50,00 €</i>
<i>Fabricación Impresora 3D</i>	<i>42,50€</i>
<i>TOTAL</i>	<i>92,50 €</i>

Costes del personal

El autor de este trabajo de fin de grado se considera en calidad de ingeniero. Por tanto, considerando el salario actual de un ingeniero y las horas empleadas a la realización del proyecto, en la Tabla A. 9 se hace una valoración aproximada de la cuantía del coste del personal, en la cual se incluye retenciones e impuestos.

Tabla A. 9 Costes totales de personal

<i>Personal</i>	<i>Horas</i>	<i>Precio/hora</i>	<i>Precio</i>
<i>Ingeniero</i>	<i>840</i>	<i>10</i>	<i>8400,00</i>
<i>TOTAL</i>	<i>840</i>		<i>8400,00</i>



Costes totales

CONCEPTO	PRECIO
<i>Coste de materiales</i>	33,10
<i>Coste de fabricación</i>	92,50
<i>Coste de personal</i>	8.400,00
<i>Subtotal</i>	8.525,60
<i>IVA (21%)</i>	26,37
TOTAL	8.551,97

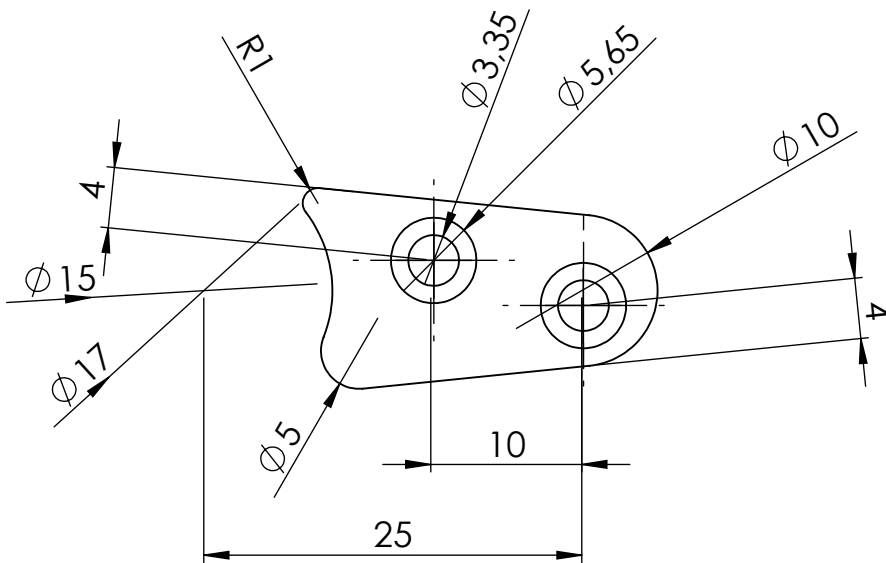
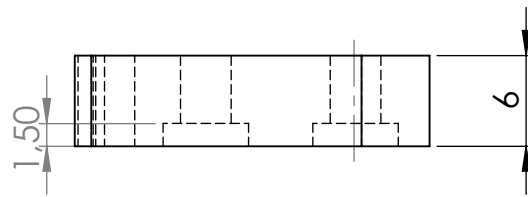
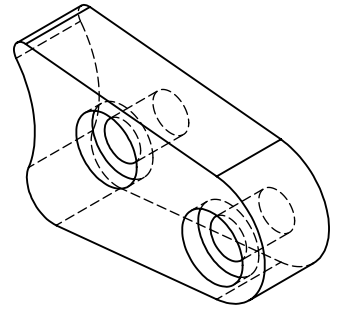
El coste total del proyecto es de OCHO MIL QUINIENTOS CINCUENTA Y UN EUROS CON NOVENTA Y SIETE CENTIMOS.


En _____, a ____ de _____ del 20__.

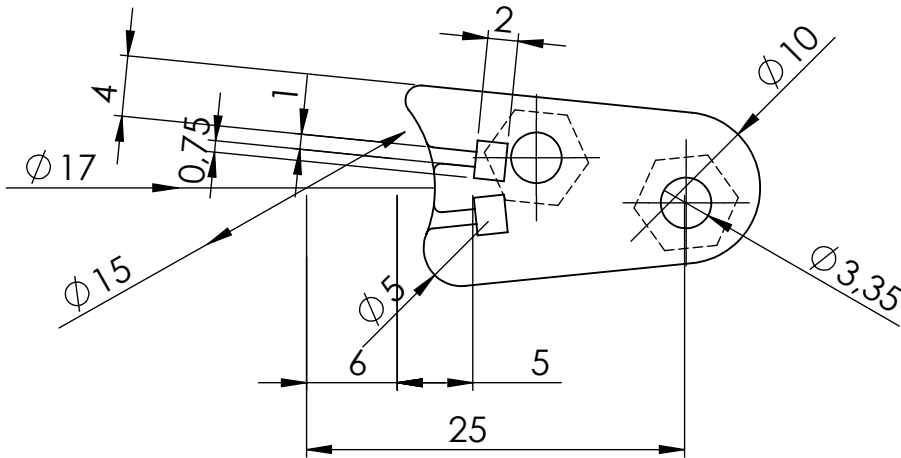
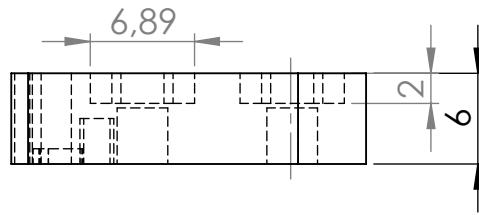
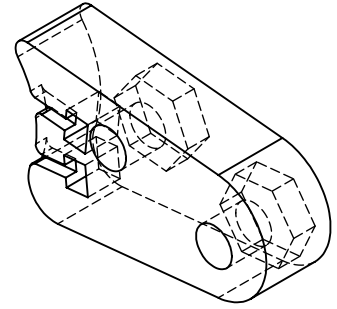
El Ingeniero.



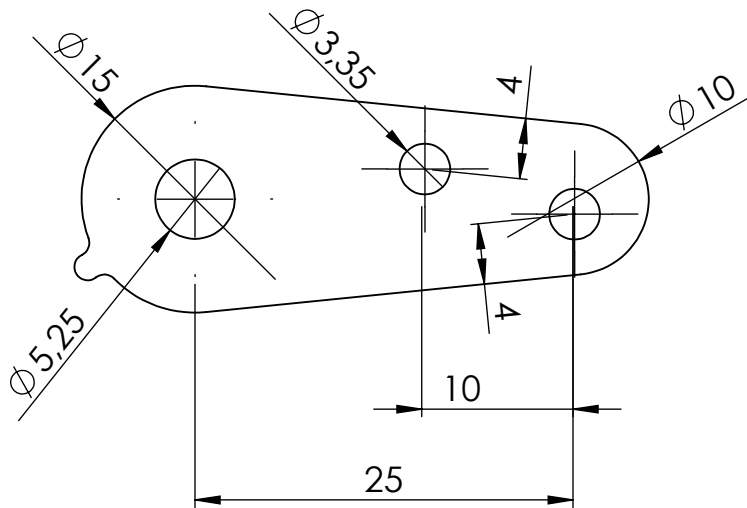
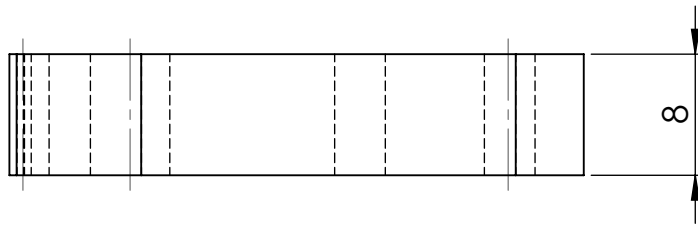
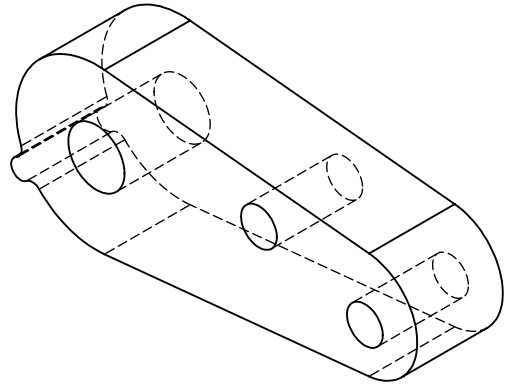
Anexo B: Planos




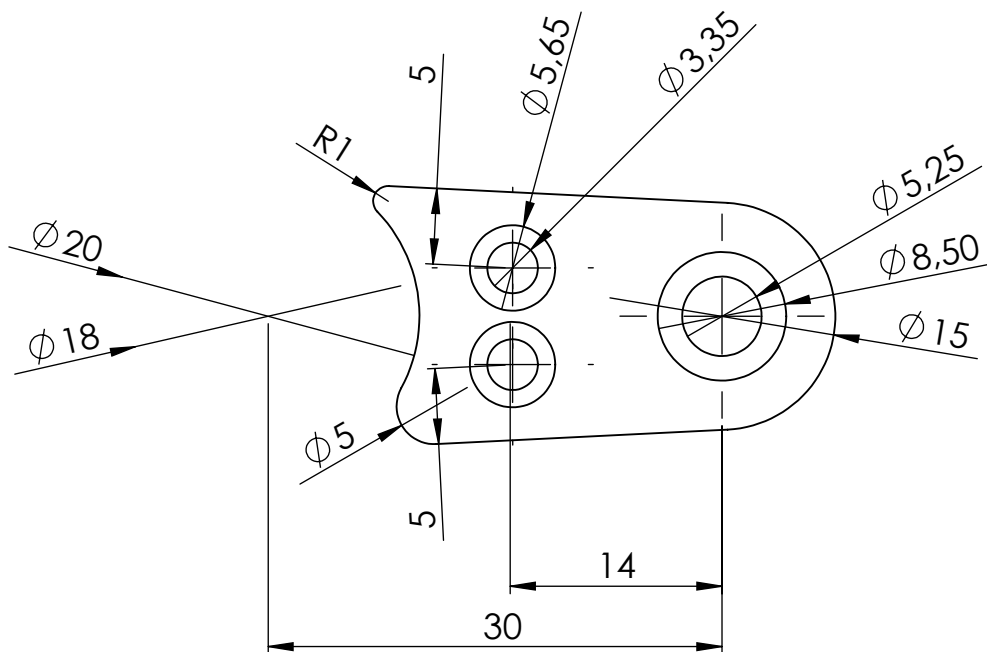
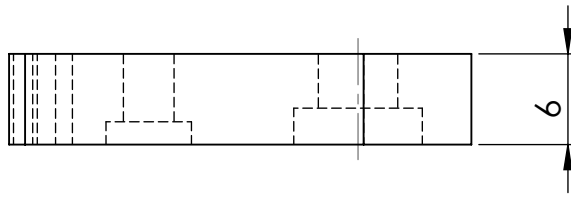
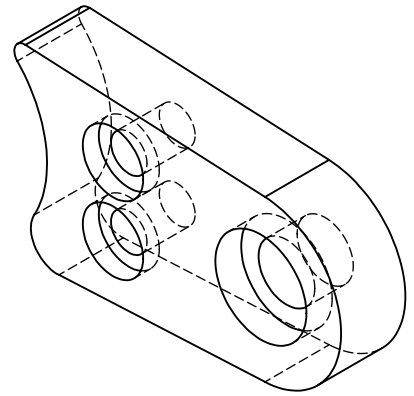
	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	<i>DISTAL 1</i>		Plano N. 1
2:1			Sustituye a:
			Sustituido por:



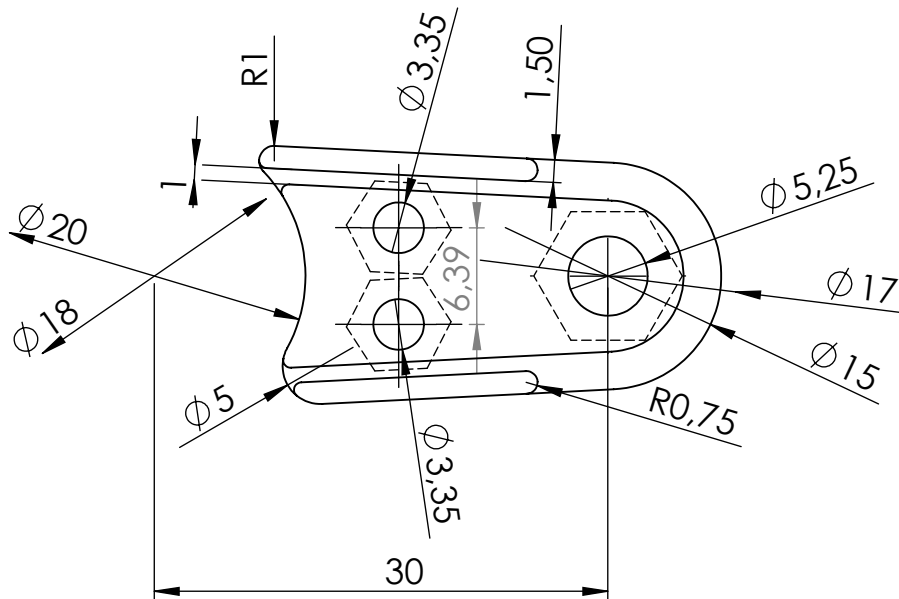
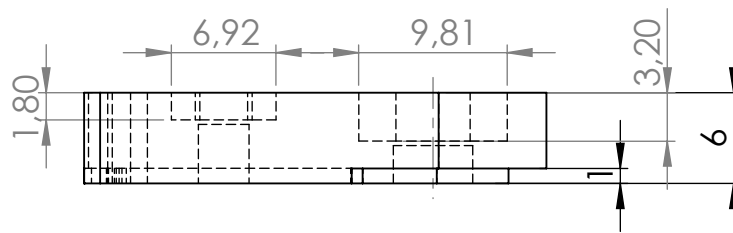
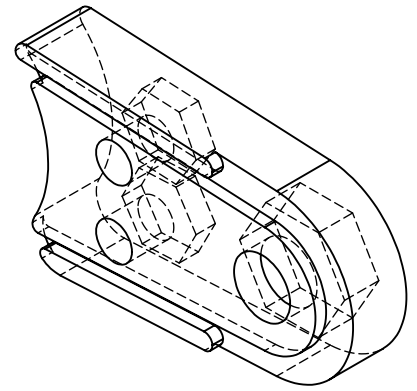
	Fecha	Nombre	 Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	<i>DISTAL 2</i>		Plano N. 2
2:1			Sustituye a:
			Sustituido por:



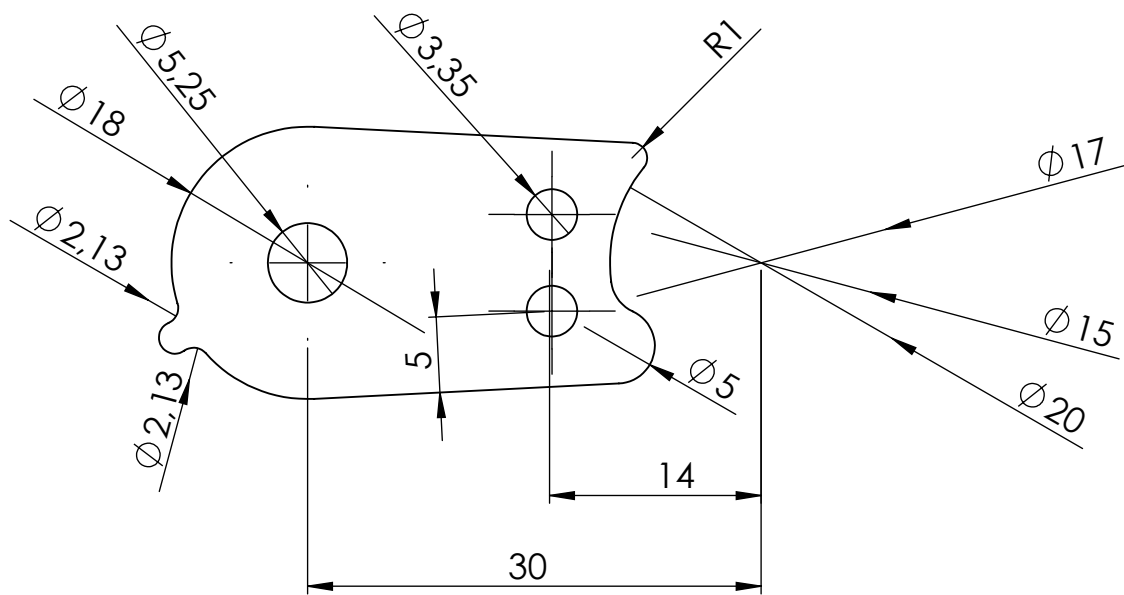
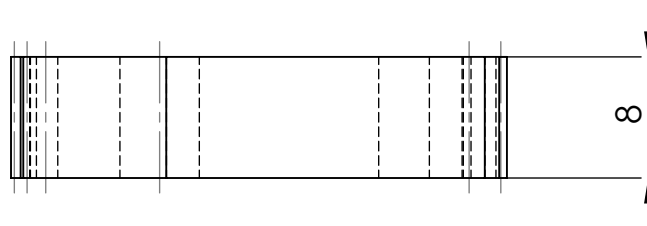
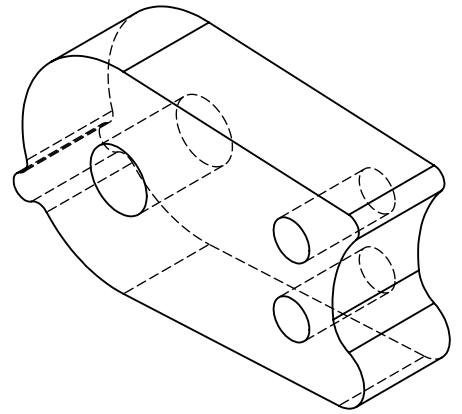
	<i>Fecha</i>	<i>Nombre</i>	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
<i>Dibujado:</i>	01/09/14	Jonathan	
<i>Comprobado:</i>			
<i>Id.s.norm:</i>			
<i>Escala</i>	<i>DISTAL 3</i>		Plano N. 3
2:1			Sustituye a:
			Sustituido por:




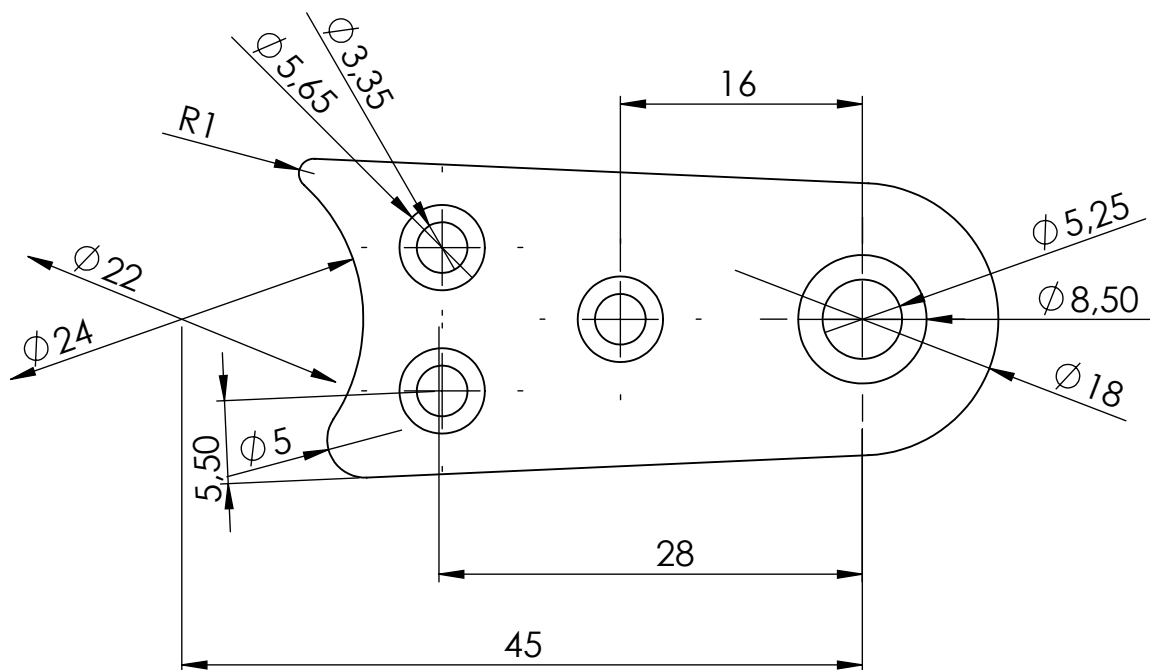
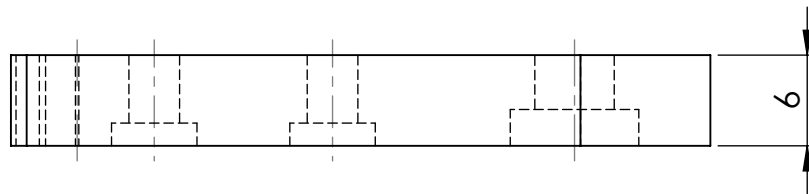
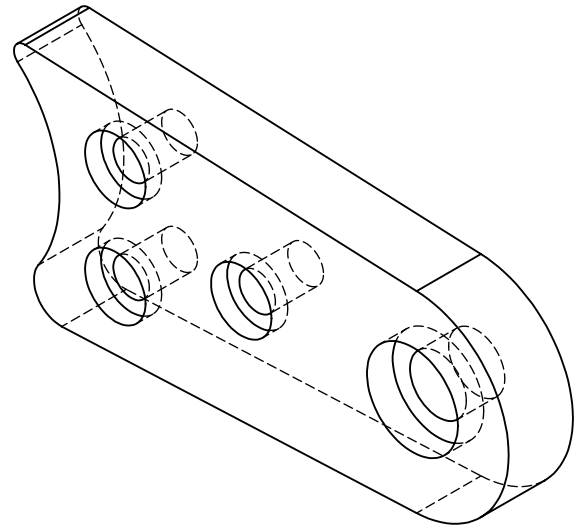
	Fecha	Nombre	 Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	MEDIAL 1		Plano N. 4
2:1			Sustituye a:
			Sustituido por:



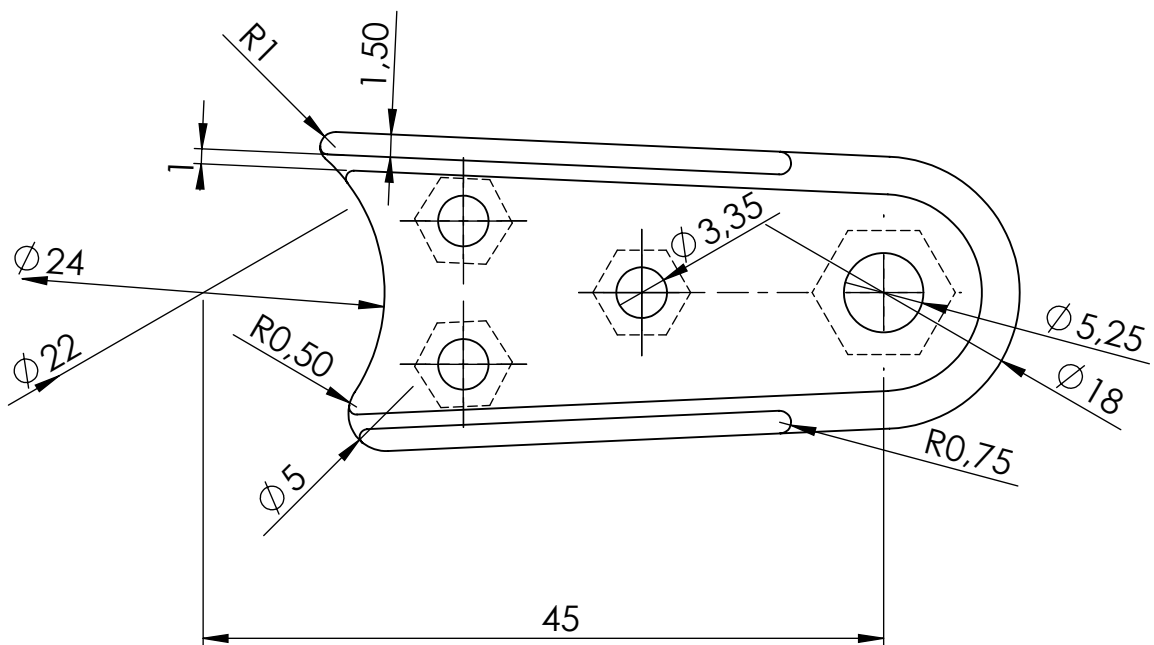
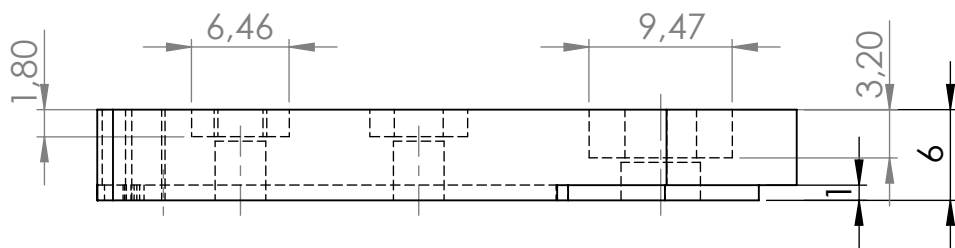
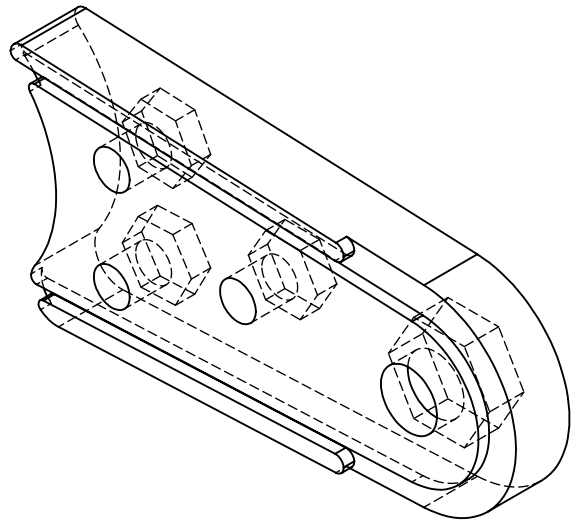
	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	MEDIAL 2		Plano N. 5
2:1			Sustituye a:
			Sustituido por:




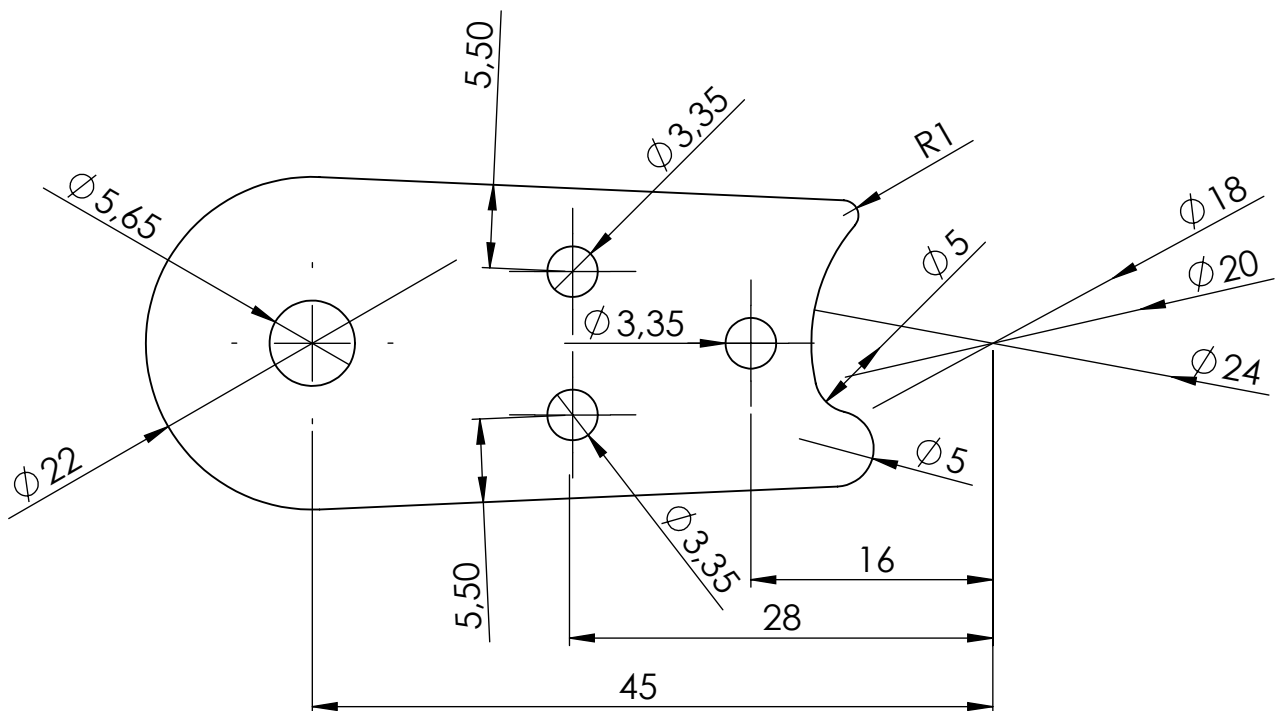
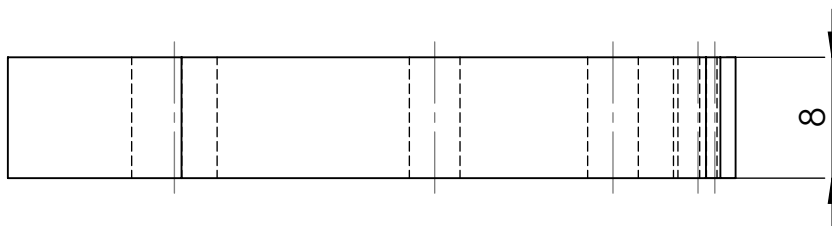
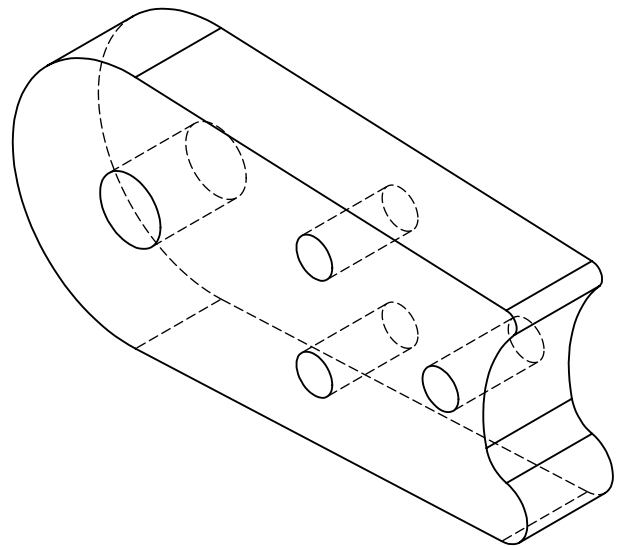
	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	MEDIAL 3		Plano N. 6
2:1			Sustituye a:
			Sustituido por:




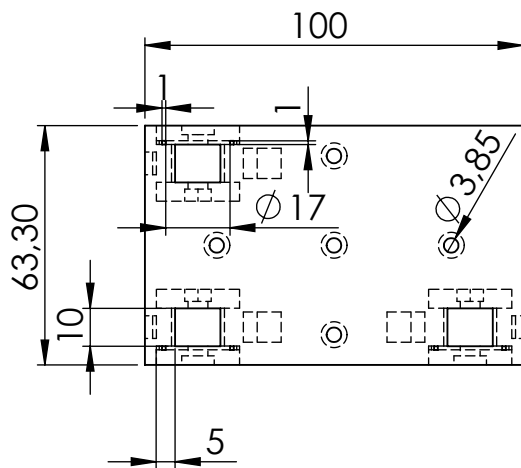
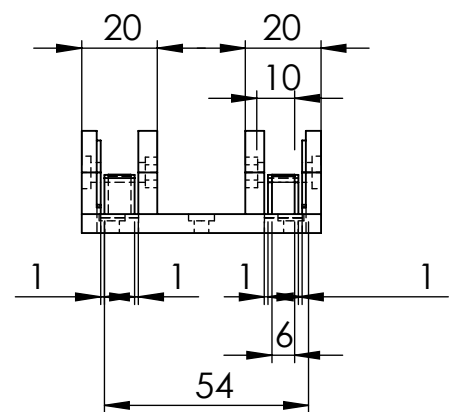
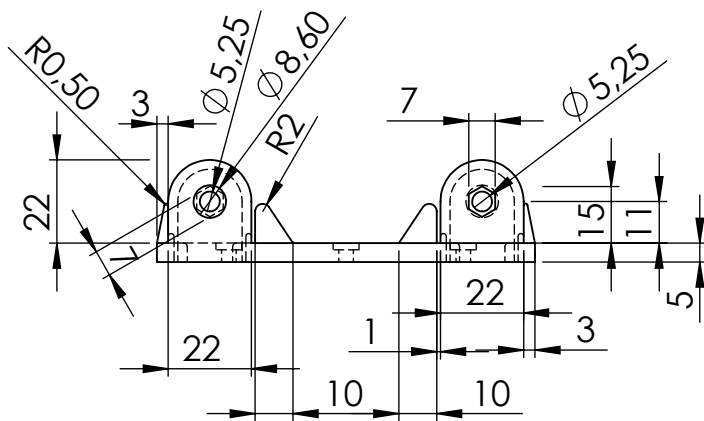
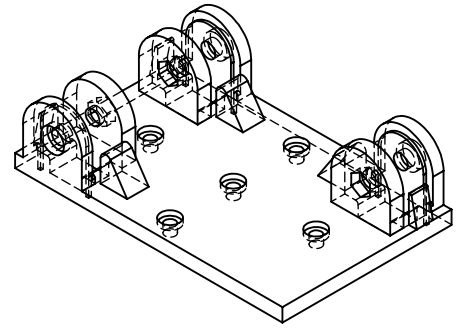
	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	PROXIMAL 1		Plano N. 7
2:1			Sustituye a:
			Sustituido por:



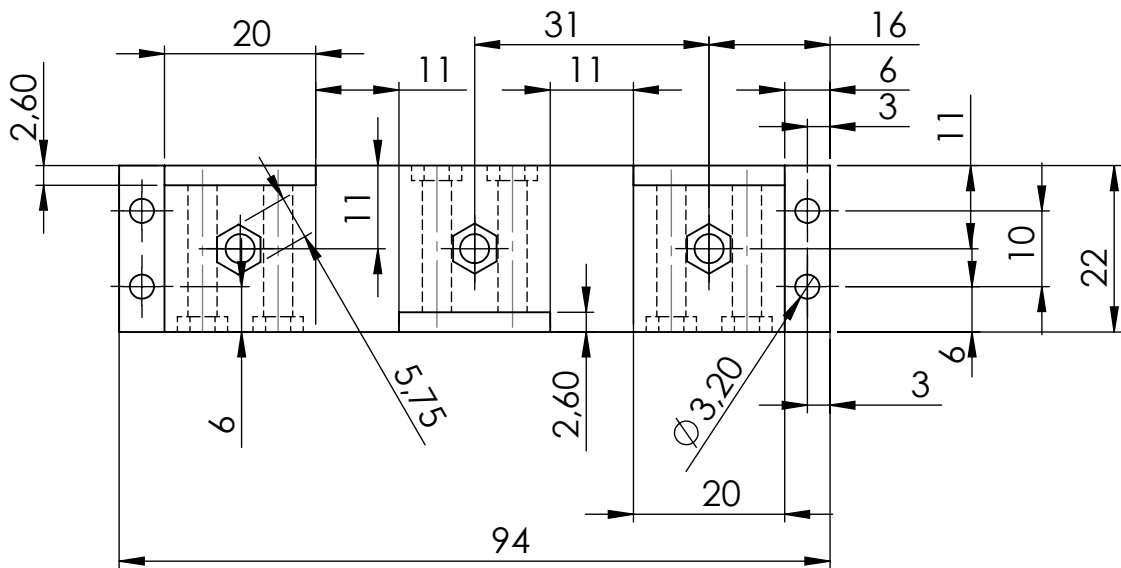
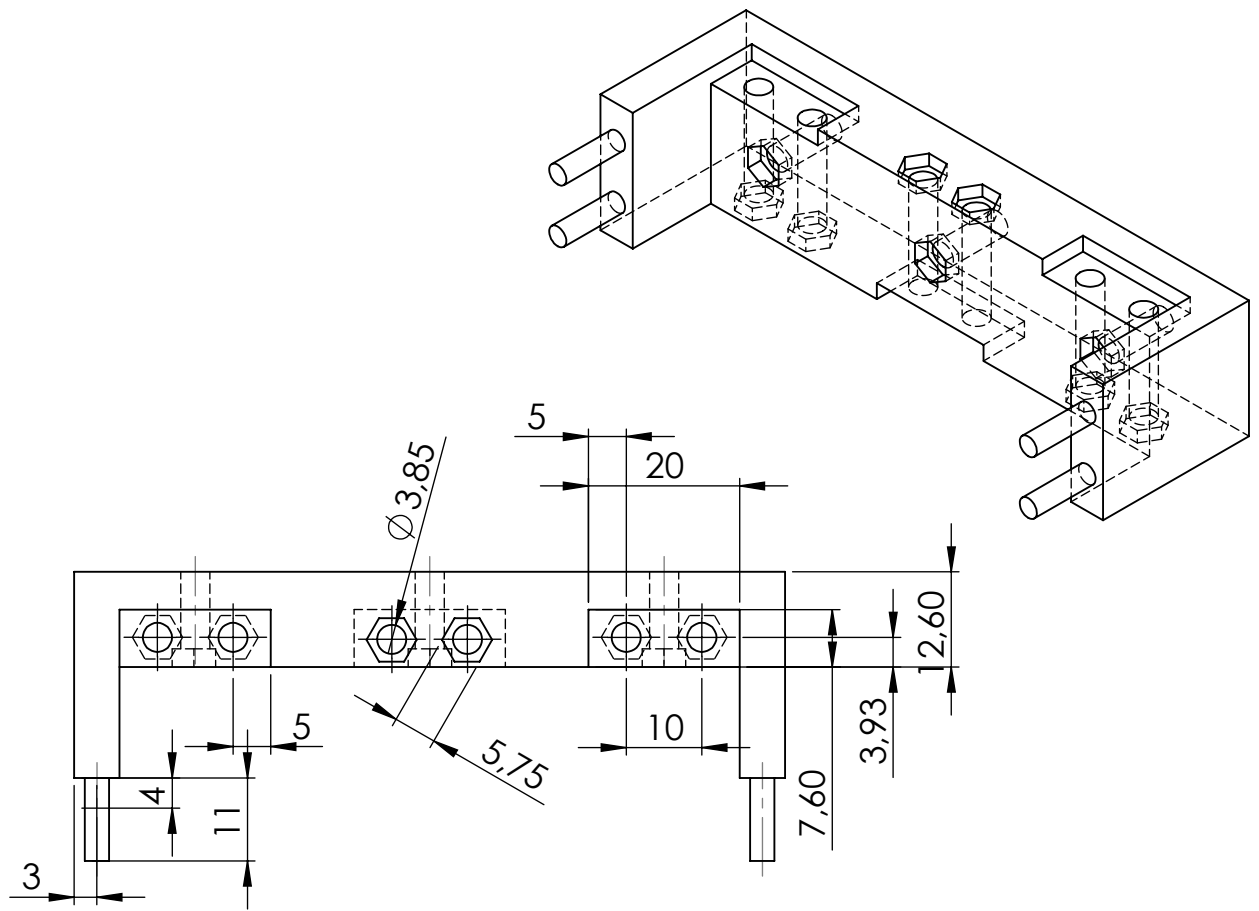
	Fecha	Nombre	 Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	PROXIMAL 2		Plano N. 8
2:1			Sustituye a:
			Sustituido por:



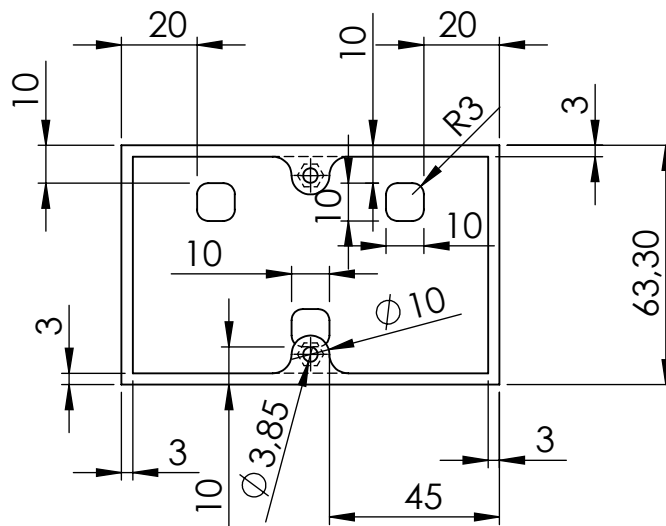
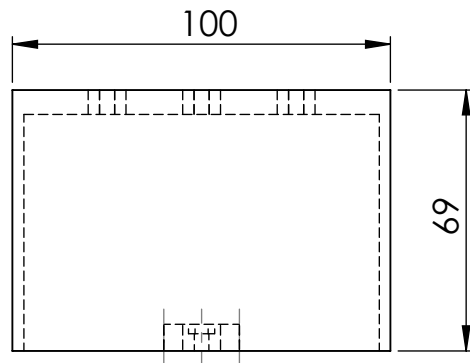
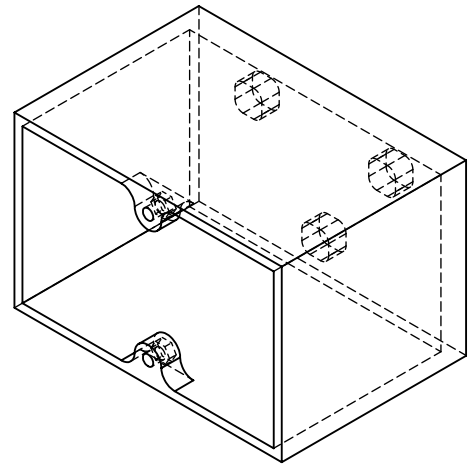
	Fecha	Nombre	 Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	PROXIMAL 3		Plano N. 9
2:1			Sustituye a:
			Sustituido por:



	<i>Fecha</i>	<i>Nombre</i>	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
<i>Dibujado:</i>	01/09/14	Jonathan	
<i>Comprobado:</i>			
<i>Id.s.norm:</i>			
<i>Escala</i>	PALMA MANO		Plano N. 10
1:1			Sustituye a:
			Sustituido por:



	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	<p>SOPORTE SERVO-MOTOR 2</p>		Plano N. 11 B
1:1			Sustituye a:
			Sustituido por:



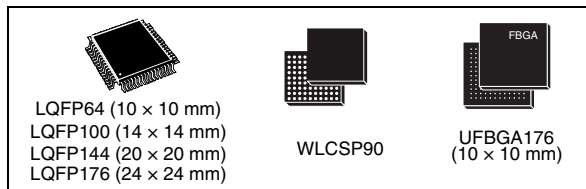
	Fecha	Nombre	 <p>Universidad Carlos III de Madrid Av. de la Universidad, 30, 28911, Madrid</p>
Dibujado:	01/09/14	Jonathan	
Comprobado:			
Id.s.norm:			
Escala	<h1>CARCASA EXTERIOR</h1>		Plano N. 12
1:2			Sustituye a:
			Sustituido por:



Anexo C: Datasheet

ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera

Datasheet - production data



Features

- Core: ARM 32-bit Cortex™-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
 - Up to 1 Mbyte of Flash memory
 - Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM
 - Flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- LCD parallel interface, 8080/6800 modes
- Clock, reset and supply management
 - 1.8 V to 3.6 V application supply and I/Os
 - POR, PDR, PVD and BOR
 - 4-to-26 MHz crystal oscillator
 - Internal 16 MHz factory-trimmed RC (1% accuracy)
 - 32 kHz oscillator for RTC with calibration
 - Internal 32 kHz RC with calibration
- Low power
 - Sleep, Stop and Standby modes
 - V_{BAT} supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
- 3×12-bit, 2.4 MSPS A/D converters: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 168 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- Debug mode
 - Serial wire debug (SWD) & JTAG interfaces
 - Cortex-M4 Embedded Trace Macrocell™
- Up to 140 I/O ports with interrupt capability
 - Up to 136 fast I/Os up to 84 MHz
 - Up to 138 5 V-tolerant I/Os
- Up to 15 communication interfaces
 - Up to 3 × I²C interfaces (SMBus/PMBus)
 - Up to 4 USARTs/2 UARTs (10.5 Mbit/s, ISO 7816 interface, LIN, IrDA, modem control)
 - Up to 3 SPIs (42 Mbits/s), 2 with muxed full-duplex I²S to achieve audio class accuracy via internal audio PLL or external clock
 - 2 × CAN interfaces (2.0B Active)
 - SDIO interface
- Advanced connectivity
 - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
 - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
 - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII
- 8- to 14-bit parallel camera interface up to 54 Mbytes/s
- True random number generator
- CRC calculation unit
- 96-bit unique ID
- RTC: subsecond accuracy, hardware calendar

Table 1. Device summary

Reference	Part number
STM32F405xx	STM32F405RG, STM32F405VG, STM32F405ZG, STM32F405OG, STM32F405OE
STM32F407xx	STM32F407VG, STM32F407IG, STM32F407ZG, STM32F407VE, STM32F407ZE, STM32F407IE

1 Introduction

This datasheet provides the description of the STM32F405xx and STM32F407xx lines of microcontrollers. For more details on the whole STMicroelectronics STM32™ family, please refer to [Section 2.1: Full compatibility throughout the family](#).

The STM32F405xx and STM32F407xx datasheet should be read in conjunction with the STM32F4xx reference manual.

The reference and Flash programming manuals are both available from the STMicroelectronics website www.st.com.

For information on the Cortex™-M4 core, please refer to the Cortex™-M4 programming manual (PM0214) available from www.st.com.

2 Description

The STM32F405xx and STM32F407xx family is based on the high-performance ARM[®] Cortex™-M4 32-bit RISC core operating at a frequency of up to 168 MHz. The Cortex-M4 core features a Floating point unit (FPU) single precision which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security. The Cortex-M4 core with FPU will be referred to as Cortex-M4F throughout this document.

The STM32F405xx and STM32F407xx family incorporates high-speed embedded memories (Flash memory up to 1 Mbyte, up to 192 Kbytes of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers, a true random number generator (RNG). They also feature standard and advanced communication interfaces.

- Up to three I²Cs
- Three SPIs, two I²Ss full duplex. To achieve audio class accuracy, the I2S peripherals can be clocked via a dedicated internal audio PLL or via an external clock to allow synchronization.
- Four USARTs plus two UARTs
- An USB OTG full-speed and a USB OTG high-speed with full-speed capability (with the ULPI),
- Two CANs
- An SDIO/MMC interface
- Ethernet and the camera interface available on STM32F407xx devices only.

New advanced peripherals include an SDIO, an enhanced flexible static memory control (FSMC) interface (for devices offered in packages of 100 pins and more), a camera interface for CMOS sensors. Refer to [Table 2: STM32F405xx and STM32F407xx: features and peripheral counts](#) for the list of peripherals available on each part number.

The STM32F405xx and STM32F407xx family operates in the –40 to +105 °C temperature range from a 1.8 to 3.6 V power supply. The supply voltage can drop to 1.7 V when the device operates in the 0 to 70 °C temperature range using an external power supply supervisor: refer to [Section : Internal reset OFF](#). A comprehensive set of power-saving mode allows the design of low-power applications.

The STM32F405xx and STM32F407xx family offers devices in various packages ranging from 64 pins to 176 pins. The set of included peripherals changes with the device chosen.

These features make the STM32F405xx and STM32F407xx microcontroller family suitable for a wide range of applications:

- Motor drive and application control
- Medical equipment
- Industrial applications: PLC, inverters, circuit breakers
- Printers, and scanners
- Alarm systems, video intercom, and HVAC
- Home audio appliances

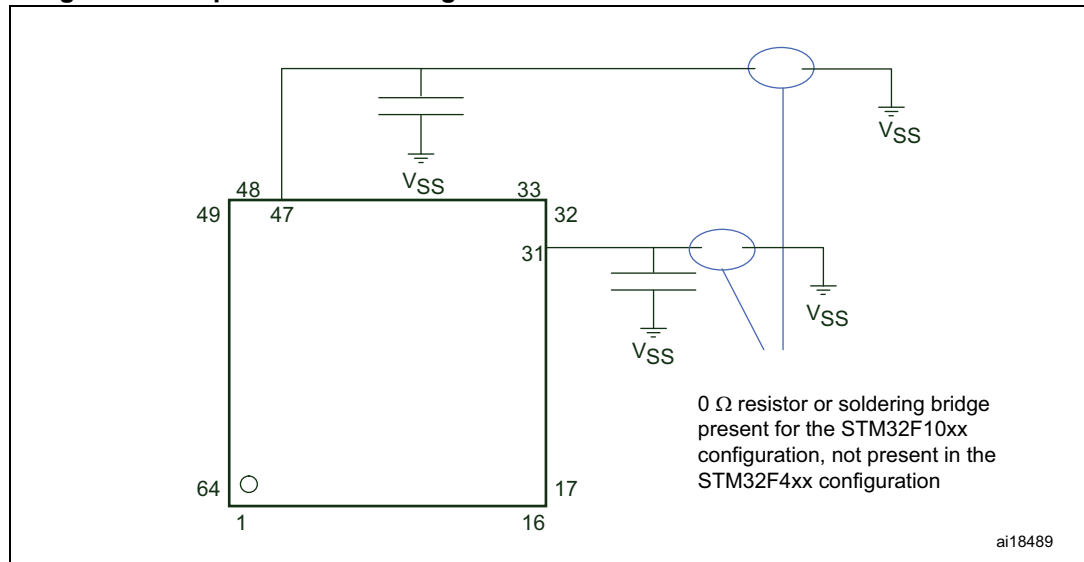
2.1 Full compatibility throughout the family

The STM32F405xx and STM32F407xx are part of the STM32F4 family. They are fully pin-to-pin, software and feature compatible with the STM32F2xx devices, allowing the user to try different memory densities, peripherals, and performances (FPU, higher frequency) for a greater degree of freedom during the development cycle.

The STM32F405xx and STM32F407xx devices maintain a close compatibility with the whole STM32F10xxx family. All functional pins are pin-to-pin compatible. The STM32F405xx and STM32F407xx, however, are not drop-in replacements for the STM32F10xxx devices: the two families do not have the same power scheme, and so their power pins are different. Nonetheless, transition from the STM32F10xxx to the STM32F40x family remains simple as only a few pins are impacted.

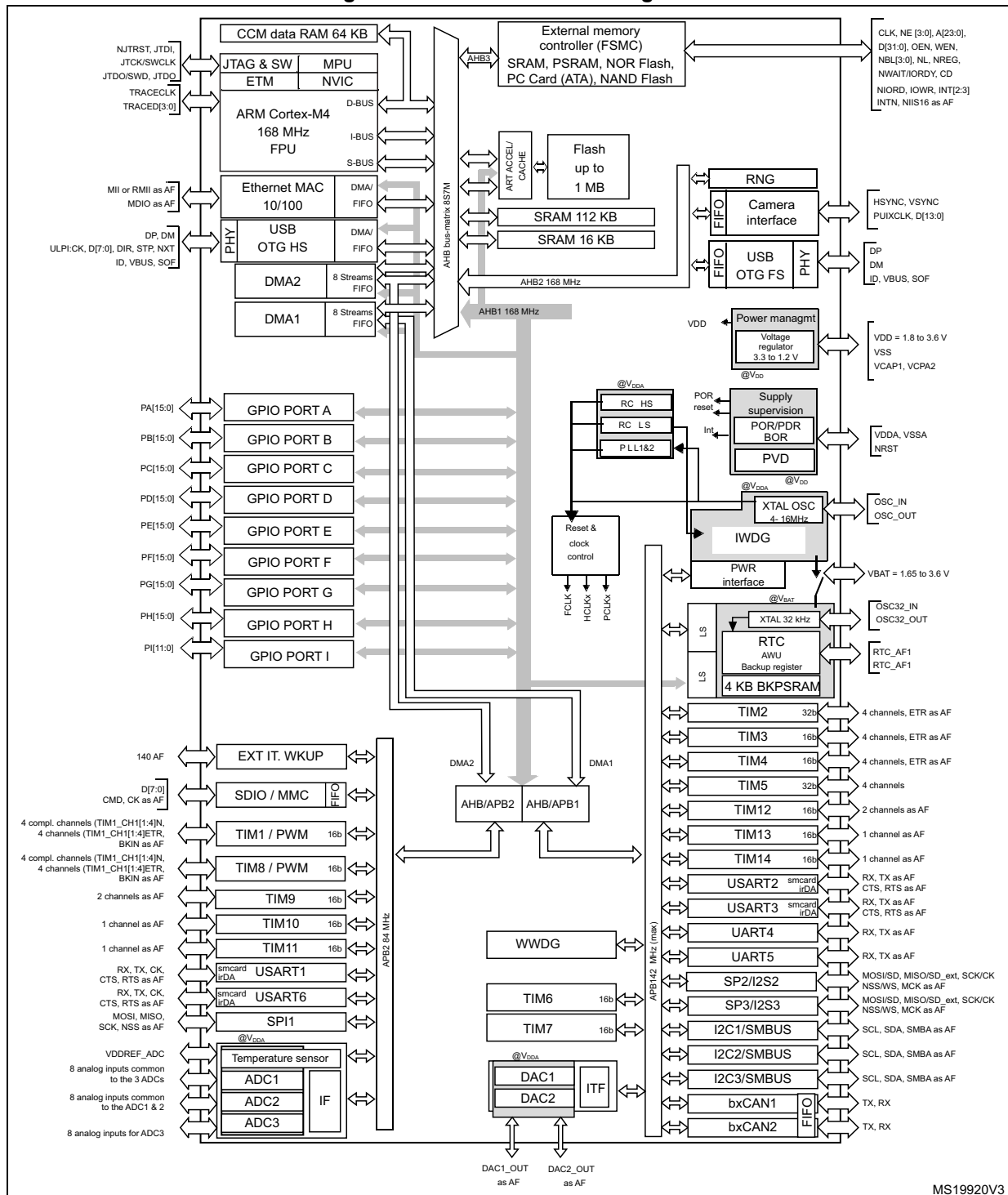
Figure 4, Figure 3, Figure 2, and Figure 1 give compatible board designs between the STM32F40x, STM32F2xxx, and STM32F10xxx families.

Figure 1. Compatible board design between STM32F10xx/STM32F4xx for LQFP64



2.2 Device overview

Figure 5. STM32F40x block diagram



1. The timers connected to APB2 are clocked from TIMxCLK up to 168 MHz, while the timers connected to APB1 are clocked from TIMxCLK either up to 84 MHz or 168 MHz, depending on TIMPRE bit configuration in the RCC_DCKCFGR register.
2. The camera interface and ethernet are available only on STM32F407xx devices.



2.2.1 ARM® Cortex™-M4F core with embedded Flash and SRAM

The ARM Cortex-M4F processor is the latest generation of ARM processors for embedded systems. It was developed to provide a low-cost platform that meets the needs of MCU implementation, with a reduced pin count and low-power consumption, while delivering outstanding computational performance and an advanced response to interrupts.

The ARM Cortex-M4F 32-bit RISC processor features exceptional code-efficiency, delivering the high-performance expected from an ARM core in the memory size usually associated with 8- and 16-bit devices.

The processor supports a set of DSP instructions which allow efficient signal processing and complex algorithm execution.

Its single precision FPU (floating point unit) speeds up software development by using metalanguage development tools, while avoiding saturation.

The STM32F405xx and STM32F407xx family is compatible with all ARM tools and software.

[Figure 5](#) shows the general block diagram of the STM32F40x family.

Note: Cortex-M4F is binary compatible with Cortex-M3.

2.2.2 Adaptive real-time memory accelerator (ART Accelerator™)

The ART Accelerator™ is a memory accelerator which is optimized for STM32 industry-standard ARM® Cortex™-M4F processors. It balances the inherent performance advantage of the ARM Cortex-M4F over Flash memory technologies, which normally requires the processor to wait for the Flash memory at higher frequencies.

To release the processor full 210 DMIPS performance at this frequency, the accelerator implements an instruction prefetch queue and branch cache, which increases program execution speed from the 128-bit Flash memory. Based on CoreMark benchmark, the performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from Flash memory at a CPU frequency up to 168 MHz.

2.2.3 Memory protection unit

The memory protection unit (MPU) is used to manage the CPU accesses to memory to prevent one task to accidentally corrupt the memory or resources used by any other active task. This memory area is organized into up to 8 protected areas that can in turn be divided up into 8 subareas. The protection area sizes are between 32 bytes and the whole 4 gigabytes of addressable memory.

The MPU is especially helpful for applications where some critical or certified code has to be protected against the misbehavior of other tasks. It is usually managed by an RTOS (real-time operating system). If a program accesses a memory location that is prohibited by the MPU, the RTOS can detect it and take action. In an RTOS environment, the kernel can dynamically update the MPU area setting, based on the process to be executed.

The MPU is optional and can be bypassed for applications that do not need it.

2.2.4 Embedded Flash memory

The STM32F40x devices embed a Flash memory of 512 Kbytes or 1 Mbytes available for storing programs and data.

2.2.5 CRC (cyclic redundancy check) calculation unit

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from a 32-bit data word and a fixed generator polynomial.

Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. In the scope of the EN/IEC 60335-1 standard, they offer a means of verifying the Flash memory integrity. The CRC calculation unit helps compute a software signature during runtime, to be compared with a reference signature generated at link-time and stored at a given memory location.

2.2.6 Embedded SRAM

All STM32F40x products embed:

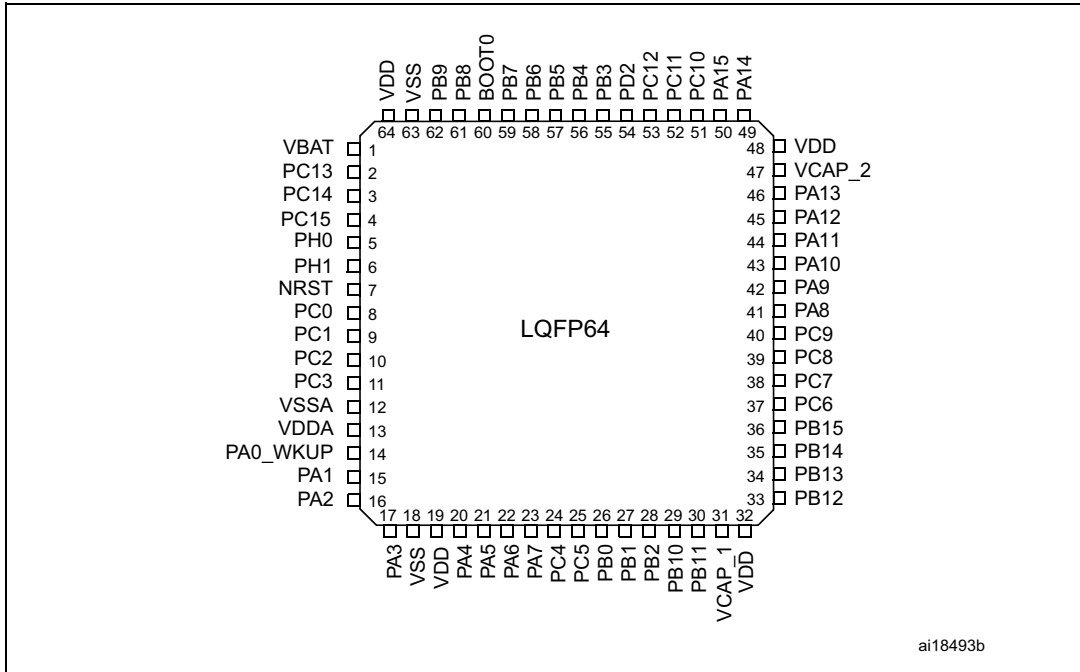
- Up to 192 Kbytes of system SRAM including 64 Kbytes of CCM (core coupled memory) data RAM
RAM memory is accessed (read/write) at CPU clock speed with 0 wait states.
- 4 Kbytes of backup SRAM
This area is accessible only from the CPU. Its content is protected against possible unwanted write accesses, and is retained in Standby or VBAT mode.

2.2.7 Multi-AHB bus matrix

The 32-bit multi-AHB bus matrix interconnects all the masters (CPU, DMAs, Ethernet, USB HS) and the slaves (Flash memory, RAM, FSMC, AHB and APB peripherals) and ensures a seamless and efficient operation even when several high-speed peripherals work simultaneously.

3 Pinouts and pin description

Figure 12. STM32F40x LQFP64 pinout



5 Electrical characteristics

5.1 Parameter conditions

Unless otherwise specified, all voltages are referenced to V_{SS} .

5.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at $T_A = 25\text{ }^\circ\text{C}$ and $T_A = T_{Amax}$ (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation ($\text{mean} \pm 3\Sigma$).

5.1.2 Typical values

Unless otherwise specified, typical data are based on $T_A = 25\text{ }^\circ\text{C}$, $V_{DD} = 3.3\text{ V}$ (for the $1.8\text{ V} \leq V_{DD} \leq 3.6\text{ V}$ voltage range). They are given only as design guidelines and are not tested.

Typical ADC accuracy values are determined by characterization of a batch of samples from a standard diffusion lot over the full temperature range, where 95% of the devices have an error less than or equal to the value indicated ($\text{mean} \pm 2\Sigma$).

5.1.3 Typical curves

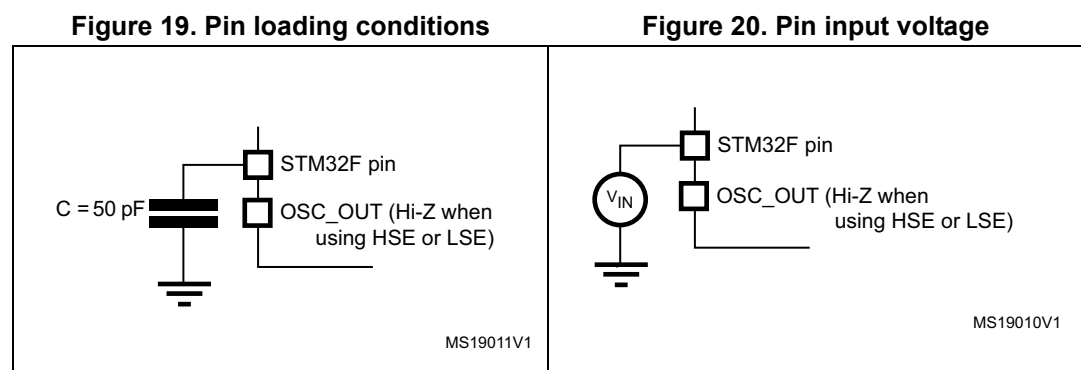
Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

5.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in [Figure 19](#).

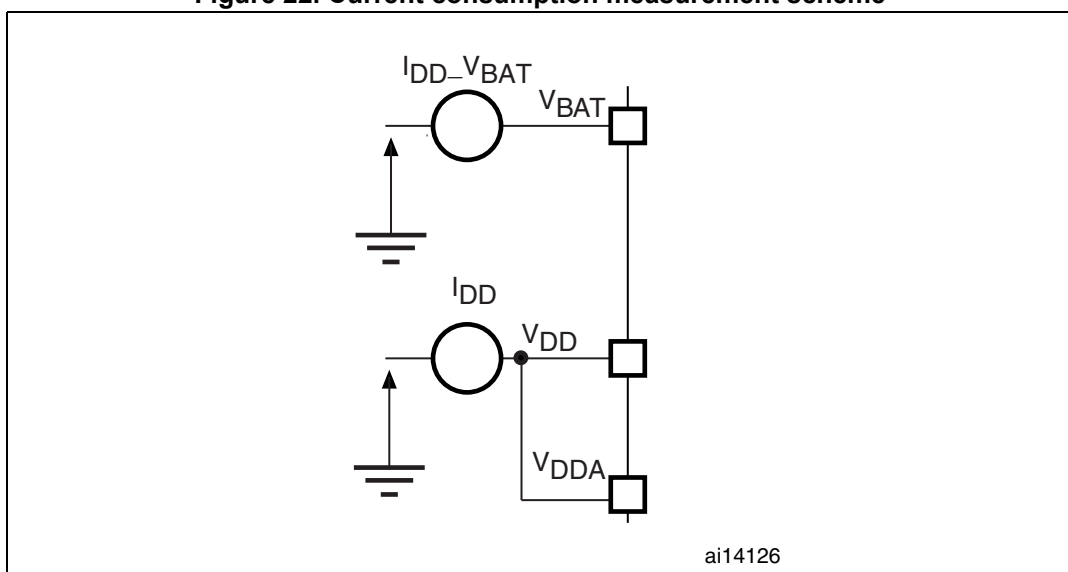
5.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in [Figure 20](#).



5.1.7 Current consumption measurement

Figure 22. Current consumption measurement scheme



5.2 Absolute maximum ratings

Stresses above the absolute maximum ratings listed in [Table 11: Voltage characteristics](#), [Table 12: Current characteristics](#), and [Table 13: Thermal characteristics](#) may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 11. Voltage characteristics

Symbol	Ratings	Min	Max	Unit
$V_{DD}-V_{SS}$	External main supply voltage (including V_{DDA} , V_{DD}) ⁽¹⁾	-0.3	4.0	V
V_{IN}	Input voltage on five-volt tolerant pin ⁽²⁾	$V_{SS}-0.3$	$V_{DD}+4$	
	Input voltage on any other pin	$V_{SS}-0.3$	4.0	
$ \Delta V_{DDx} $	Variations between different V_{DD} power pins	-	50	mV
$ V_{SSx} - V_{SS} $	Variations between all the different ground pins	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see Section 5.3.14: Absolute maximum ratings (electrical sensitivity)		

1. All main power (V_{DD} , V_{DDA}) and ground (V_{SS} , V_{SSA}) pins must always be connected to the external power supply, in the permitted range.
2. V_{IN} maximum value must always be respected. Refer to [Table 12](#) for the values of the maximum allowed injected current.

Table 12. Current characteristics

Symbol	Ratings	Max.	Unit
I_{VDD}	Total current into V_{DD} power lines (source) ⁽¹⁾	150	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink) ⁽¹⁾	150	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current source by any I/Os and control pin	25	
$I_{INJ(PIN)}$ ⁽²⁾	Injected current on five-volt tolerant I/O ⁽³⁾	-5/+0	
	Injected current on any other pin ⁽⁴⁾	±5	
$\Sigma I_{INJ(PIN)}$ ⁽⁴⁾	Total injected current (sum of all I/O and control pins) ⁽⁵⁾	±25	

- All main power (V_{DD} , V_{DDA}) and ground (V_{SS} , V_{SSA}) pins must always be connected to the external power supply, in the permitted range.
- Negative injection disturbs the analog performance of the device. See note in [Section 5.3.20: 12-bit ADC characteristics](#).
- Positive injection is not possible on these I/Os. A negative injection is induced by $V_{IN} < V_{SS}$. $I_{INJ(PIN)}$ must never be exceeded. Refer to [Table 11](#) for the values of the maximum allowed input voltage.
- A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. $I_{INJ(PIN)}$ must never be exceeded. Refer to [Table 11](#) for the values of the maximum allowed input voltage.
- When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values).

Table 13. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature	125	°C

5.3 Operating conditions

5.3.1 General operating conditions

Table 14. General operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{HCLK}	Internal AHB clock frequency	VOS bit in PWR_CR register = 0 ⁽¹⁾	0		144	MHz
		VOS bit in PWR_CR register = 1	0		168	
f_{PCLK1}	Internal APB1 clock frequency		0		42	
f_{PCLK2}	Internal APB2 clock frequency		0		84	
V_{DD}	Standard operating voltage		1.8 ⁽²⁾		3.6	V
V_{DDA} ⁽³⁾⁽⁴⁾	Analog operating voltage (ADC limited to 1.2 M samples)	Must be the same potential as V_{DD} ⁽⁵⁾	1.8 ⁽²⁾		2.4	V
	Analog operating voltage (ADC limited to 1.4 M samples)		2.4		3.6	
V_{BAT}	Backup operating voltage		1.65		3.6	V

UTCLM78XX LINEAR INTEGRATED CIRCUIT

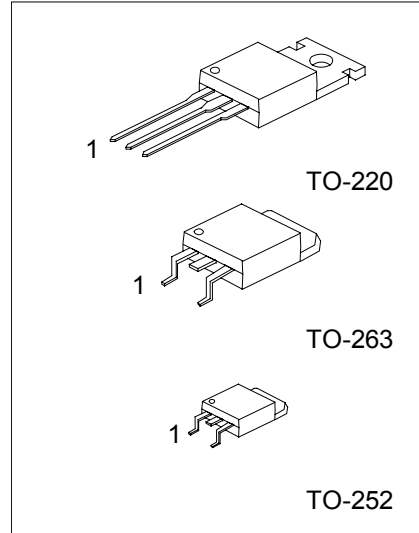
3-TERMINAL 1A POSITIVE VOLTAGE REGULATOR

DESCRIPTION

The UTC 78XX family is monolithic fixed voltage regulator integrated circuit. They are suitable for applications that required supply current up to 1 A.

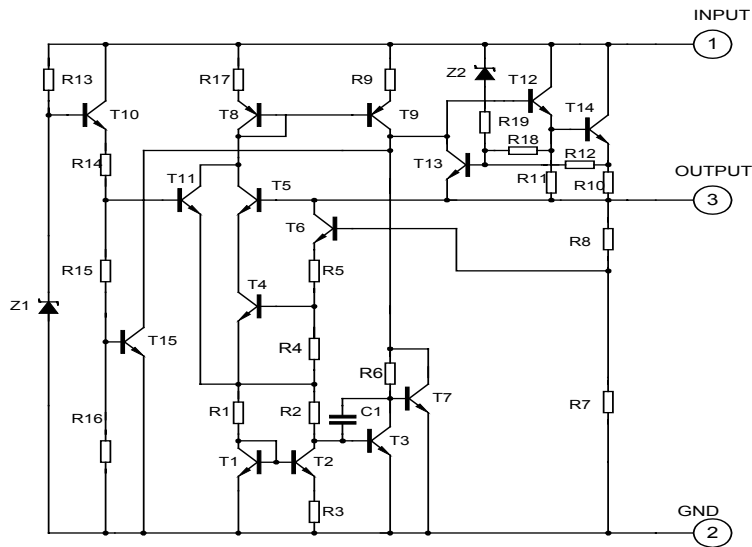
FEATURES

- *Output current up to 1.5 A
- *Fixed output voltage of 5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V and 24V available
- *Thermal overload shutdown protection
- *Short circuit current limiting
- *Output transistor SOA protection



1: Input 2: GND 3: Output

TEST CIRCUIT



UTC LM78XX LINEAR INTEGRATED CIRCUIT

ABSOLUTE MAXIMUM RATINGS

(Operating temperature range applies unless otherwise specified)

PARAMETER	SYMBOL	RATING	UNIT
Input voltage(for Vo=5~18V) (for Vo=24V)	Vi	35	V
		40	V
Output Current	Io	1	A
Power Dissipation	PD	Internally Limited	W
Operating Junction Temperature Range	TOPR	-20 +150	°C
Storage Temperature Range	TSTG	-55 +150	°C

UTC LM7805 ELECTRICAL CHARACTERISTICS

(Vi=10V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	4.80	5.0	5.20	V
		Vi =7.5V to 20V, Io=5mA - 1.0A,PD<15W	4.75		5.25	V
Load Regulation	ΔVo	Tj=25°C,Io=5mA - 1.5A			50	mV
		Tj=25°C,Io=0.25A - 0.75A			25	mV
Line regulation	ΔVo	Vi =7V to 25V,Tj=25°C			50	mV
		Vi =7.5V to 20V,Tj=25°C,Io=1A			50	mV
Quiescent Current	Iq	Tj=25°C, Io=<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =7.5V to 20V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		40		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-0.6		mV/°C
Ripple Rejection	RR	Vi =8V - 18V,f=120Hz,Tj=25°C	62	80		dB
Peak Output Current	IPK	Tj=25°C		1.8		A
Short-Circuit Current	Isc	Vi=35V, Tj=25°C		250		mA
Dropout Voltage	Vd	Tj=25°C		2.0		V

UTC LM7806 ELECTRICAL CHARACTERISTICS

(Vi=11V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	5.76	6.0	6.24	V
		Vi =8.5V to 21V, Io=5mA - 1.0A, PD<15W	5.70		6.30	V
Load Regulation	ΔVo	Tj=25°C,Io=5mA - 1.5A			60	mV
		Tj=25°C,Io=0.25A - 0.75A			30	mV
Line regulation	ΔVo	Vi =8V to 25V,Tj=25°C			60	mV
		Vi =8.5V to 21V,Tj=25°C,Io=1A			60	mV
Quiescent Current	Iq	Tj=25°C, Io=<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =8.5V to 21V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		45		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-0.7		mV/°C
Ripple Rejection	RR	Vi =9V - 19V,f=120Hz,Tj=25°C	59	75		dB

UTC UNISONIC TECHNOLOGIES CO., LTD. 2

UTC LM78XX LINEAR INTEGRATED CIRCUIT

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Peak Output Current	IPK	Tj=25°C		1.8		A
Short-Circuit Current	Isc	VI=35V, Tj=25°C		250		mA
Dropout Voltage	Vd	Tj=25°C		2.0		V

UTC LM7808 ELECTRICAL CHARACTERISTICS

(VI=14V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	7.68	8.0	8.32	V
		Vi =10.5V to 23V, Io=5mA - 1.0A, PD<15W	7.60		8.40	V
Load Regulation	ΔVo	Tj=25°C, Io=5mA - 1.5A			80	mV
		Tj=25°C, Io=0.25A - 0.75A			40	mV
Line regulation	ΔVo	Vi =10.5V to 25V, Tj=25°C			80	mV
		Vi =10.5V to 23V, Tj=25°C, Io=1A			80	mV
Quiescent Current	Iq	Tj=25°C, Io<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =10.5V to 23V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		58		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-0.9		mV/°C
Ripple Rejection	RR	Vi =11.5V to 21.5V, f=120Hz, Tj=25°C	56	72		dB
Peak Output Current	IPK	Tj=25°C		1.8		A
Short-Circuit Current	Isc	VI=35V, Tj=25°C		250		mA
Dropout Voltage	Vd	Tj=25°C		2.0		V

UTC LM7809 ELECTRICAL CHARACTERISTICS

(VI=15V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	Vo	Tj=25°C, Io=5mA - 1.0A	8.64	9.0	9.36	V
		Vi =11.5V to 24V, Io=5mA - 1.0A, PD<15W	8.55		9.45	V
Load Regulation	ΔVo	Tj=25°C, Io=5mA - 1.5A			90	mV
		Tj=25°C, Io=0.25A - 0.75A			45	mV
Line regulation	ΔVo	Vi =11.5V to 25 V, Tj=25°C, PD<15W			90	mV
		Vi =11.5V to 24V, Tj=25°C, Io<=1A			90	mV
Quiescent Current	Iq	Tj=25°C, Io<1A			8.0	mA
Quiescent Current Change	ΔIq	Vi =11.5V to 24V			1.0	mA
	ΔIq	Io=5mA - 1.0A			0.5	mA
Output Noise Voltage	VN	10Hz<=f<=100kHz		58		μV
Temperature coefficient of Vo	ΔVo/ΔT	Io=5mA		-1.1		mV/°C
Ripple Rejection	RR	Vi =12.5V to 22.5V, f=120Hz, Tj=25°C	56	72		dB
Peak Output Current	IPK	Tj=25°C		1.8		A
Short-Circuit Current	Isc	VI=35V, Tj=25°C		250		mA
Dropout Voltage	Vd	Tj=25°C		2.0		V

UTC LM78XX LINEAR INTEGRATED CIRCUIT

UTC LM7810 ELECTRICAL CHARACTERISTICS

($V_I=16V$, $I_o=0.5A$, $T_j=0^\circ C - 125^\circ C$, $C_1=0.33\mu F$, $C_o=0.1\mu F$, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j=25^\circ C$, $I_o=5mA - 1.0A$	9.60	10.0	10.40	V
		$V_I=12.5V$ to $25V$, $I_o=5mA - 1.0A$, $PD \leq 15W$	9.50		10.50	V
Load Regulation	ΔV_o	$T_j=25^\circ C$, $I_o=5mA - 1.5A$			100	mV
		$T_j=25^\circ C$, $I_o=0.25A - 0.75A$			50	mV
Line regulation	ΔV_o	$V_I=13V$ to $25V$, $T_j=25^\circ C$			100	mV
		$V_I=13V$ to $25V$, $T_j=25^\circ C$, $I_o \leq 1A$			100	mV
Quiescent Current	I_q	$T_j=25^\circ C$, $I_o < 1A$			8.0	mA
Quiescent Current Change	ΔI_q	$V_I=12.6V$ to $25V$			1.0	mA
	ΔI_q	$I_o=5mA - 1.0A$			0.5	mA
Output Noise Voltage	V_N	$10Hz \leq f \leq 100kHz$		58		μV
Temperature coefficient of V_o	$\Delta V_o/\Delta T$	$I_o=5mA$		-1.1		$mV/^\circ C$
Ripple Rejection	RR	$V_I=13V - 23V$, $f=120Hz$, $T_j=25^\circ C$	56	72		dB
Peak Output Current	I_{PK}	$T_j=25^\circ C$		1.8		A
Short-Circuit Current	I_{SC}	$V_I=35V$, $T_j=25^\circ C$		250		mA
Dropout Voltage	V_d	$T_j=25^\circ C$		2.0		V

UTC LM7812 ELECTRICAL CHARACTERISTICS

($V_I=19V$, $I_o=0.5A$, $T_j=0^\circ C - 125^\circ C$, $C_1=0.33\mu F$, $C_o=0.1\mu F$, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j=25^\circ C$, $I_o=5mA - 1.0A$	11.52	12.0	12.48	V
		$V_I=14.5V$ to $27V$, $I_o=5mA - 1.0A$, $PD \leq 15W$	11.40		12.60	V
Load Regulation	ΔV_o	$T_j=25^\circ C$, $I_o=5mA - 1.5A$			120	mV
		$T_j=25^\circ C$, $I_o=0.25A - 0.75A$			60	mV
Line regulation	ΔV_o	$V_I=14.5V$ to $30V$, $T_j=25^\circ C$			120	mV
		$V_I=14.6V$ to $27V$, $T_j=25^\circ C$, $I_o=1A$			120	mV
Quiescent Current	I_q	$T_j=25^\circ C$, $I_o < 1A$			8.0	mA
Quiescent Current Change	ΔI_q	$V_I=14.5V$ to $30V$			1.0	mA
	ΔI_q	$I_o=5mA - 1.0A$			0.5	mA
Output Noise Voltage	V_N	$10Hz \leq f \leq 100kHz$		75		μV
Temperature coefficient of V_o	$\Delta V_o/\Delta T$	$I_o=5mA$		-1.5		$mV/^\circ C$
Ripple Rejection	RR	$V_I=15V - 25V$, $f=120Hz$, $T_j=25^\circ C$	55	72		dB
Peak Output Current	I_{PK}	$T_j=25^\circ C$		1.8		A
Short-Circuit Current	I_{SC}	$V_I=35V$, $T_j=25^\circ C$		250		mA
Dropout Voltage	V_d	$T_j=25^\circ C$		2.0		V

UTC LM78XX LINEAR INTEGRATED CIRCUIT

UTC LM7815 ELECTRICAL CHARACTERISTICS

($V_I=23V$, $I_o=0.5A$, $T_j=0^\circ C - 125^\circ C$, $C_1=0.33\mu F$, $C_o=0.1\mu F$, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j=25^\circ C$, $I_o=5mA - 1.0A$	14.40	15.0	15.60	V
		$V_I=17.5V$ to $30V$, $I_o=5mA - 1.0A$, $PD<15W$	14.25		15.75	V
Load Regulation	ΔV_o	$T_j=25^\circ C$, $I_o=5mA - 1.5A$			150	mV
		$T_j=25^\circ C$, $I_o=0.25A - 0.75A$			75	mV
Line regulation	ΔV_o	$V_I=18.5V$ to $30V$, $T_j=25^\circ C$			150	mV
		$V_I=17.7V$ to $30V$, $T_j=25^\circ C$, $I_o=1A$			150	mV
Quiescent Current	I_q	$T_j=25^\circ C$, $I_o<1A$			8.0	mA
Quiescent Current Change	ΔI_q	$V_I=17.5V$ to $30V$			1.0	mA
	ΔI_q	$I_o=5mA - 1.0A$			0.5	mA
Output Noise Voltage	V_N	$10Hz \leq f \leq 100kHz$		90		μV
Temperature coefficient of V_o	$\Delta V_o/\Delta T$	$I_o=5mA$		-1.8		$mV/^\circ C$
Ripple Rejection	RR	$V_I=18.5V$ to $28.5V$ $f=120Hz$, $T_j=25^\circ C$	54	70		dB
Peak Output Current	I_{PK}	$T_j=25^\circ C$		1.8		A
Short-Circuit Current	I_{SC}	$V_I=35V$, $T_j=25^\circ C$		250		mA
Dropout Voltage	V_d	$T_j=25^\circ C$		2.0		V

UTC LM7818 ELECTRICAL CHARACTERISTICS

($V_I=27V$, $I_o=0.5A$, $T_j=0^\circ C - 125^\circ C$, $C_1=0.33\mu F$, $C_o=0.1\mu F$, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j=25^\circ C$, $I_o=5mA - 1.0A$	17.28	18.0	18.72	V
		$V_I=21V$ to $33V$, $I_o=5mA - 1.0A$	17.10		18.90	V
Load Regulation	ΔV_o	$T_j=25^\circ C$, $I_o=5mA - 1.5A$			180	mV
		$T_j=25^\circ C$, $I_o=0.25A - 0.75A$			90	mV
Line regulation	ΔV_o	$V_I=21V$ to $33V$, $T_j=25^\circ C$			180	mV
		$V_I=21V$ to $33V$, $T_j=25^\circ C$, $I_o \leq 1A$, $PD<15W$			180	mV
Quiescent Current	I_q	$T_j=25^\circ C$, $I_o \leq 1A$			8.0	mA
Quiescent Current Change	ΔI_q	$V_I=21.5V$ to $33V$			1.0	mA
	ΔI_q	$I_o=5mA - 1.0A$			0.5	mA
Output Noise Voltage	V_N	$10Hz \leq f \leq 100kHz$		110		μV
Temperature coefficient of V_o	$\Delta V_o/\Delta T$	$I_o=5mA$		-2.2		$mV/^\circ C$
Ripple Rejection	RR	$V_I=22V - 32V$, $f=120Hz$, $T_j=25^\circ C$	53	69		dB
Peak Output Current	I_{PK}	$T_j=25^\circ C$		1.8		A
Short-Circuit Current	I_{SC}	$V_I=35V$, $T_j=25^\circ C$		250		mA
Dropout Voltage	V_d	$T_j=25^\circ C$		2.0		V

UTC LM78XX LINEAR INTEGRATED CIRCUIT

UTC LM7824 ELECTRICAL CHARACTERISTICS

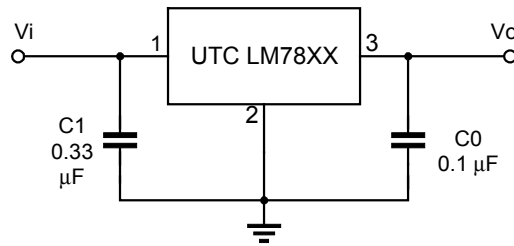
($V_i=33V$, $I_o=0.5A$, $T_j=0^\circ C - 12^\circ C$, $C_1=0.33\mu F$, $C_o=0.1\mu F$, unless otherwise specified)(Note 1)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j=25^\circ C$, $I_o=5mA - 1.0A$	23.04	24.0	24.96	V
		$V_i=27V$ to $38V$, $I_o=5mA - 1.0A$	22.80		25.20	V
Load Regulation	ΔV_o	$T_j=25^\circ C$, $I_o=5mA - 1.5A$			240	mV
		$T_j=25^\circ C$, $I_o=0.25A - 0.75A$			120	mV
Line regulation	ΔV_o	$V_i=27V$ to $38V$, $T_j=25^\circ C$			240	mV
		$V_i=27V$ to $38V$, $T_j=25^\circ C$, $I_o=1A$			240	mV
Quiescent Current	I_q	$T_j=25^\circ C$, $I_o < 1A$			8.0	mA
Quiescent Current Change	ΔI_q	$V_i=28V$ to $38V$			1.0	mA
	ΔI_q	$I_o=5mA - 1.0A$			0.5	mA
Output Noise Voltage	V_N	$10Hz \leq f \leq 100kHz$		170		μV
Temperature coefficient of V_o	$\Delta V_o / \Delta T$	$I_o=5mA$		-2.8		$mV/^\circ C$
Ripple Rejection	RR	$V_i=28V - 38V$, $f=120Hz$, $T_j=25^\circ C$	50	66		dB
Peak Output Current	I_{PK}	$T_j=25^\circ C$		1.8		A
Short-Circuit Current	I_{SC}	$V_i=35V$, $T_j=25^\circ C$		250		mA
Dropout Voltage	V_d	$T_j=25^\circ C$		2.0		V

Note 1: The Maximum steady state usable output current are dependent on input voltage, heat sinking, lead length of the package and copper pattern of PCB. The data above represents pulse test conditions with junction temperatures specified at the initiation of test.

Note 2: Power dissipation $< 0.5W$

APPLICATION CIRCUIT



Note 1: To specify an output voltage, substitute voltage value for "XX".

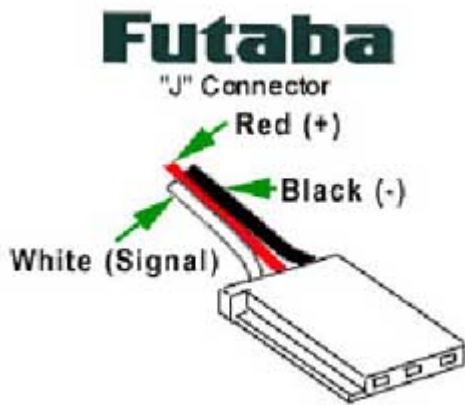
Note 2: Bypass capacitors are recommended for optimum stability and transient response and should be located as close as possible to the regulators.

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

S3003 FUTABA SERVO



...S3003 FUTABA SERVO...

Detailed Specifications

Control System:	+Pulse Width Control 1520usec Neutral	Current Drain (4.8V):	7.2mA/idle
Required Pulse:	3-5 Volt Peak to Peak Square Wave	Current Drain (6.0V):	8mA/idle
Operating Voltage:	4.8-6.0 Volts	Direction:	Counter Clockwise/Pulse Traveling 1520-1900usec
Operating Temperature Range:	-20 to +60 Degree C	Motor Type:	3 Pole Ferrite
Operating Speed (4.8V):	0.23sec/60 degrees at no load	Potentiometer Drive:	Indirect Drive
Operating Speed (6.0V):	0.19sec/60 degrees at no load	Bearing Type:	Plastic Bearing
Stall Torque (4.8V):	44 oz/in. (3.2kg.cm)	Gear Type:	All Nylon Gears
Stall Torque (6.0V):	56.8 oz/in. (4.1kg.cm)	Connector Wire Length:	12"
Operating Angle:	45 Deg. one side pulse traveling 400usec	Dimensions:	1.6" x 0.8"x 1.4" (41 x 20 x 36mm)
360 Modifiable:	Yes	Weight:	1.3oz. (37.2g)