

ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE SEVILLA



Trabajo Fin de Grado en Ingeniería en Electrónica Industrial

CONTROL EMPOTRADO DE SISTEMA KAFO

Autor: Jose Manuel Paez Romero

Tutores: Alejandro Linares Barranco

Sevilla, septiembre de 2019

ÍNDICE GENERAL

Normativa y Procedimiento	7
Definiciones y Abreviaturas	9
CAPÍTULO 1	11
1. Introducción.....	11
2. Antecedentes.....	12
CAPÍTULO 2	14
1. Biomecánica del miembro inferior	14
2. Diseño del prototipo.....	19
3. Sistema de control.....	23
4. Actuadores y Sensores	26
5. Alimentación del sistema y otros	30
CAPÍTULO 3	32
1. Análisis del Sistema de Control.....	32
2. Entorno de Programación y otros programas	39
CAPÍTULO 4	46
1. Planificación	46
2. Puesta en marcha	47
3. Conclusiones	50

4. Líneas de trabajos futuros	52
CAPÍTULO 5	53
1. Presupuesto	53
CAPÍTULO 6	57
1. Ficha técnica (DATASHEET)	57
Bibliografía	59
Anexo I	61
1. Código del Proyecto	61
Anexo II	62
1. Diagrama de Gant y Calendario	62
Anexo III	63
2. Planos del dispositivo	63

Índice de Tablas

Tabla 1: Presupuesto Hardware.....	53
Tabla 2: Presupuesto Software	54
Tabla 3: Presupuesto del Personal	55
Tabla 4: Presupuesto de ejecución del Proyecto	56

Índice de Figuras

Figura 1: Dispositivo diseñado	11
Figura 2: Ortesis HKAFO	12
Figura 3: Ángulos bipedestación [6]	15
Figura 4: Ejes del cuerpo humano [7].....	16
Figura 5: Ángulos rodilla [6].....	16
Figura 6:Ángulos tobillo [6].....	17
Figura 7:Simulación posiciones extremas de nuestro mecanismo.....	17
Figura 8:Posiciones extremas de nuestro mecanismo.....	18
Figura 9:KAFO Bitutor [8].....	19
Figura 10:Tutor Muslo	20
Figura 11:Rotor Muslo.....	20
Figura 12:Articulación de rodilla	20
Figura 13:Tutor gemelo	21
Figura 14:Rotor gemelo.....	21
Figura 15: Soporte del Pie.....	21
Figura 16: Estructura completa dispositivo.....	22
Figura 17:Placa desarrolladora [9]	23
Figura 18: Motor paso a paso	26
Figura 19:Servomotor digital	27

Figura 20: Características Servomotor.....	27
Figura 21:IMU 5DOF	27
Figura 22: Características MPU 6050	28
Figura 23:MPU 6050	28
Figura 24: Características RCW-0001.....	29
Figura 25:RCW-0001	29
Figura 26: Sensor de ultrasonidos	29
Figura 27:Batería externa.....	30
Figura 28:Módulo regulador de voltaje	30
Figura 29: Módulo convertidor de USB a TTL.....	31
Figura 30: Características Módulo convertidor USB a TTL [10]	31
Figura 31:Esquema básico reloj sistema [11].....	32
Figura 32: Configuración Reloj del sistema.....	33
Figura 33:Delay 1us	34
Figura 34:Delay 1ms	34
Figura 35:Pines PWM	35
Figura 36: Configuración común Timer	36
Figura 37:Configuración Interrupción Timer.....	36
Figura 38:Prioridad de las interrupciones.....	36
Figura 39: Configuración comunicación I2C.....	37
Figura 40: Configuración comunicación USART	37
Figura 41: Configuración GPIO's.....	38
Figura 42: GPIO's modos de funcionamiento.....	38
Figura 43: IDE ECLIPSE [12].....	39
Figura 44: Stdperiph.....	40
Figura 45:Pasos para crear un proyecto	41
Figura 46:Analizador lógico.....	42

Figura 47: Programa Logic Analyzer	43
Figura 48: Características Tera Term.....	43
Figura 49Tera term configuración 1	43
Figura 50:Tera Term configuración 2	44
Figura 51: Dispositivo diseñador en INVENTOR.....	44
Figura 52: Inventor pieza completa	45
Figura 53:Posición 1(Inicio).....	47
Figura 54: Posición 2.....	48
Figura 55:Posición 3.....	48
Figura 56. Posición 4.....	48
Figura 57:Posición 5(final).....	49

Normativa y Procedimiento

Este proyecto ha sido guiado en su elaboración a través de las siguientes Normas UNE.

- UNE 157001:2014. Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico.
- UNE-ISO 999:2014. Información y documentación. Directrices sobre el contenido, la organización y presentación de índices.
- UNE-ISO 690:2013. Información y documentación. Directrices para la redacción de referencias bibliográficas y de citas de recursos de información.
- UNE 50103:1990. Resúmenes.
- UNE 1027:1995. Dibujos técnicos. Plegado de planos.
- UNE 1032:1982. Dibujos técnicos. Principios generales de representación.
- UNE-EN 62023:2002. Estructuración de la información y documentación técnicas.
- UNE 1039:1994. Dibujos técnicos. Acotación.
- UNE-EN ISO 5456-1:1996. Dibujos, técnicos. Métodos de proyección. Parte 1: Sinopsis.

- UNE-EN ISO 5456-2:1996. Dibujos técnicos. Métodos de proyección. Parte 2: Representaciones ortográficas.
- UNE-EN ISO 5456-3:1996. Dibujos técnicos. Métodos de proyección. Parte 3: Representaciones axonométricas.
- UNE-EN ISO 5457:1999. Documentación técnica de producto. Formatos y presentación de los elementos gráficos de las hojas de dibujo.
- UNE 50136: Documentación. Presentación de tesis y documentos similares.
- Normas de realización de los TFG. Documento para poder presentar un TFG en la Escuela Politécnica Superior de Sevilla.

Definiciones y Abreviaturas

- Definiciones:

Órtesis: Es un dispositivo o mecanismos que se adapta al paciente, para ayudar en alguna deficiencia motriz ya sea en el miembro inferior o superior del cuerpo humano. [1]

Bipedestación: Bipedestación significa estar o permanecer de pie. [2]

Sedestación: Mantenerse sentado de forma autónoma sin necesidad de apoyo externo.

Sistema empotrado o embebido: Los sistemas empotrados son sistemas informáticos dedicados a aplicaciones de control. [3]

Microcontrolador: Es un circuito integrado que consta de una CPU, memorias (RAM y ROM), periféricos, puertos de entrada y de salida. [4]

Sensor de proximidad: Sensor que detecta un objeto u obstáculo dentro de un rango determinado.

- Abreviaturas:

TTL: Dispositivo de comunicación Lógica transistor-transistor. Transistor-transistor logic.

USB: Periférico Universal serial bus. Bus universal serial

IDE: Integrated Development Environment. Entorno de desarrollo integrado

IMU: inertial measurement unit. Una unidad de medición inercial

MPU: Micro Processing Unit. Unidad de microprocesamiento

MEMS: MicroElectroMechanical Systems. Sistemas micro-electromecánicos

PWM: Pulse width Moduation. Modulación por ancho de pulso.

I2C: Inter integrated circuit.

USART: Universal Synchronous-Asynchronous Receiver-Transmitter.

HKAFO: Hit-knee-ankle-foot.

KAFO: knee-ankle-foot.

CAPÍTULO 1

1. Introducción

Este proyecto contempla el estudio y análisis de un sistema que pueda ayudar a las personas con movilidad reducida en su miembro inferior. En él se verán materias tanto de mecánica como de electrónica, pasando del análisis de las extremidades inferiores del ser humano como a la aplicación de tecnologías actuales en sistemas de ayuda como son las órtesis.

Nuestro proyecto describe los movimientos necesarios para que el usuario pueda llegar a realizar la acción de sedestación y bipedestación, para ello se ha diseñado un sencillo mecanismo de un solo tutor que abarca una pierna de un individuo adulto a escala 1:2.



Figura 1: Dispositivo diseñado

2. Antecedentes

Desde la antigüedad el ser humano se las ha ingeniado para poder facilitar la movilidad cuando un miembro, ya sea por enfermedad o por accidente, ha perdido su función o parte de ella. Para ello están las órtesis o prótesis, herramientas que han ido evolucionando desde un simple apoyo rudimentario hasta dispositivos con inteligencia capaz de poder adaptarse a las diferentes situaciones cotidianas del día a día.



Figura 2: Ortesis HKAFO

Pero, ¿qué es lo que nos ha hecho decidir a realizar un proyecto de estas características? La respuesta es sencilla, hay un creciente movimiento en el mundo, personas que con una simple impresora 3D hacen realidad proyectos con un análisis profundo y detalles dignos de proyectos de investigación, por lo tanto, es aquí donde nace la necesidad de poder aplicar los conocimientos desarrollados durante mis años en la carrera al concepto de ortesis, más concretamente al aparato de marcha. Dicha órtesis, el aparato de marcha, en la cual nos centraremos también es conocida por sus siglas en ingles KAFO que comprende rodilla (knee),

tobillo (ankle) y pie (foot). En medida que la ortesis sube hacia el torso se añaden articulaciones y, por lo tanto, siglas en ingles de dichas articulaciones.

Para nuestro proyecto se usará como paciente una pierna diseñada en 3D [5], la función de nuestra órtesis será la de ayudar al paciente a sentarse y a levantarse, o lo que es lo mismo, realizar la acción de bipedestación y sedestación, para ello nuestra órtesis medirá la distancia a la que se encuentra nuestro apoyo (lugar donde nos sentaremos) y dependiendo si se encuentra en un rango aceptable para poder sentarnos efectuará la acción.

En este documento iremos describiendo el proceso diseño y desarrollo del proyecto, pasando desde la elección de los componentes hasta el diseño de la ortesis y el porqué de su elección.

CAPÍTULO 2

1. Biomecánica del miembro inferior

Ya que nuestro mecanismo se centra en las articulaciones de rodilla y tobillo, serán esas a las que se analizarán y a las cuales se adaptará el dispositivo para realizar los movimientos más fluidos.

Se establecerán dos posiciones que se debe mantener en periodos prolongados de tiempo, son dos situaciones claves de nuestro proyecto, como son la bipedestación y la sedestación:

- Estado de bipedestación: Esta posición está determinada por una alineación de las articulaciones de cadera-rodilla-tobillo, es una posición en la cual nuestro cuerpo se encuentra en equilibrio y totalmente erguido, en esta posición el centro de gravedad se encuentra elevado y para las personas con movilidad reducida es difícil de mantenerse sin un apoyo externo (muletas).

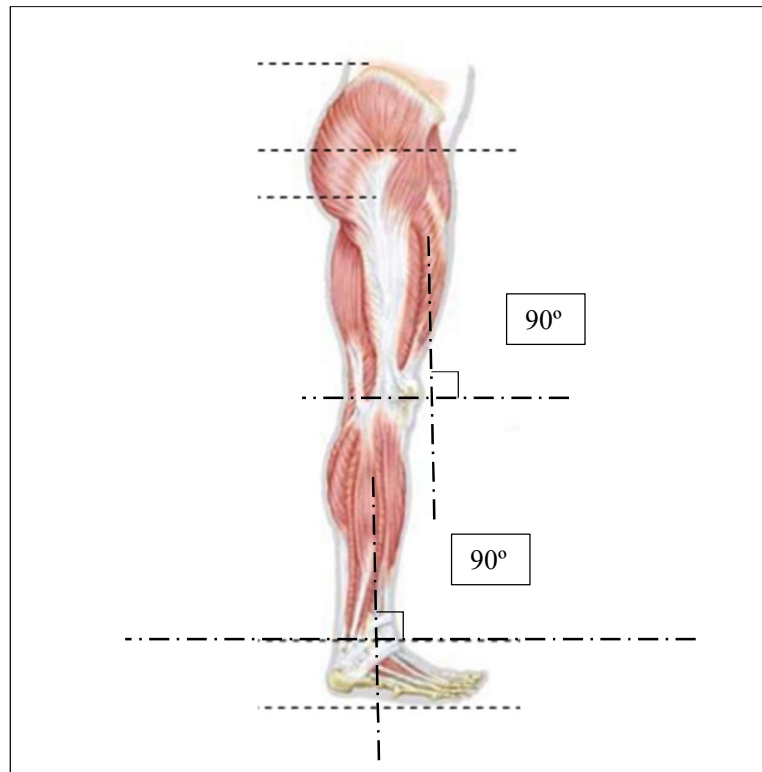


Figura 3: Ángulos bipedestación [6]

- Estado de sedestación: En este estado la articulación de rodilla esta, normalmente, flexionada unos 90° , la articulación de tobillo por su parte se encuentra en su estado normal, es decir, ni en extensión ni en flexión, aunque previamente para llegar a esa posición ha de hacer un movimiento de dorsiflexión. La parte difícil de este estado es llegar a levantarse manteniendo el equilibrio, para ello el centro de gravedad solo debe moverse a través del eje longitudinal, de esa manera conseguiremos una total estabilidad.

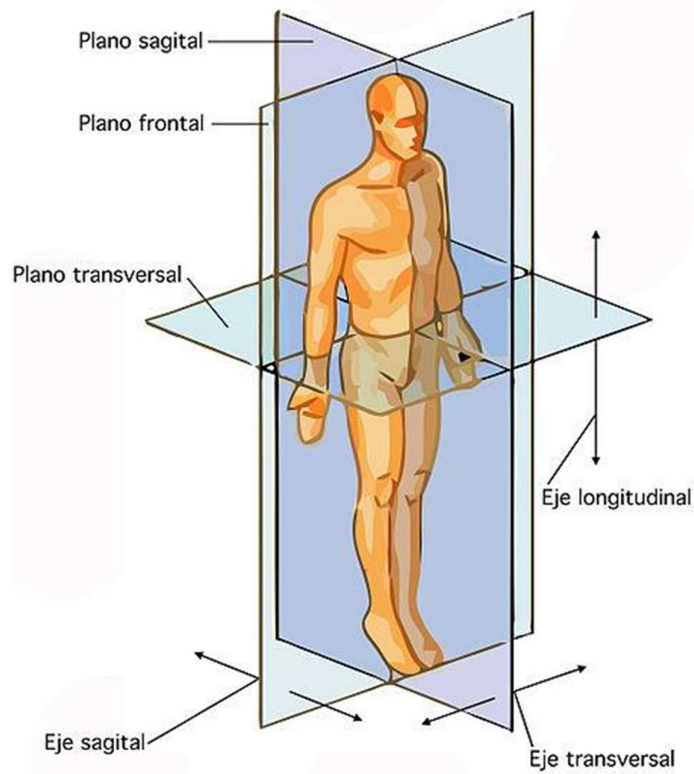


Figura 4: Ejes del cuerpo humano [7]

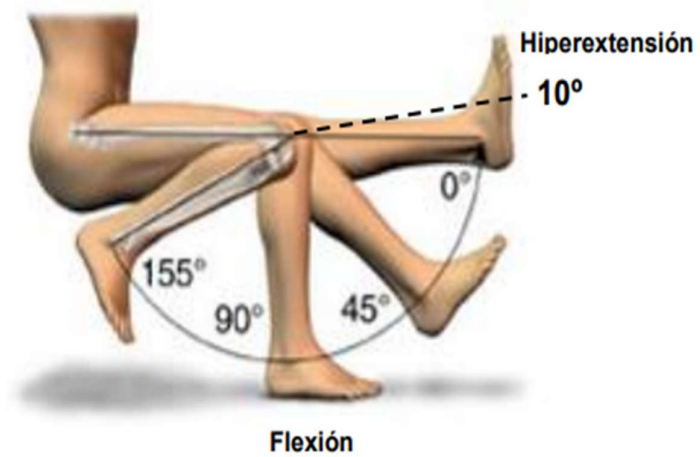


Figura 5: Ángulos rodilla [6]

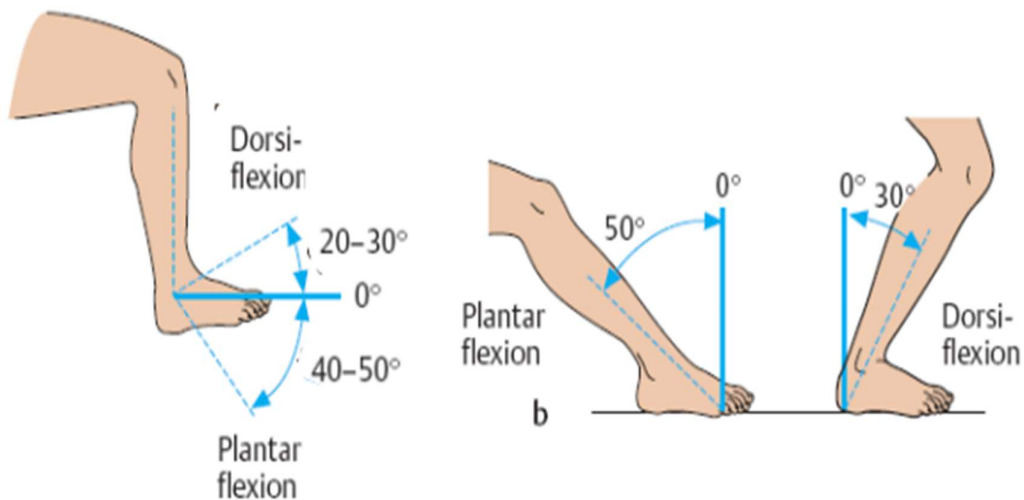


Figura 6:Ángulos tobillo [6]

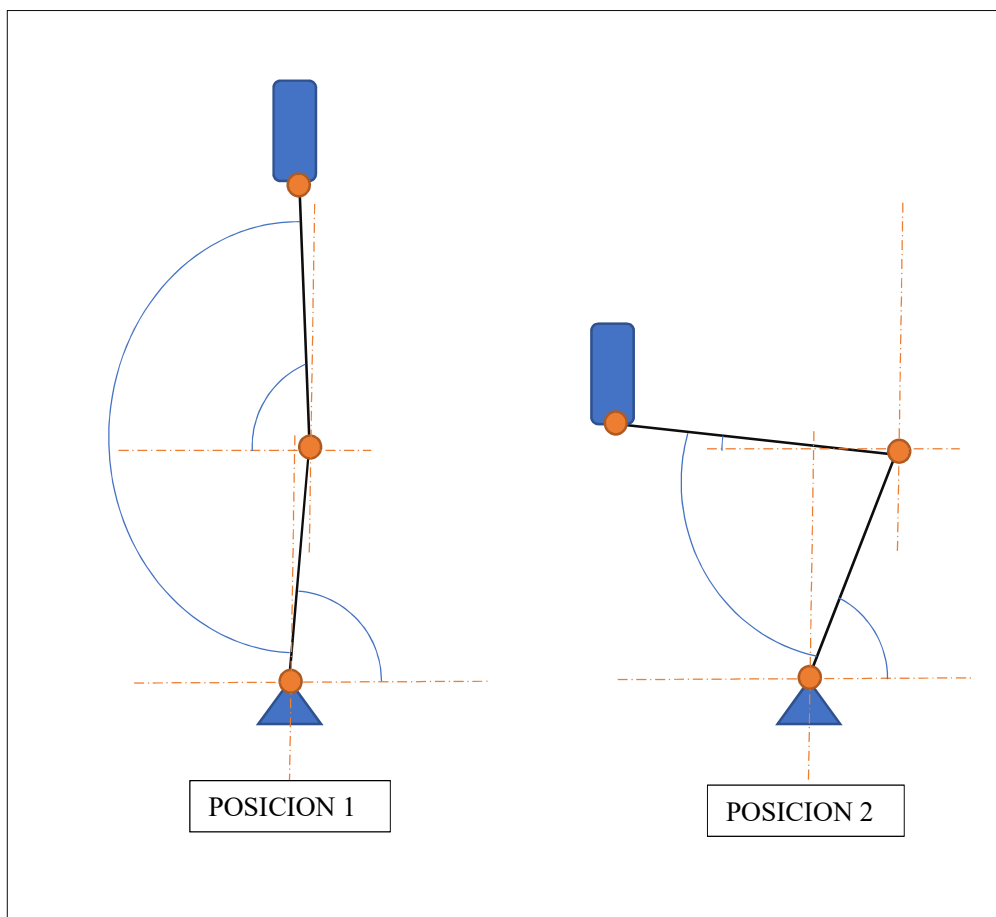


Figura 7:Simulación posiciones extremas de nuestro mecanismo

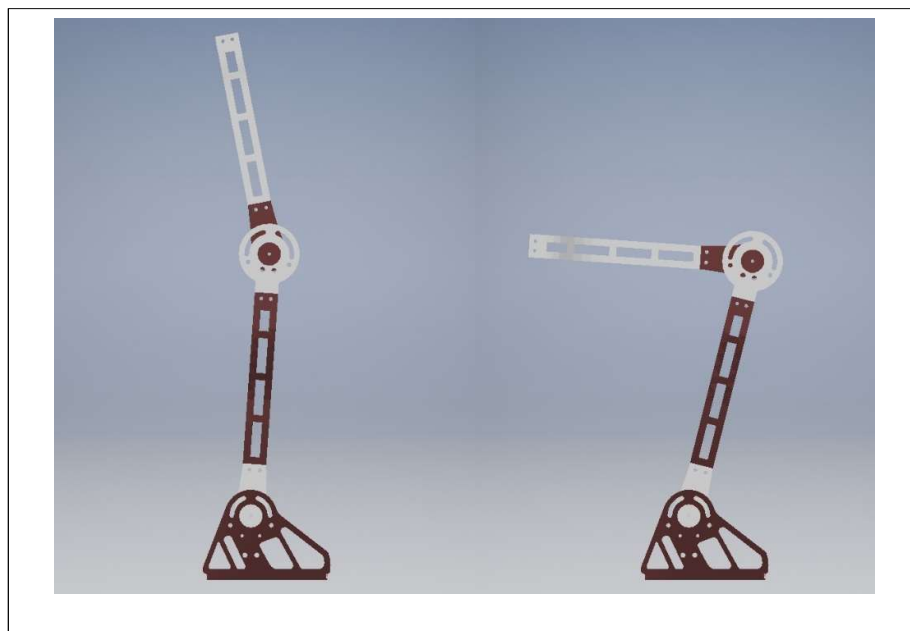


Figura 8: Posiciones extremas de nuestro mecanismo

2. Diseño del prototipo

Hemos partido del sistema KAFO para nuestro diseño del prototipo, pero en vez de hacerlo en formato bitutor como se ve en los productos de ortopedia, hemos preferido hacerlo mediante un solo tutor. La decisión de esto era simplificar el modelo ya que es solo un prototipo, no es una versión final del dispositivo. Las dimensiones están hechas a escala 1:2 y el material que se ha utilizado para las piezas impresas en 3D ha sido PLA (ácido poliláctico) y ABS (acrilonitrilo butadieno estireno).



Figura 9:KAFO Bitutor [8]

El prototipo está compuesto por:

- Tutor Muslo(superior):
Parte final del del dispositivo,
su finalidad es sujeción y soporte.



Figura 10:Tutor Muslo

- Rotor Muslo:
Junto con la parte tutor muslo
forman la articulación de la rodilla.

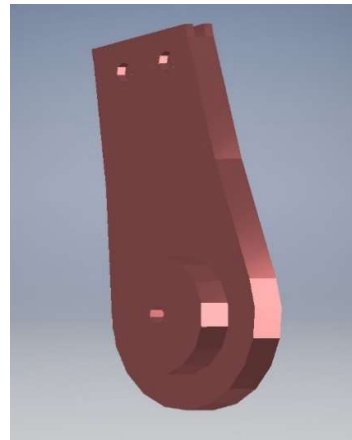


Figura 11:Rotor Muslo

- Articulación de rodilla:
Junto con el rotor muslo forman
la articulación de la rodilla.



Figura 12:Articulación de rodilla

- Tutor gemelo (pantorrilla):
Parte final del tutor del gemelo
su finalidad es sujeción y soporte.

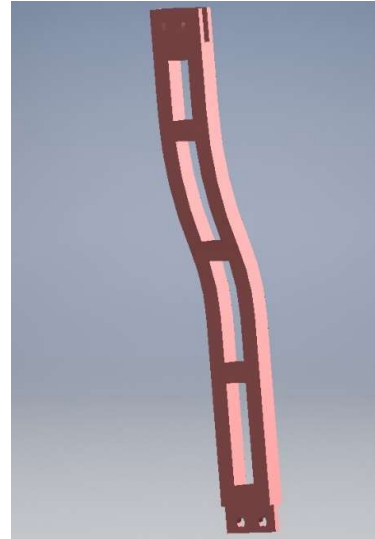


Figura 13:Tutor gemelo

- Rotor gemelo:
Junto con el soporte del pie forman
la articulación del tobillo.

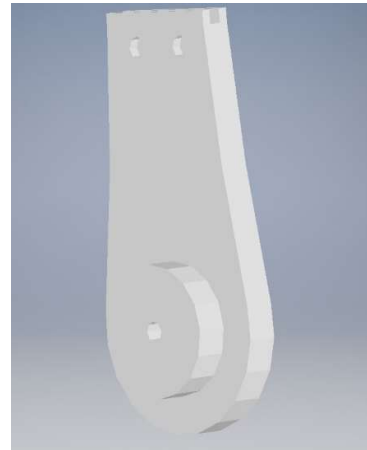


Figura 14:Rotor gemelo

- Soporte del pie:
Junto con el rotor gemelo forman
la articulación del tobillo y además
es el soporte de la estructura.



Figura 15: Soporte del Pie

La estructura completa de nuestro dispositivo quedaría de la siguiente manera, debemos tener en cuenta que no hemos añadido actuadores, sensores, cables ni sistema de anclaje del dispositivo a la pierna del paciente.



Figura 16: Estructura completa dispositivo

3. Sistema de control

El cerebro de nuestro proyecto será la placa STM32F4 Discovery, la cual presenta una gran variedad de periféricos y unas características las cuales la hacen idóneas para este proyecto, entre otras tenemos:

- Está compuesto de un microcontrolador STM32F407VGT6:
 - ✓ Núcleo ARM Cortex M4 de 32 bits a 168 MHz
 - ✓ 1 MB Flash.
 - ✓ 192 KB RAM.
 - ✓ 14 timers, 6 USART, 3 I2C, 3 ADC, 2 DAC,
 - ✓ ...
- ST-LINK V2 es el Programador/depurador interno de la placa (no necesitaremos un dispositivo externo).
- Cuatro leds, botón de Reset y botón de usuario.
- ...

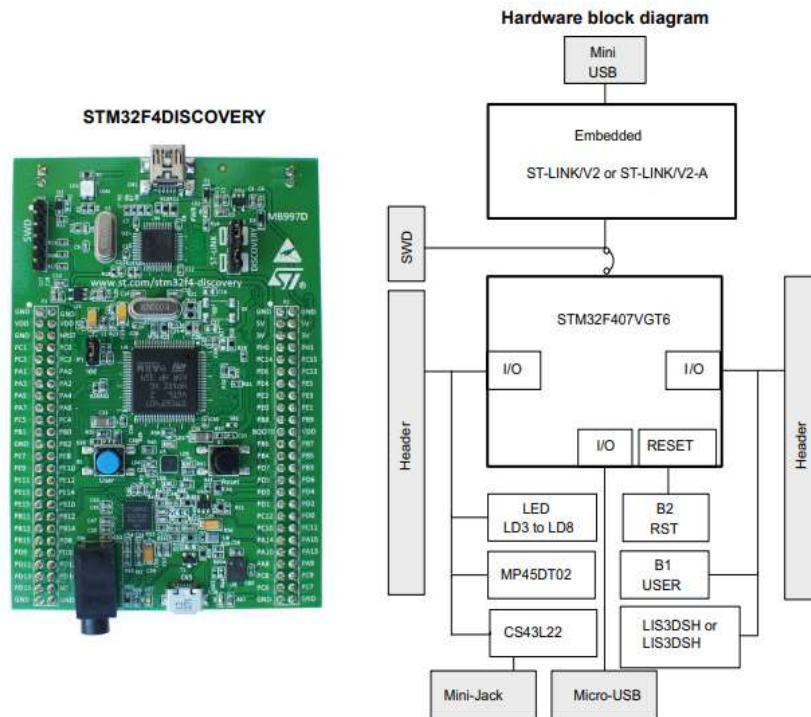


Figura 17:Placa desarrolladora [9]

Los periféricos que se han usado en nuestro proyecto son:

- TIMERS. Se configurarán como:
 - PWM: Se utilizará para controlar los servomotores digitales. Ambos se podrán controlar con un único TIMER ya que disponen de 4 CCR (capture/compare register).
 - INTERRUPCIONES: Será el revisor en nuestro diagrama de flujo, es decir, irá revisando y cambiando el estado en el que se encuentra nuestro programa.
- Protocolo de comunicación USART.
 - RECIBIR: Se configurará para recibir los datos de los sensores, así como el estado del dispositivo.
- Protocolo de comunicación I2C.
 - Se encargará de enviar los datos a la placa desde el sensor MPU6050.
- GPIO's. Se configurarán como:
 - PA0: Se configurará para que inicie tanto la acción de bipedestación como la de sedestación.
 - PA8 y PA10: Serán el TRIGGER y el ECHO en el sensor de ultrasonidos.
 - TRIGGER -> INPUT
 - ECHO -> OUTPUT
 - PD12 y PD13: Se configurarán como salidas PWM.
 - PB6 y PB7: Serán los pines utilizados en la comunicación I2C (SCL y SDA respectivamente).
 - PB10: Pin que se utilizará para la transmisión de datos por la USART.
 - PA8 y PA10: Se configurarán para ser usador por el sensor de proximidad, como TRIGGER y ECHO respectivamente.

Mas adelante se expondrá la configuración de dichos periféricos de una forma más detallada y extensa, además de los archivos del propio programa que se encuentra en el ANEXO I.

4. Actuadores y Sensores

Los actuadores y sensores que hemos elegido para nuestro proyecto han sido aquellos con los que hemos podido realizar las acciones requeridas para cada una de las funciones esenciales del proyecto:

Para la realización de nuestro proyecto hemos necesitado de:

- Servomotores. Se encargará de hacer el movimiento tanto en la articulación del tobillo como de la rodilla.

En un principio elegimos motores paso a paso, de 64 pasos por vueltas, precisos, pero nos dimos cuenta que tenían un par bajo y no sería suficiente para cuando el mecanismo estuviera completo (dispositivo + motores + pierna impresa 3D), por lo tanto, lo descartamos.



Figura 18: Motor paso a paso

Después de ello nos planteamos servomotores, ya que suelen ser muy compactos y con un par alto. Nos decantamos por el Digital Mg996r. Dichos motores disponen de unas características correctas para lo que estamos buscando, bajo peso, gran par y un ángulo de giro de 180°.



Figura 19: Servomotor digital

Características:

- Peso: 55 g
- Dimensión: 40.7 x 19.7 x 42.9 mm aprox.
- Par de parada: 9,4 kgf · cm (4,8 V), 11 kgf · cm (6 V)
- Velocidad de funcionamiento: 0.17 s / 60° (4.8 V), 0.14 s / 60° (6 V)
- Voltaje de funcionamiento: 4.8 V a 7.2 V
- Corriente de funcionamiento 500 mA - 900 mA (6V)
- Corriente de bloqueo 2.5 A (6V)
- Ancho de banda muerta: 5 μ s
- Diseño de doble rodamiento de bolas estable y a prueba de golpes
- Rango de temperatura: 0 °C –4.8 V a 7.2 V– 900 mA (6V) diseño de doble rodamiento de bolas 55 °C...

Figura 20: Características Servomotor

- Acelerómetro. En un principio estuvimos probando con el IMU 5 Degrees of Freedom IDG500 / ADXL335, el cual dispone de un acelerómetro ADXL335, debido a problemas con la inestabilidad de los valores decidimos de elegir otro acelerómetro.

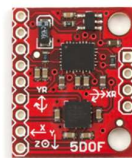


Figura 21: IMU 5DOF

Y para ellos se eligió la MPU6050 (2x), que contiene un acelerómetro MEMS, con una resolución de 16 bits, además de

la capacidad de poder utilizar una única comunicación I2C para dos dispositivos.

Características:

- El acelerómetro MEMS de triple eje en MPU-6050 incluye una amplia gama de características:
- Acelerómetro de triple eje de salida digital con rango programable de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$
- Corriente normal de funcionamiento del acelerómetro: $500 \mu A$
- Corriente en modo acelerómetro de baja potencia: $10\mu A$ a $1.25Hz$, $20\mu A$ a $5Hz$, $60\mu A$ a $20Hz$, $110\mu A$ a $40Hz$
- Detección de orientación y señalización.

Figura 22: Características MPU 6050

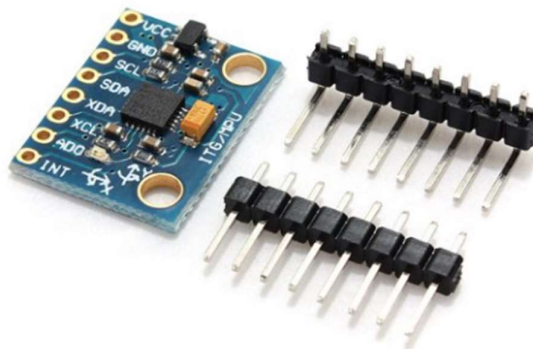


Figura 23: MPU 6050

- Sensor de ultrasonido. Hemos elegido el sensor RCW-0001, es un sensor con las mismas prestaciones que el HC-SR04, pero con unas dimensiones más pequeñas. Este te permite medir distancias con un margen de error muy pequeño. Elegimos este y no otro debido a que queríamos que fuese pequeño y que su rango de medida fuese lo más cercano a 0 cm , en sus especificaciones dice que su distancia de detección es de $1 \text{ cm} - 450 \text{ cm}$.

Características:

- Fuente de alimentación: + 5V DC
- Corriente de reposo: <2mA
- Corriente de trabajo: 15mA
- Ángulo efectivo: <15 °
- Distancia de rango: 2 cm - 400 cm
- Resolución: 0.3 cm
- Ángulo de medición: 30 grados
- Ancho de pulso de entrada de disparo: 10uS

Figura 24: Características RCW-0001



Figura 25:RCW-0001

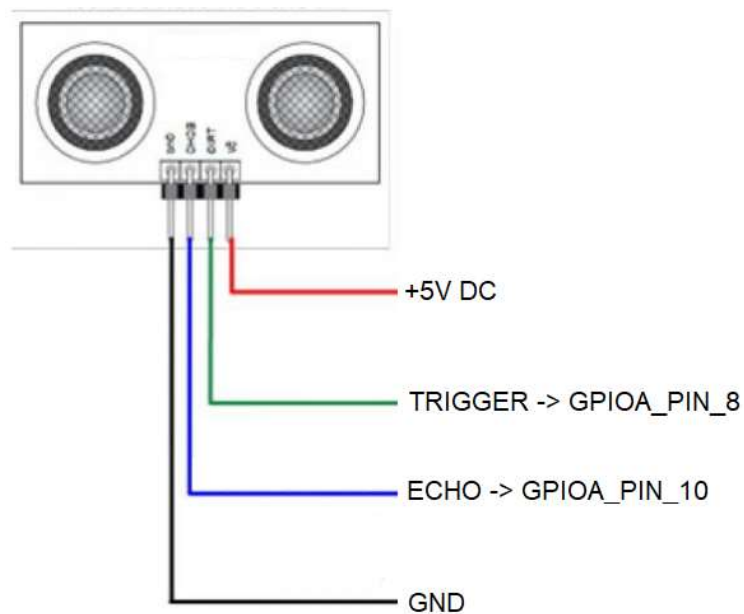


Figura 26: Sensor de ultrasonidos

5. Alimentación del sistema y otros

Nuestro sistema se alimenta a través de nuestro ordenador mediante una conexión USB, aunque se puede alimentar a través una fuente de alimentación externa siempre que sea de 5V. A su vez, nuestra placa alimenta varios sensores, dos MPU 6050 y un sensor de proximidad RCW-0001.

Los motores son alimonados mediante una fuente de alimentación externa tipo A.E.G. de 8,4V y un amperaje de 1100 mAh.



Figura 27: Batería externa

Para regular la tensión que les llega a los motores, hemos decidido escoger el módulo regulador de voltaje LM317, por su tamaño y su precio, gracias a sus perforaciones en la placa podemos añadirlo a nuestro conjunto y así tener todo recogido en un único PCB.

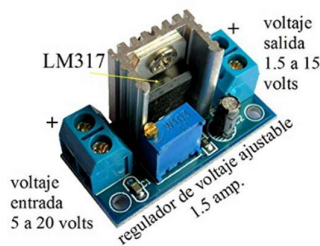


Figura 28: Módulo regulador de voltaje

Por último, tenemos el dispositivo por el cual se transmitirán y recibirán los datos. Es un módulo convertidor de USB a TTL

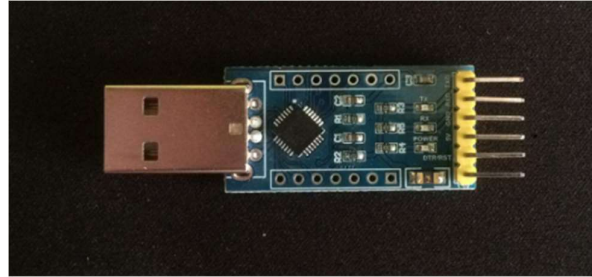


Figura 29: Módulo convertidor de USB a TTL

Características:

Pin	Type	Function
DCD	In	"Data carrier detect" control input (active low)
DTR	Out	"Data terminal ready" control output (active low) (often used with DSR)
DSR	In	"Data set ready" control input (active low) (often used with DTR)
RTS	Out	"Ready to send" control output (often used with CTS)
CTS	In	"Clear to send" control input (often used with RTS)
GND	Power	Ground
TX	Out	Asynchronous serial data transmit (idle high)
V _{BUS}	Power	USB bus voltage (+5V)
RX	In	Asynchronous serial data receive
SUSPEND	Out	Driven high when in USB suspend state
$\overline{\text{SUSPEND}}$	Out	Driven low when in USB suspend state
$\overline{\text{RST}}$	In	Device reset
RI	In	"Ring indicator" control input (active low)
V _{DD}	Power	3.3 V voltage regulator output

Figura 30: Características Módulo convertidor USB a TTL [10]

CAPÍTULO 3

1. Análisis del Sistema de Control

Como se ha visto para controlar nuestro sistema vamos a utilizar diferentes periféricos de nuestra placa, ahora veremos cómo los hemos configurados y el porqué de dicha configuración.

1.1.RCC (Clock control register)

Lo primero que debemos de tener es un reloj de sistema preciso, nuestro microcontrolador dispone de varias fuentes de reloj para su selección y configuración:

- El reloj interno de alta velocidad (HSI)
- El reloj externo de alta velocidad (HSE)
- El reloj de fase bloqueada (PLL).

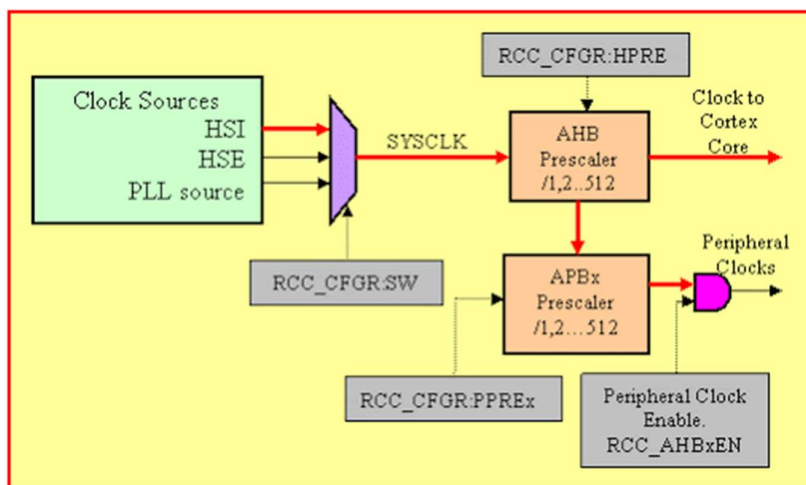


Figura 31:Esquema básico reloj sistema [11]

Con el registro RCC se puede habilitar y configurar la fuente del reloj que deseamos utilizar. La velocidad del microcontrolador viene dada por el valor PLL, que viene por defecto a 16, esto es, la gama STM32F4xx viene con un oscilador RC de 16MHz, se puede usar, pero no es preciso y suele dar fallos en comunicaciones, tales como USART o en periféricos como TIMER's. Se cambiará por uno más preciso:

- En el archivo stm32F4xx.h:

```
#define HSE_VALUE __((uint32_t)8000000) /*!< Value of the External oscillator in Hz */
```
- En el archivo system_stm32F4xx.c:

```
/* PLL_VCO = (HSE_VALUE or HSI_VALUE / PLL_M) * PLL_N */
#define PLL_M 8
```

Figura 32: Configuración Reloj del sistema

Con esta configuración se consigue una mayor precisión y se evita posibles problemas de comunicación entre nuestra placa y los diferentes periféricos.

1.2.SysTick

El SysTick es un temporizador que encontramos en el núcleo de ARM (STM32F407VG). Para nuestro caso lo utilizaremos para configurar un DELAY, una de las funciones más importantes para un proyecto de este tipo, es esencial tener un DELAY que sea preciso y con este se consigue uno de hasta 1 microsegundo.

La configuración del DELAY será para que se produzca una interrupción cada 1 microsegundo o un milisegundo. Para ello debemos de dividir la frecuencia del reloj del sistema:

- * Frecuencia del sistema / 1000 ----->1 ms *
- * Frecuencia del sistema / 100000 ----->10us *
- * Frecuencia del sistema / 1000000 ----->1us *

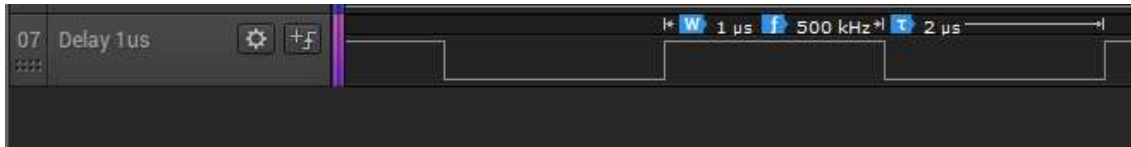


Figura 33:Delay 1us

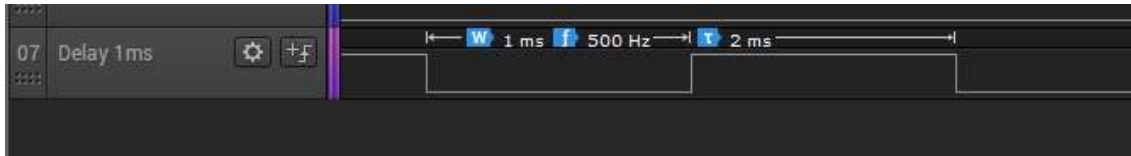


Figura 34:Delay 1ms

Como se muestra en ambas imágenes, se obtiene un DELAY exacto, es cierto que según vaya ejecutándose diferentes rutinas y subrutinas en nuestro programa se irá creando un retardo que sumando el conjunto de todo el programa puede que afecte a estas funciones, sobre todo en el caso de 1ms, pero se ha comprobado que no es un problema para la aplicación que vamos a ejecutar, pero es algo de lo que se debe tener en cuenta para futuras modificaciones.

1.3.TIMER's

Se configurarán dos TIMER's, uno para el control de los dos servomotores, como un PWM donde se utilizarán dos canales y el otro para control del programa y su estado en su modo de interrupción. Para ambos casos hay que prestar especial atención a dos parámetros:

El period (periodo): especifica el valor del período que se cargará en el registro activo de recarga automática en el próximo evento de actualización. Este parámetro debe ser un número entre 0x0000 y 0xFFFF (16 bits). "TIM_TimeBaseStructure.TIM_Period".

El prescaler (preescalador) Especifica el valor del preescalador utilizado para dividir el reloj TIM. Este parámetro puede ser un número entre 0x0000 y 0xFFFF (16 bits). "TIM_TimeBaseStructure.TIM_Prescaler".

$$F_{out} = \frac{Timer_Clock}{(Period - 1) * (Prescaler - 1)}$$

F_{out} = frecuencia requerida en su salida.

$Timer_Clock$ = Reloj del TIMER que se va a utilizar.

$Period$ = "TIM_TimeBaseStructure.TIM_Period".

$Prescaler$ = "TIM_TimeBaseStructure.TIM_Prescaler".

Como se comentó anteriormente la placa STM32F4DISCOVERY dispone de 14 TIMER's, y están organizados de la siguiente manera:

- TIM1 - TIM8 y TIM2 a TIM5 y TIM9 a TIM14. Son muy parecidos entre ellos.
- TIM6 y TIM7. Son los que presentan más limitados.

Se elegirá a los del primer grupo, están conectados al bus APB1 que tiene una frecuencia de 84 MHz.

1.3.1. PWM

Se elegirá a los del primer grupo, están conectados al bus APB1 que tiene una frecuencia de 84 MHz. Estos motores funcionan con una señal PWM de 50Hz por lo tanto esa será la frecuencia de salida. Aplicando la fórmula:

$$Period = 1999$$

$$Prescaler = 84$$

Se utilizará el TIMER's 4, con los pines PD12 Y PD13.

PD12	FSMC_A17/ TIM4_CH1/ USART3_RTS	59	-	-	-	-	GREEN	-	-	-	-	-	-	-	44	-
PD13	FSMC_A18/ TIM4_CH2	60	-	-	-	-	ORANGE	-	-	-	-	-	-	-	45	-

Figura 35:Pines PWM

```
TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1; // Se divide
entre 1
TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
TIM_TimeBaseInit(TIM4,&TIM_TimeBaseStructure);
TIM_Cmd(TIM4,ENABLE);
TIM_OCInitStructure.TIM_OCpolarity=TIM_OCpolarity_High; //Se active
en alto
TIM_OCInitStructure.TIM_Pulse=500; //CCR1_Val; valor inicial
TIM_OC1Init(TIM4,&TIM_OCInitStructure);
```

Figura 36: Configuración común Timer

Y así quedaría la configuración de un canal con un PWM de 50 Hz.

1.3.2. INTERRUPCION

Para la configuración de un TIMER en modo interrupción, hay que añadir una parte más al código anterior, lo único que se cambiaría de la primera parte será la frecuencia o si prefiere la inversa, es decir, el tiempo con el que el TIMER se desborda y se produce la interrupción. El código nuevo sería:

```
TIM_ITConfig (TIM2, TIM_IT_Update, ENABLE);

NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn; //Habilitamos el IRON
del TIM2
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; //Prioridad 0
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0; //Subprioridad 0
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //Habilitamos
NVIC_Init (&NVIC_InitStructure); //Hacemos efectiva la configuración

TIM_Cmd (TIM2, ENABLE); //Habilitamos
```

Figura 37: Configuración Interrupción Timer

Como no va a ver otra interrupción le daremos tanto prioridad 0 como sub-prioridad 0, estos datos son relevantes cuando hay más de una interrupción y funcionan por grupos.

priority group	pre-emption priority		sub priority	
	bit	number	bit	number
0	0	1	4	16
1	1	2	3	8
2	2	4	2	4
3	3	8	1	2
4	4	16	0	1

Figura 38: Prioridad de las interrupciones

1.4.PROTOCOLO DE COMUNICACIÓN I2C

```
// configuramos I2C1
I2C_InitStruct.I2C_ClockSpeed = 100000;           // 100kHz
I2C_InitStruct.I2C_Mode = I2C_Mode_I2C;          // modo I2C
I2C_InitStruct.I2C_DutyCycle = I2C_DutyCycle_2;  // 50% duty cycle --> el
standard
I2C_InitStruct.I2C_OwnAddress1 = 0x00;           // dirección
I2C_InitStruct.I2C_Ack = I2C_Ack_Disable;        // desactivamos el ack
cuando lee
I2C_InitStruct.I2C_AcknowledgedAddress = I2C_AcknowledgedAddress_7bit; //
dirección de 7 bit
I2C_Init(I2C1, &I2C_InitStruct);                 // inicializamos I2C1

// Habilitamos I2C1
I2C_Cmd(I2C1, ENABLE);
```

Figura 39: Configuración comunicación I2C

1.5.PROTOCOLO DE COMUNICACIÓN USART

```
Configuracion_USART.USART_BaudRate=115200;// número de unidades de señal
por segundo
Configuracion_USART.USART_HardwareFlowControl=USART_HardwareFlowControl_None;
Configuracion_USART.USART_Mode=USART_Mode_Tx;           //Solo
transmitimos
Configuracion_USART.USART_Parity=USART_Parity_No;       //No
paridad
Configuracion_USART.USART_StopBits=USART_StopBits_1;   //1 bit de
parada
Configuracion_USART.USART_WordLength=USART_WordLength_8b; //8 bits de
longitud
USART_Init(USART3,&Configuracion_USART);

USART_Cmd(USART3,ENABLE);
//Habilitamos USART
```

Figura 40: Configuración comunicación USART

1.6.GPIO's

Para la configuración de los GPIO's es siempre la misma solo cambiando ciertos parámetros para cada función. Esta configuración que se presenta a continuación puede valer para cuando su función es de salida de un TIMER, USART, I2C, ...

```

Configuracion GPIO.GPIO Mode=GPIO Mode AF;//Modo funcion alternativa
Configuracion GPIO.GPIO Pin=GPIO_Pin_10;//Pin q se va a utilizar
Configuracion GPIO.GPIO PuPd=GPIO PuPd UP;//habilitamos resistencias pull
up
Configuracion GPIO.GPIO Speed=GPIO_Speed_100MHz;//velocidad

GPIO_Init(GPIOB,&Configuracion GPIO);//Inicializa la configuracion del
puerto

GPIO_PinAFConfig(GPIOB,GPIO_PinSource10,GPIO_AF_USART3);//Escogemos que
funcion alternativa va a realizar

```

Figura 41: Configuración GPIO's

Tipos de función de los GPIO's:

```

typedef enum
{
    GPIO Mode IN    = 0x00, /*!< GPIO Input Mode */
    GPIO Mode OUT   = 0x01, /*!< GPIO Output Mode */
    GPIO Mode AF    = 0x02, /*!< GPIO Alternate function Mode */
    GPIO Mode AN    = 0x03 /*!< GPIO Analog Mode */
}GPIO Mode TypeDef;

```

Figura 42: GPIO's modos de funcionamiento

2. Entorno de Programación y otros programas

2.1. System workbench (IDE)

El entorno de desarrollo System Workbench, llamado SW4STM32, es un entorno de desarrollo/programación de software de múltiples sistemas operativos, es gratuito basado en Eclipse, que admite la gama completa de microcontroladores STM32 y placas asociadas.

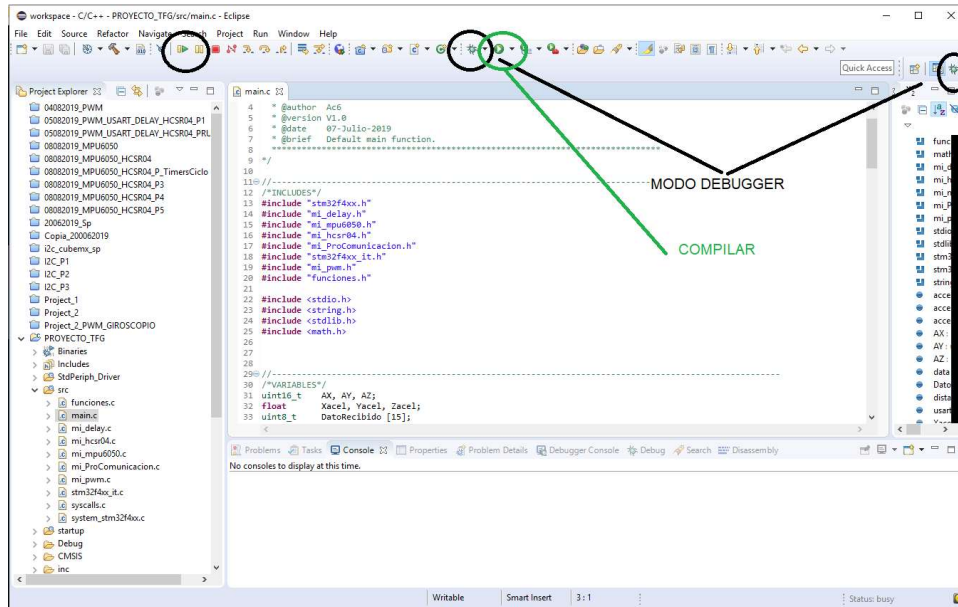


Figura 43: IDE ECLIPSE [12]

SW4STM32 se puede descargar de www.openstm32.org, donde además disponen de foros, blogs y soporte técnico. Se debe registrar para poder descargar el ejecutable, una vez realizado el registro, los usuarios recibirán instrucciones de la instalación a través del directorio Documentation > System Workbench. Las características que presenta el entorno son:

- Soporte integral para microcontroladores STM32, placas STM32 Nucleo, kits de descubrimiento y placas de evaluación, así como firmware STM32 (biblioteca periférica estándar o STM32Cube HAL)
- Compilador GCC C / C ++
- Depurador basado en GDB
- Eclipse IDE con gestión de trabajo en equipo

- Compatible con complementos de Eclipse
- Soporte ST-LINK
- Sin límite de tamaño de código
- Compatibilidad con múltiples sistemas operativos: Windows[®], Linux y OS X[®] [13]



Para finalizar se debe de indicar una serie de pasos de forma gráfica de cómo crear un proyecto, el último paso es el más importante ya que es donde se decidirá que firmware es con el que se va a trabajar. Se ha decidido por el “STM32 Standard Peripheral Library Expansion” [14] ya que se encuentra en una posición consolidada en el mundo laboral, aunque las librerías HAL (nueva versión) poco a poco se está estandarizando, pero aún es pronto para ello.



Figura 44: Stdperiph

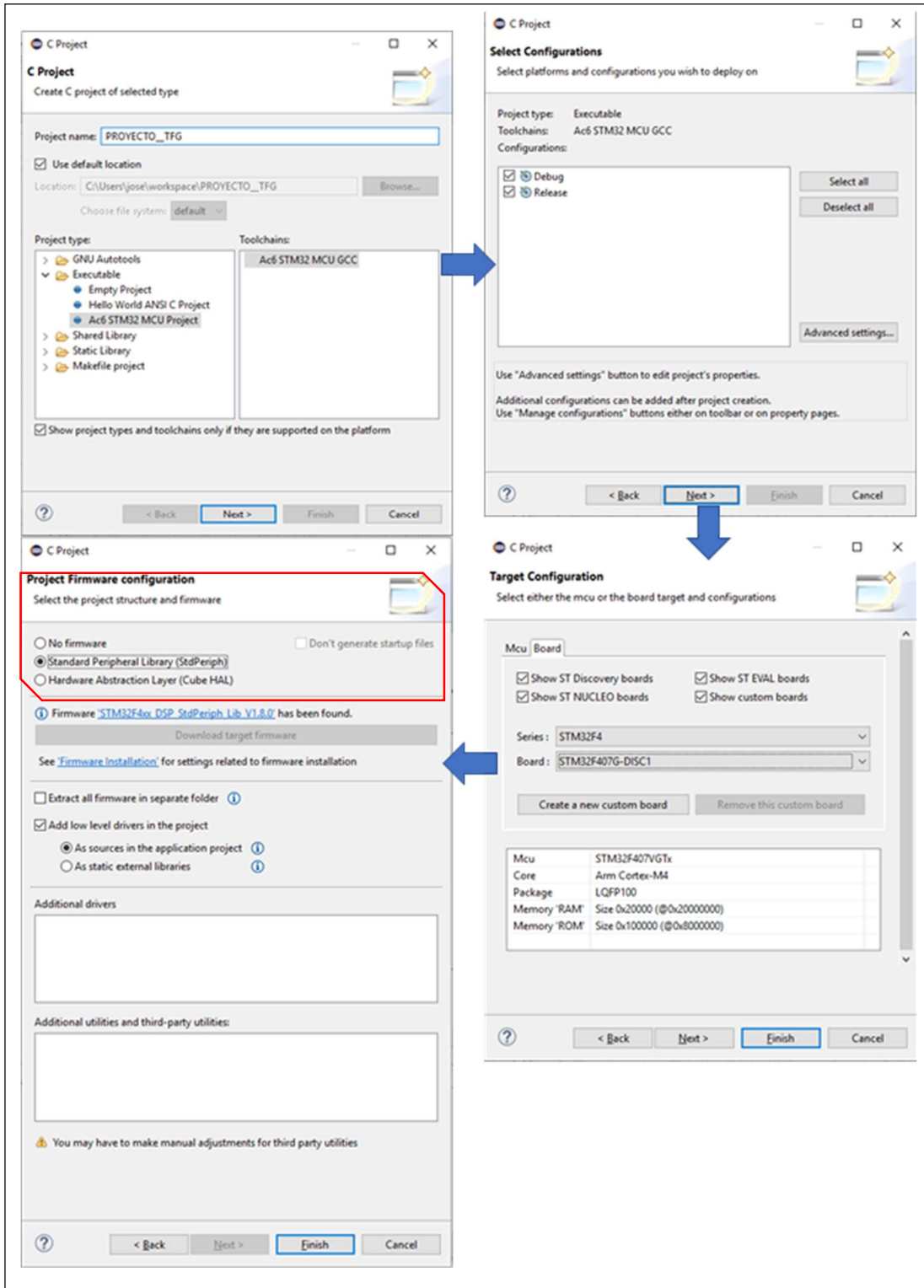


Figura 45:Pasos para crear un proyecto

2.2.Logic Analyzer SALEAE (IDE)

El analizador lógico SALEAE es un software muy potente, con la capacidad de poder capturar hasta 7 señales diferentes (de un circuito digital). La información se muestra en una pantalla en forma de diagrama temporal. Se puede utilizar para decodificar protocolos de comunicación, tales como: UART, SPI, I2C, 1-Wire, I2S, CAN, USB, ... [15]

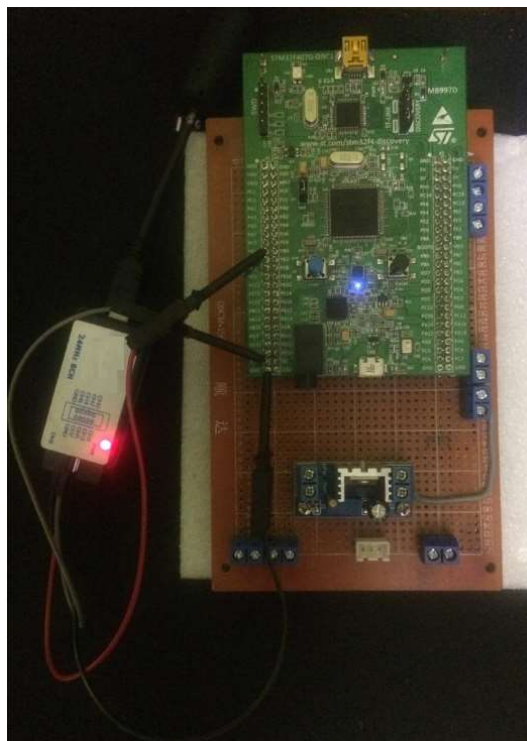


Figura 46:Analizador lógico

El procedimiento es muy simple, se conecta el analizador al PC mediante su conexión USB 2.0 (Mini-B 5 Pin) y al abrir el programa ya reconoce al dispositivo que se hubiese conectado. Se puede configurar la velocidad (Sample Rate) y la duración del muestreo de la señal.

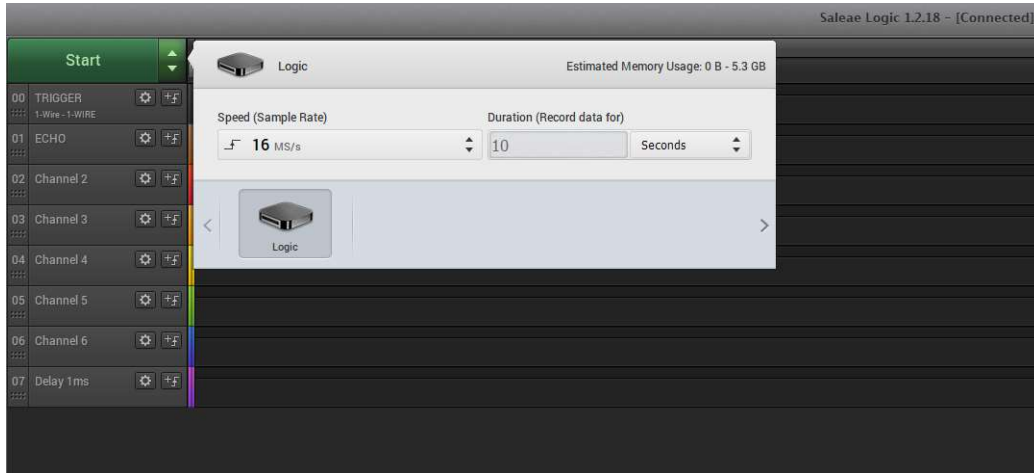


Figura 47: Programa Logic Analyzer

2.3.Tera Term

Tera Term es un emulador de terminal de software libre (programa de comunicación) que admite:

- Conexiones de puerto serie.
- Conexiones TCP / IP (telnet, SSH1, SSH2).
- Registro de reproducción.
- Comunicación IPv6.
- Scripts que usan el "Lenguaje Tera Term".

Figura 48: Características Tera Term

Su configuración es muy sencilla, primero se conecta al PC.

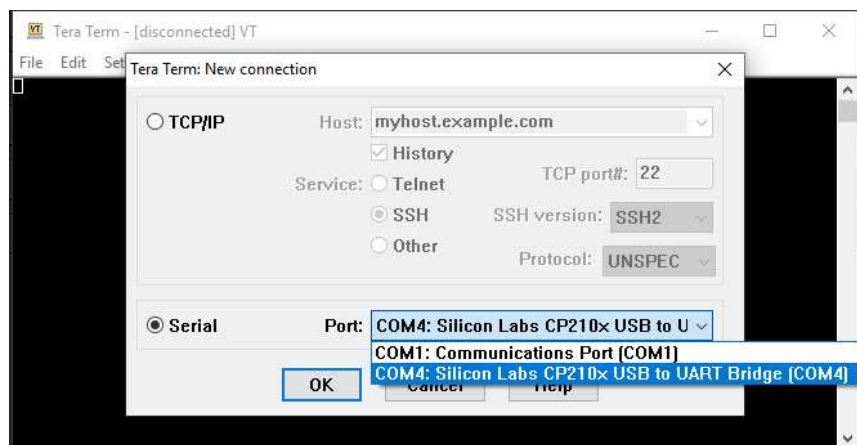


Figura 49: Tera term configuración 1

Y a continuación en la pestaña “Setup” se selecciona “Serial port setup” y ahí se configura con los mismos parámetros como se configuró la USART.

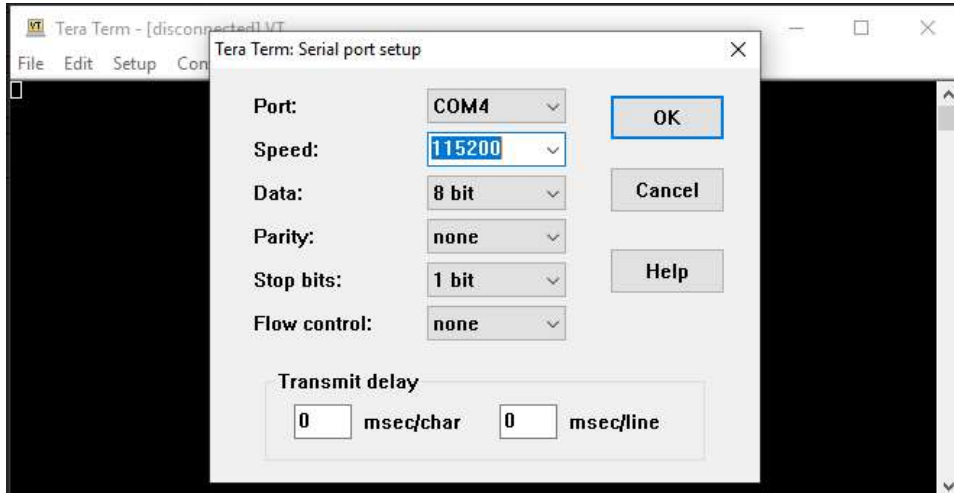


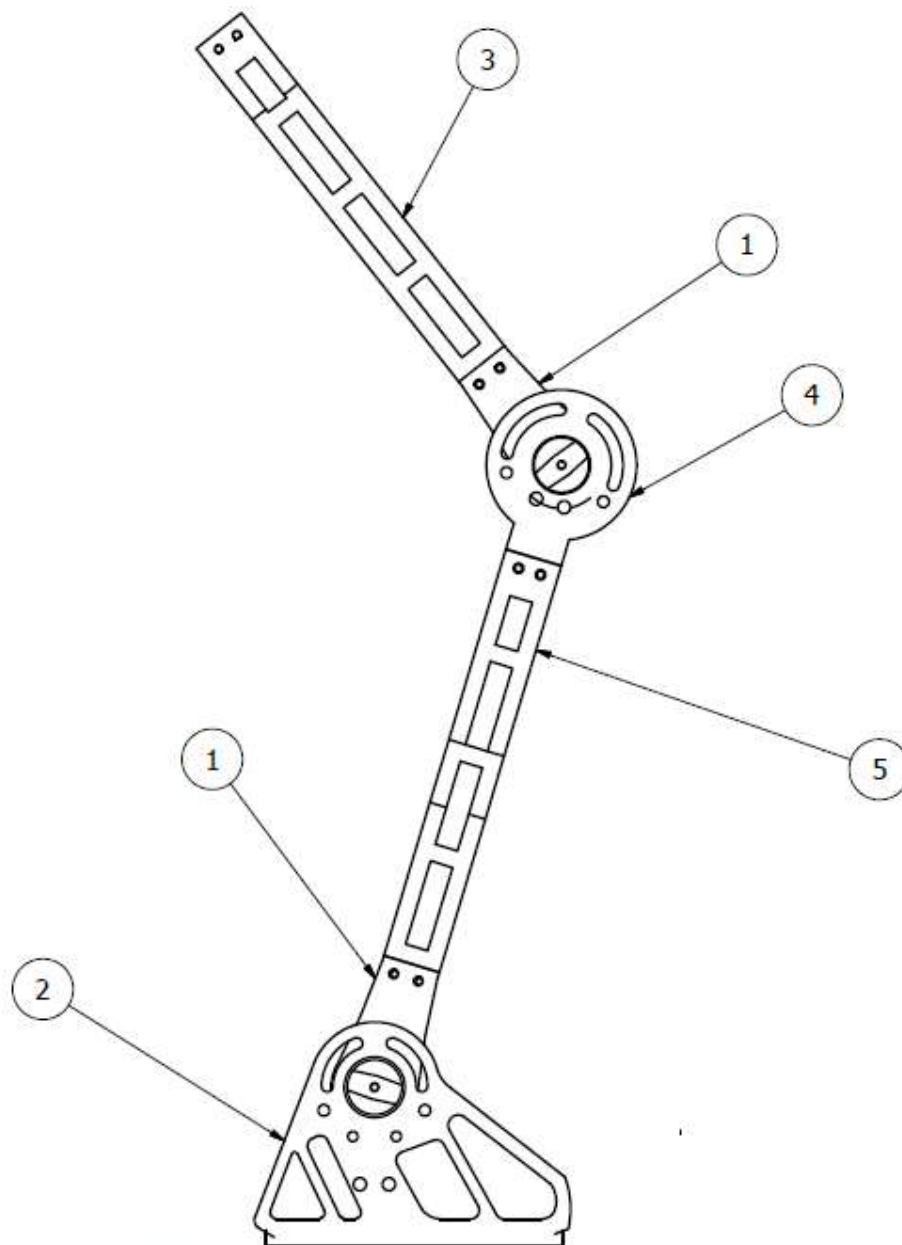
Figura 50: Tera Term configuración 2

2.4.AUTODESK INVENTOR

Autodesk Inventor es un sistema de diseño mecánico y modelado en 3D, que permite el diseño de piezas complejas y su ensamblado, así como simulación dinámica, análisis de tensión, visualización y renderización. De hecho, estas imágenes, están renderizadas, aunque debido a Word no se aprecie la calidad en ellas.



Figura 51: Dispositivo diseñador en INVENTOR



ELEMENTO	CTDAD	Nº DE PIEZA
1	2	Rotor
2	1	Pie
3	1	Tutor muslo
4	1	Final tutor Rodilla
5	1	Tutor gemelo

Figura 52: Inventor pieza completa

CAPÍTULO 4

1. Planificación

Al tratarse un proyecto experimental y al ser la primera versión, la planificación será evaluada por cumplir objetivos y de no ponerse fechas límites:

- Análisis del aparato de marcha
- Diseño del dispositivo
- Elección de componentes
- Ubicación de los componentes en el diseño.
- Elección de la placa desarrolladora.
- Elección del IDE y su firmware.
- Instruirse en la programación del sistema empotrado.
- Elección de los periféricos que se van a usar de la placa
- Programación de los periféricos.
- Conexión superficial de los componentes a la placa
- Implementación del código principal.
- Test de pruebas.

- Conexionado fijo final de los componentes y placa.
- Test finales.

2. Puesta en marcha

Primero se presenta un diagrama de flujo básico del programa principal:

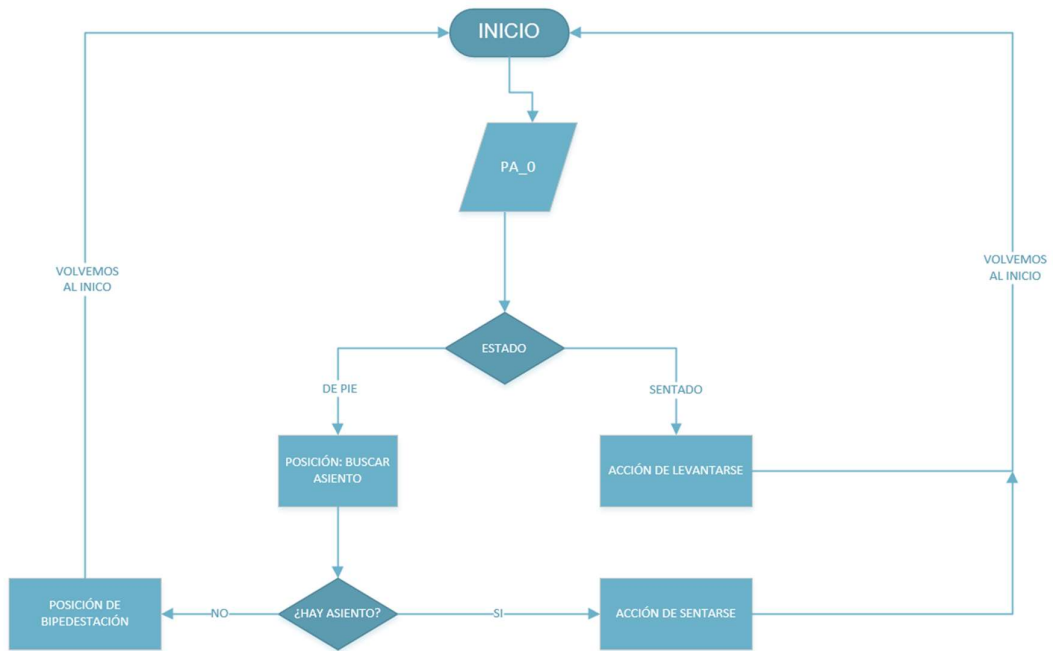


Figura 53: Diagrama de flujo. Programa principal

Y a continuación, la acción de sentarse:



Figura 54: Posición 1 (Inicio).



Figura 55: Posición 2.



Figura 56: Posición 3.



Figura 57. Posición 4.



Figura 58:Posición 5(final).

3. Conclusiones

La idea de realizar un proyecto como este es propia del exponente, y de ahí seguramente algunos aspectos negativos en los resultados que se han obtenido durante el proyecto. Desde el comienzo se vio que el proyecto era más amplio que un trabajo de fin de grado al uso. Y esto es debido a que en el proyecto se han realizado tareas de diferentes ámbitos de la ingeniería y no solo eso, también al análisis de dos funciones básicas pero complejas de la biomecánica del miembro inferior, como son la bipedestación y la sedestación.

Al tocar dos ramas tan diferentes de la ingeniería como son el diseño 3D y mecanismos, por un lado y sistemas empotrados por otro, se ve la dificultad del proyecto, al ser novedoso la línea de planificación no será tan clara. Es cierto que el proyecto ha llegado a su objetivo, realizar ambas acciones, pero como se verá más adelante ha habido una serie de problemas y dificultades que se deberán pulir en el futuro.

Pero empecemos por los retos, uno de ellos fue la placa desarrolladora, la STM32F4DISCOVERY, elegir esta placa no fue un capricho, sino una oportunidad para poder evolucionar como ingeniero y adquirir los conocimientos necesarios para poder aplicarlo al ámbito laboral, ya que los productos de la empresa STMICROELECTRONICS junto con el núcleo que ARM son utilizados en gran número de empresas, así que su adquisición de conocimientos me brindaba la oportunidad de añadir a mi curriculum un aspecto que me abriría algunas oportunidades.

Otro de los retos fue el diseñar el dispositivo completamente desde cero, es cierto que partimos de la base de algunos modelos establecidos, el objetivo no era rediseñar las órtesis, sino adaptar el modelo más conveniente a nuestro proyecto. El software utilizado fue AUTODES INVENTOR, había otras opciones (CATIA, SOLIDWORKS), pero este nos brindaba la opción de tener durante dos años una licencia gratuita y además de ser muy intuitivo, por lo que la evolución en él ha sido muy rápida.

El último de los retos tiene que ver con los mecanismos y como adaptarlo al proyecto que se debe realizar. Al no tener experiencia en este campo, ha sido uno de los desafíos más complicados que se ha tenido que resolver durante la elaboración de este proyecto. El principal escollo fue adaptar un mecanismo monotutor en donde todo el peso de los componentes (sensores y actuadores) está situado en un único lado, eso y el material utilizado (ABS para las piezas impresas), hacen que el sistema sea un poco inestable en ciertos momentos, dificultando la lectura en los diferentes sensores. Además, al no haber precedentes, el método hasta que se consiguió que todo encajase fue de ensayo y error, algo que sin duda nos ralentizó en la elaboración y ejecución.

Una de las partes que no se ha podido abordar fue el sistema de sujeción, estos sistemas son personalizados o fabricados con materiales elásticos y eso se salía completamente de nuestra función. La forma que se ha decidido de fijar ha sido con bridas, con la intención de ser una opción temporal. Además, el utilizar servomotores de uso “domestico” y forzarlo hasta sus límites, nos ha llevado a tener que renovar alguno que otro.

4. Líneas de trabajos futuros

Como se ha comentado en el punto anterior hay una serie de dificultades que se deben de mejorar en futuras versiones.

Realizar el mecanismo mediante un modelo bitutor, eso le dará estabilidad al sistema, además de poder utilizar 4 servomotores para así dar más fuerza/par al dispositivo, con ello conseguiremos que el peso se reparta mejor en toda la superficie del mecanismo.

Los motores en este proyecto han sido suficientes, es cierto que se estaba trabajando con una pierna impresa en 3D a una escala 1:2 y además hueca por dentro para así no añadir más peso de lo necesario, pero es de obligación que se estudió otros motores con mayor par y más fiables.

Mejorar el sistema de detección, ya que uno óptico sería mucho más fiable y con mayor precisión.

Implementar un sistema de sujeción por correas de cuero o un sistema adaptado mixto de velcro y nylon.

CAPÍTULO 5

1. Presupuesto

Se hará un estudio de la viabilidad del proyecto, en él se desarrollará los costes materiales y personales, al ser un proyecto experimental, en el presupuesto no se contemplará el coste del desarrollo sino una estimación de lo que podría costar el producto en sí, tanto materiales como mano de obra.

A continuación, se detalla el presupuesto correspondiente al desarrollo del proyecto.

1.1. Presupuesto Hardware

Tabla 1: Presupuesto Hardware

CANTIDAD	DESCRIPCIÓN	PRECIO	
		PRECIO UNITARIO (€)	PRECIO TOTAL (€)
1	STM32F4 - DISCOVERY	20	20
2	MPU 6050	5	10
2	SERVOMOTOR DIGITAL MG996R	16	32

1	RCW-0001	10	10
1	MÓDULO CONVERTIDOR DE USB A TTL	10	10
1	BATERIA EXTERNA A.E.G	20	20
1	REGULADOR DE VOLTAJE	5	5
1	PROTOTIPO PCB PERFORADO	0,30	0,30
10(m)	CABLE (awg 18)	1	10
10	CONECTORES VARIOS	0,10	1
60 (g)	PIEZAS IMPRESAS ABS	0,50(g)	30
10(g)	ESTAÑO	0,1	1
COSTE TOTAL MATERIALES			149,3 €

1.2. Presupuesto software

Tabla 2: Presupuesto Software

CANTIDAD	DESCRIPCIÓN	PRECIO	
		PRECIO UNITARIO (€)	PRECIO TOTAL (€)
1	SYSTEM WORKBENCH	*1	*1
1	LOGIC ANALYZER	*1	* 1
1	TERA TERM	*1	*1
1	AUTODESK INVENTOR	2.613,60(año)	1.089(*2)
1	PAQUETE OFFICE	10,50 €(mes)	52,5(*2)
COSTE TOTAL SOFTWARE			1.061,4 €

*1 Programa gratuito por lo que el coste es cero para el cliente.

*2 Los cálculos se han realizado en base a la duración del proyecto, 5 meses

1.3. Presupuesto del personal

Se ha determinado el coste del personal, utilizando las tareas del diagrama de GANT y su cantidad en horas. El coste de un ingeniero por horas lo hemos estimado en unos 13€/h con un salario anual de 27.000€ [16]

Tabla 3: Presupuesto del Personal

DESCRIPCIÓN	DURACIÓN (horas)	PRECIO TOTAL (€)
ANÁLISIS DEL APARATO DE MARCHA	24	312
DISEÑO DEL DISPOSITIVO	120	1560
ELECCIÓN DE COMPONENTES	24	312
UBICACIÓN DE LOS COMPONENTES EN EL DISEÑO	16	208
ELECCIÓN DE LA PLACA DESARROLLADORA	16	208
ELECCIÓN DEL IDE Y SU FIRMWARE	16	208
INSTRUIRSE EN LA PROGRAMACIÓN DEL SISTEMA EMPOTRADO	160	2080
ELECCIÓN DE LOS PERIFÉRICOS QUE SE VAN A USAR DE LA PLACA	24	312
PROGRAMACIÓN DE LOS PERIFÉRICOS	100	1300
CONEXIONADO SUPERFICIAL DE LOS COMPONENTES A LA PLACA	16	208
IMPLEMENTACIÓN DEL CÓDIGO PRINCIPAL	120	1560
TEST DE PRUEBAS	40	520
CONEXIONADO FIJO FINAL DE LOS COMPONENTES Y PLACA	24	312
TEST FINALES	40	520
TOTAL		9620 €

1.4. Presupuesto de ejecución del proyecto

Tabla 4: Presupuesto de ejecución del Proyecto

PRESUPUESTO TOTAL	COSTES (€)
Costes del material	149,3 €
Costes del software	1.061,4 €
Costes del personal	9620 €
TOTAL	10830.7
Gastos Generales (13%)	1407.991
Beneficio Industrial (6%)	649.842
TOTAL (SIN IVA)	12888.533
I.V.A (21%)	2706.59193
PRESUPUESTO TOTAL DE EJECUCIÓN POR CONTRATA	15595.12493 €

CAPÍTULO 6

1. Ficha técnica (DATASHEET)

Debido a lo amplio y extensos que son añadiremos la URL de estos documentos para no hacer el documento tan pesado ni tedioso a la hora de leerlo.

- ST DISCOVERY MANUAL DE USUARIO UM1472
“https://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf”
- MANUAL DE REFERENCIA
“https://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf”
- MPU6050 – ESPECIFICACIONES DEL PRODUCTO
“https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf”
- RCW-0001 ->HC SR04 SENSOR ULTRASONIDO
“<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>”
- SERVOMOTOR MG996R
“https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf”

- INTRODUCCIÓN AL IDE ECLIPSE

“[http://dis.um.es/~bmoros/privado/bibliografia/tutorial%20eclipse%20para%20novatos%20java%20\(Pollino\).pdf](http://dis.um.es/~bmoros/privado/bibliografia/tutorial%20eclipse%20para%20novatos%20java%20(Pollino).pdf)”

- GETTING STARTED WITH STM32 MCU DISCOVERY

“https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/d0/f7/45/d7/3b/96/47/b3/DM00285842/files/DM00285842.pdf/jcr:content/translations/en.DM00285842.pdf”

- MANUAL PRACTICO AUTODESK INVENTOR

“http://www.vc.ehu.es/Dtecnico/descargas/Manual_Practico_Inventor.pdf”

Bibliografía

- [1] O. d. ortesis. [En línea]. Available: <https://apps.who.int/iris/bitstream/handle/10665/259508/9789243512488-part1-spa.pdf;jsessionid=86DAD17A7E4054BDDEA49A6E693A2CC7?sequence=1>.
- [2] D. bipedestacion. [En línea]. Available: <https://neurorhb.com/blog-dano-cerebral/la-bipedestacion/>.
- [3] R. M. D. Sist_ empotrado. [En línea]. Available: <http://personales.upv.es/rmartin/elixir/tema1/sisempotrados.html>.
- [4] m. i. microcontrolador. [En línea]. Available: <https://mr5informaticosgj.wordpress.com/primer-departamental/microcontroladores/>.
- [5] a. P. 3D, «<https://www.thingiverse.com>,» [En línea]. Available: <https://www.thingiverse.com/thing:720493>.
- [6] F. H. A. Stengele. [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_s_f/capitulo2.pdf.
- [7] l. e. d. c. humano. [En línea]. Available: <https://www.lifeder.com/planos-anatomicos-ejes/>.
- [8] o. k. bitutor. [En línea]. Available: <https://www.ottobock.es/ortetica/soluciones/aqualine/>.
- [9] s. placa. [En línea]. Available: https://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf.
- [10] E. C. U. a. TTL. [En línea]. Available: <https://www.electronicaplugandplay.com/comunicaciones/product/309-usb-to-serial-ttl-cp2102>.
- [11] i. bipedestacion. [En línea]. Available: <https://www.periodicoelnazareno.es/hospital-san-agustin-ofrece-consejos-saludables-para-disfrutar-la-semana-santa-inout/>.
- [12] e. IDE. [En línea]. Available: <https://www.eclipse.org/>.
- [13] s. S. workbench. [En línea]. Available: https://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-ides/sw4stm32.html.
- [14] s. S. S. P. L. Expansion. [En línea]. Available: https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-standard-peripheral-library-expansion.html.
- [15] S. a. logico. [En línea]. Available: <https://support.saleae.com/getting-started/setup>.
- [16] i. s. ingeniero. [En línea]. Available: <https://www.indeed.es/salaries/Ingeniero/a-industrial-Salaries>.
- [17] J. K. clock. [En línea]. Available: <https://sites.google.com/site/johnkneenmicrocontrollers/clocks/clk407>.

- [18] S. y. A. .. IDE. [En línea]. Available: <http://www.openstm32.org/HomePage>.
- [19] T. M. stm32f4. [En línea]. Available: <https://stm32f4-discovery.net/2015/01/properly-set-clock-speed-stm32f4xx-devices/>.

Anexo I

1. Código del Proyecto

Carpeta contenedora del proyecto integro.

Anexo II

1. Diagrama de Gant y Calendario

Anexo III

2. Planos del dispositivo

Además de en formato físico (papel) los planos también se encontrarán en la carpeta ANEXO II:

- Soporte Pie.
- Rotor (2x).
- Tutor gemelo.
- Articulación rodilla.
- Tutor muslo.
- Ensamblaje.