



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO SISTEMA DE ASISTENCIA A INIDENTES PARA PRÁCTICA DE LA EQUITACIÓN EN PISTA

Autor: María del Carmen Rubiales Mena

Director: José Daniel Muñoz Frías

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Sistema de asistencia a invidentes para práctica de la equitación en pista

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2019/2020 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.



Fdo.: María del Carmen Rubiales Mena Fecha: 30 / 06 / 2020

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO



Fdo.: José Daniel Muñoz Frías Fecha: 30 / 06 / 2020



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO SISTEMA DE ASISTENCIA A INVIDENTES PARA PRÁCTICA DE LA EQUITACIÓN EN PISTA

Autor: María del Carmen Rubiales Mena

Director: José Daniel Muñoz Frías

Madrid

SISTEMA DE ASISTENCIA A INIDENTES PARA PRÁCTICA DE LA EQUITACIÓN EN PISTA

Autor: Rubiales Mena, María del Carmen.

Director: Muñoz Frías, José Daniel.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este proyecto se diseña un sistema de ayuda a inidentes para la práctica de la equitación en pista. Cuenta con diversos sensores que miden la distancia a la que se encuentra el caballo del usuario de las paredes del recinto y del caballo precedente.

Palabras clave: Inidentes, ultrasonidos, distancia, caballos

1. Introducción

Este proyecto es la continuación del que realizó Miguel Ángel Sánchez Sánchez en 2017-2018 con el mismo nombre [1], que a su vez es continuación de otro proyecto anterior realizado en 2015-2016 por Paula Melero Álvarez. En dichos proyectos se diseñaba un sistema para ayudar a inidentes a practicar la equitación, en concreto la doma.

La doma consiste en la práctica de ejercicios y figuras dentro de un recinto cerrado de arena. Puede realizarse tanto de manera individual como en grupo, siendo esta última en la que se centra este Trabajo de Fin de Grado. Para dicha disciplina es necesario mantenerse a una distancia estipulada tanto de la pared de la pista como del caballo precedente por lo que se necesita un sistema de sensores que mida la separación de las referencias en todo momento.

En el proyecto de Miguel Ángel Sánchez, se diseñó un sistema que estudiaba si el inidente se encontraba demasiado cerca o demasiado lejos de las paredes laterales para, en caso de ser así, avisarle y que este variara su ruta. Para esto, empleó sensores de ultrasonidos que medían la distancia entre el caballo y la pared usando el tiempo que tardaba el sonido emitido en ir y volver desde que salía del caballo del usuario hasta que rebotaba en la pared y regresaba al sensor.

2. Definición del proyecto

En este proyecto, el objetivo es medir la distancia entre el caballo del inidente y el caballo que lleva delante. Es necesario que la distancia sea aproximadamente de 1.5 metros para que el deporte se practique de manera segura, si fuera menos, el caballo delantero podría darle una coza al trasero y si fuera más, el caballo del inidente se separaría demasiado y podría dividir la tanda, que es el grupo de caballos que trabajan juntos durante una clase. Esta distancia se mide con sensores de ultrasonidos colocados sobre los dos caballos que se comunican entre sí, como se muestra en la Ilustración 1.

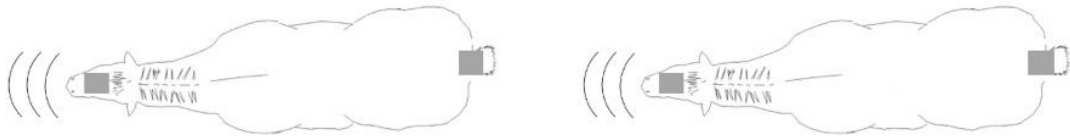


Ilustración 1: Esquema de funcionamiento del sistema [2]

3. Descripción del sistema y de la metodología

El emisor del caballo del usuario lanza una señal hacia el receptor del caballo delantero. Cuando este sensor recibe la señal, lanza otra de vuelta, que llegará hasta el receptor del caballo del invidente. Midiendo los tiempos que tarden las señales en ir y volver se puede obtener la distancia entre ambos caballos. Sabiendo esta distancia se puede avisar al invidente si se acerca o se aleja demasiado de su guía.

En la Ilustración 2 se muestra un esquema simplificado del funcionamiento del sistema que a continuación se detalla.

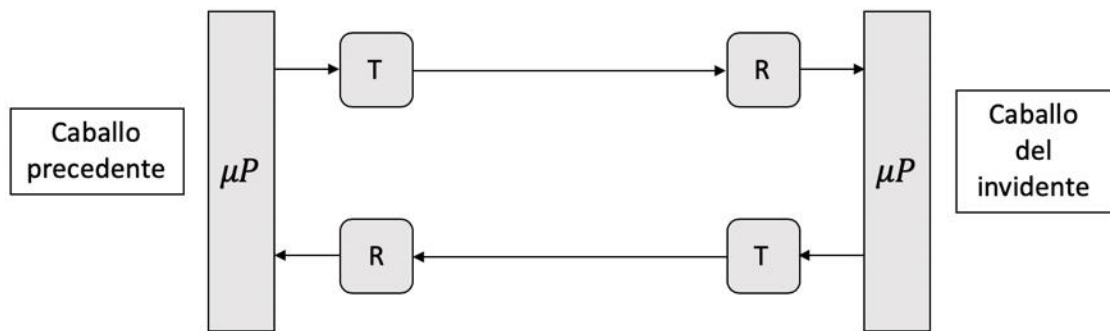


Ilustración 2: Funcionamiento detallado del sistema

3.1. Comunicación entre los sensores

La señal transmitida desde el microprocesador del caballo del invidente tiene una frecuencia de 40 kHz.

Se generan varios pulsos entre cero y cinco voltios durante la mitad del periodo y la otra mitad permanece a cero. Repitiendo este ciclo de pulsos dos veces se forma la señal emitida en ambas transmisiones, puede observarse la forma final de la onda en la Ilustración 3.

Al ser una señal con varios pulsos se evita confundirla a su recepción con una señal de continua, es decir, se facilita ver el cambio en la señal de entrada de los receptores cuando llega esta onda.

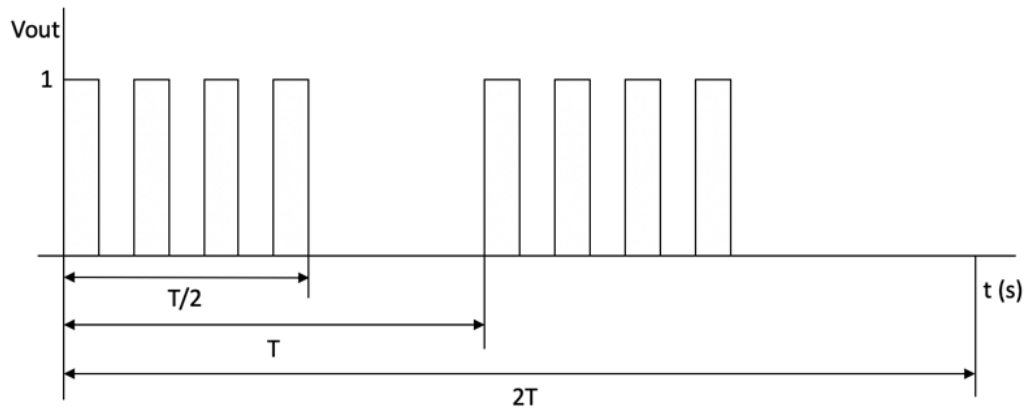


Ilustración 3: Ejemplo de señal PWM

Este proceso de comunicación se repite de manera constante para asegurar la medida correcta de la distancia entre ambos caballos. Si se diera el caso de que la señal se pierde entre ambos caballos, el microprocesador del caballo del invidente está programado para reemitir la primera señal si durante diez milisegundos no recibe nada desde el otro microprocesador.

3.2. Cálculo de la distancia entre los caballos

Para el cálculo de la distancia se necesita la velocidad del sonido en el aire y el tiempo transcurrido entre la emisión y la recepción de las señales en el microprocesador del invidente. El tiempo transcurrido entre esas señales se obtiene desde el microprocesador del caballo del invidente y la velocidad del sonido en el aire es 343.2 metros por segundo. Con estos datos se emplea la Ecuación:

$$\text{Ecuación: } distancia [m] = 0.5 \cdot velocidad_{sonido,aire} \left[\frac{m}{s} \right] \cdot tiempo [s] = 0.5 \cdot 343.2 \cdot tiempo$$

Como la distancia a medir con los sensores puede encontrarse entre cero y dos metros, el incremento de tiempo medido se encontrará entre cero y aproximadamente seis milisegundos.

Por lo tanto, el margen explicado anteriormente de diez milisegundos asegura además que se barre todo el rango de distancias medibles por los sensores.

3.3. Comunicación con el usuario

Una vez se tiene la distancia entre ambos caballos, si es necesario se avisa al usuario de que debe corregir la distancia que tiene en ese momento con el caballo precedente. Este aviso se hace mediante un actuador que vibra y se sitúa cerca de la mano del jinete, de manera que no pueda confundirse con los actuadores del proyecto anterior.

En función de si el jinete está demasiado cerca o demasiado lejos, el actuador vibra con más o menos intensidad para que el usuario pueda diferenciarlas.

3.4. Placa de circuito impreso (PCB)

Después de haber diseñado la comunicación con el usuario, y habiendo comprobado que tanto los circuitos como la programación de los microprocesadores es correcta mediante distintos ensayos, se une en una placa de circuito impreso los circuitos desarrollados en este proyecto con los desarrollados en el anterior.

De esta manera, se unifican en la misma PCB los circuitos para medir todas las distancias necesarias para asegurar la seguridad del jinete.

3.5. Metodología

Para realizar este Trabajo de Fin de Grado se han dividido las tareas en cuatro bloques. En primer lugar, se ha investigado el funcionamiento de una sola pareja de emisor/receptor. Una vez se ha conseguido medir distancias con ellos, se ha desarrollado la comunicación entre dos parejas de emisor y receptor, y se ha estudiado cuál es la señal que se obtiene como entrada al microprocesador del invidente.

Por último, se ha programado el cálculo de la distancia en función del tiempo de espera entre la emisión y recepción en el primer microprocesador y la señal de aviso al invidente de cómo corregir su distancia.

4. Resultados

Al finalizar este proyecto, se ha conseguido desarrollar un sistema de medición de distancia por ultrasonidos. Se cumplieron los objetivos fijados al comienzo del proyecto, ya que además de medirse la distancia entre ambos caballos, se ha desarrollado un sistema de aviso al invidente y se ha unificado este proyecto con el anterior en una placa de circuito impreso.

Aunque el sistema ha sido desarrollado, no se ha podido comprobar su funcionamiento ya que, debido a la pandemia provocada por el COVID-19, las instalaciones de la universidad han permanecido cerradas durante un largo periodo de tiempo sin posibilidad alguna de acceder a ellas.

5. Conclusiones

Al terminar el proyecto se espera que se pueda medir de manera correcta la distancia con el caballo precedente, por lo que el sensor funciona de manera eficaz. Además, al unificar el código y los circuitos en la PCB con los del proyecto anterior, se consigue medir todas las distancias pertinentes para que el jinete mantenga una distancia adecuada en todo momento.

De esta manera, se puede asegurar que el jinete llevará una posición correcta y segura durante la práctica de la equitación en pista.

En futuros proyectos, podría sustituirse la alimentación actual del sistema (una batería recargable) por otra de menor tamaño que permita que el sistema sea más compacto y cómodo de llevar para el usuario. También podría optimizarse el diseño de la PCB con el mismo propósito, reducir el tamaño final del sistema.

6. Referencias

1. Sánchez Sánchez, M. A. Proyecto Fin de Grado. Última consulta: Febrero 2020. Desde: <https://repositorio.comillas.edu/xmlui/handle/11531/31913>
2. Imagen retocada posteriormente. Última consulta: Febrero 2020. Desde: <https://www.bloquesautocad.com/caballo-en-planta/>

OBSTACLES DETECTION SYSTEM TO SUPPORT BLIND PEOPLE IN THE PRACTICE OF THE EQUESTRIAN SPORT

Author: Rubiales Mena, María del Carmen.

Supervisor: Muñoz Frías, José Daniel.

Collaborating Entity: ICAI – Comillas Pontifical University

ABSTRACT

In this project, a system to support blind people who practice horse riding is designed. It consists of various sensors that measure the distance between the user's horse and the wall of the enclosure and the preceding horse.

Keywords: Blind people, ultrasound, distance, horses

1. Introduction

This project is the continuation of the one carried out by Miguel Ángel Sánchez Sánchez in 2017-2018 [1] with the same name, which is the continuation itself of another previous project written in 2015-2016 by Paula Melero Álvarez. In these projects, a system which helps blind people when practicing horse riding, specifically dressage, was designed.

Dressage consists of the practice of exercises and figures within an enclosed sand enclosure. It can be done both individually and as a group, being the last option the one that concerns this end-of-grade work.

When doing dressage, it is mandatory to keep a stipulated distance between your horse and the one that is in front of you and between your horse and the walls of the track that you are following. For these reasons, the distance must be measured all the time by a system composed by sensors.

In Miguel Ángel's project, a system that studied the position of the blind user was designed. It measures the distance between his/her horse and the walls constantly and if it is not correct, it warns the user. For this purpose, ultrasound sensors were used. They calculated the time that it took the sound to come and go from the user's horse to the walls of the track.

2. Definition of the project

In this project, the objective is to measure the distance between the blind person's horse and the horse in front of him/her. It is necessary that the distance is approximately 1.5 meters for the sport to be practiced safely, if it were less the front horse could kick the other one and if it were more the blind person's horse would separate too much and could divide the group.

The distance is measured with two pairs of ultrasonic emitter and receiver that communicate with each other, each located on a horse as shown in Illustration 1.



Illustration 1: Description of the system [2]

3. Description of the system and methodology

The emitter of the user's horse will send a signal to the receiver of the front horse. When this sensor receives it, it will send back a return signal, which will reach the receiver of the blind person's horse. By measuring the time between the start of the first signal and the arrival of the second one, both in the user's horse, the distance between both horses.

Knowing this distance, the system can alert the blind person if he/she approaches or moves too far from his/her guide. Illustration 2 shows a simplified outline of how the system works, which will be explained below.

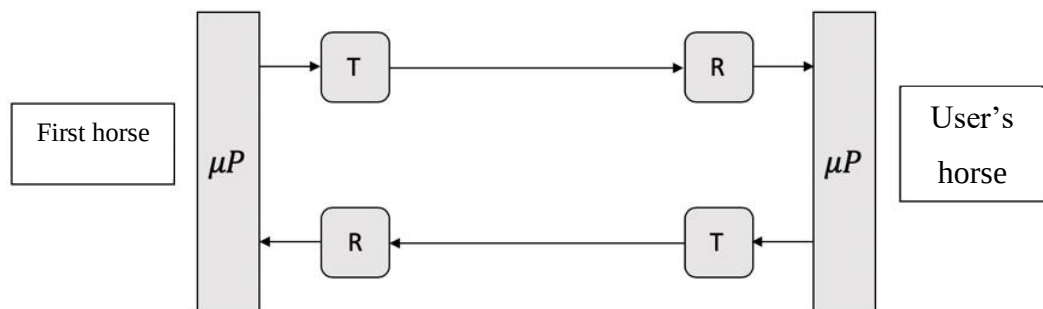


Illustration 2: Detailed description of the system

3.1. Communication between the sensors

The signal transmitted from the user horse's microprocessor has a frequency of 40 kHz.

Several pulses are generated between zero and five volts during the middle of the period and the other half of the time the signal remains at zero. Repeating this pulse cycle twice forms the signal emitted in both transmissions. An example of the signal emitted is shown in Illustration 3.

Being a signal with multiple pulses is useful to avoid confusions at its reception with a continuous signal, that is, it is easier to see the change in the input signal of the receivers when this wave arrives.

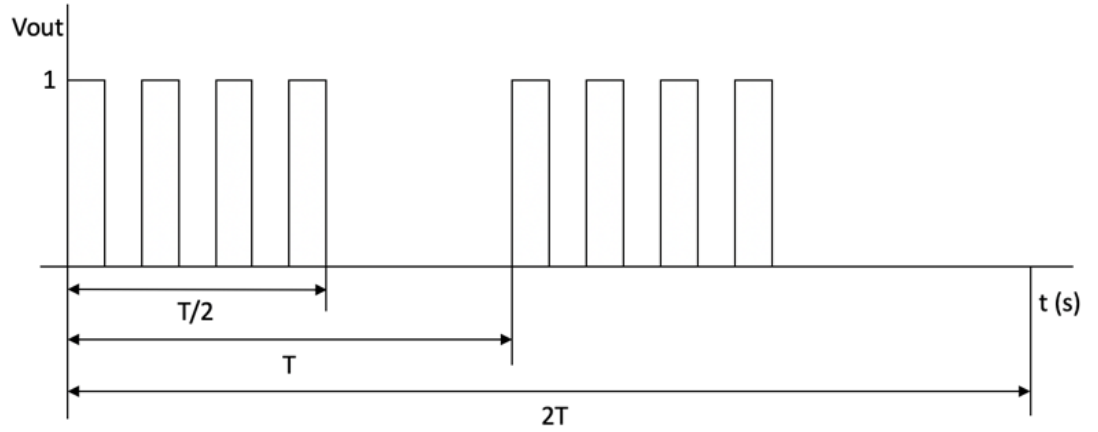


Illustration 3: PWM example

This communication process is constantly repeated to ensure the correct measurement of the distance between the two horses. If it was the case that the signal is lost between the two horses, the blind horse's microprocessor is programmed to send again the first signal if for ten milliseconds it receives nothing from the other microprocessor.

3.2. Distance between horses

When calculating the distance, the speed of sound in the air and the time elapsed between the emission and reception of the signals in the blind's microprocessor are required. That increment of time is provided by the principal microprocessor and the speed of sound in the air is 343.2 meters per second.

This distance is estimated using the following Equation 1.

$$\text{Equation 1: distance [m]} = 0.5 \cdot \text{speed}_{\text{sound,air}} \left[\frac{\text{m}}{\text{s}} \right] \cdot \text{time [s]} = 0.5 \cdot 343.2 \cdot \text{time}$$

Since the distance to be measured with the sensors can be between zero and two meters, the measured time increment will be between zero and approximately six milliseconds.

Therefore, all the distance that can be measured by the sensors would be measured before that preventive emission.

3.3. Communication with the user

Once you have the distance between the two horses, if necessary, the user is advised that he/she must correct the distance he/she has at that time with the preceding horse. This warning is made by means of a vibrating actuator that is placed near the rider's hand, so that it cannot be confused with the actuators of the previous project.

Depending on whether the rider is too close or too far away, the actuator vibrates with more or less intensity so that the user can differentiate them.

3.4. Printed circuit board (PCB)

Having designed the communication between the users and the microprocessors and having verified that both of the circuits and the microprocessors are working as they should by doing some tests, the circuits developed in this project are joined in a printed circuit board with those developed in the previous one.

By doing this, both projects are unified in the same PCB allowing it to measure all the distances necessary to ensure the safety of the rider.

3.5. Methodology

To perform this end-of-grade work, the tasks have been divided into four blocks. First, the operation of a single ultrasound sensor has been researched.

Once distances have been measured with it, the communication between two emitter and receiver pairs has been developed, and the signal obtained as an input to the blind's microprocessor has been studied.

Finally, the calculation of the distance based on the waiting time between the broadcast and reception on the first microprocessor and the warning signal to the blind person on how to correct its distance have been programmed.

4. Results

At the end of this project, an ultrasonic distance measurement system was designed. The objectives set at the beginning of the project were met, since in addition to measuring the distance between the two horses, a warning system has been developed and this project has been unified with the previous one on a printed circuit board.

Although the design of the system has been finished, its effectiveness has not been verified because, due to the pandemic caused by COVID-19, the university's facilities have remained closed for a long period of time without any possibility of accessing them.

5. Conclusions

At the end of the project, the distance with the preceding horse is correctly measured, so the sensor works effectively. In addition, by unifying the code and circuits in the PCB with those of the previous project, it is possible to measure all the relevant distances so that the rider maintains an adequate distance at all times. In this way, it can ensure that the rider will carry a correct and safe position during the practice of riding on the arena.

In future projects, the current system power (a rechargeable battery) could be replaced with a smaller one that allows the system to be more compact and comfortable to carry for the user. The PCB design could also be optimized for the same purpose, reducing the final system size.

6. References

1. Sánchez Sánchez, M. A. *Proyecto Fin de Grado*. Last visit: February 2020. From: <https://repositorio.comillas.edu/xmlui/handle/11531/31913>
2. The original image has been edited. Last visit: February 2020. From: <https://www.bloquesautocad.com/caballo-en-planta/>

Índice de la Memoria

<i>Índice de la Memoria.....</i>	<i>I</i>
<i>Índice de Figuras.....</i>	<i>VII</i>
<i>Índice de Tablas.....</i>	<i>IX</i>
<i>Capítulo 1. Introducción</i>	<i>11</i>
1.1 Motivación del proyecto	11
1.2 Recursos por emplear.....	12
1.2.1 Hardware	12
1.2.2 Software.....	13
1.2.3 Herramientas en el laboratorio.....	13
1.2.4 Resumen de las herramientas.....	14
1.3 Objetivos del proyecto	15
<i>Capítulo 2. Descripción de las Tecnologías.....</i>	<i>17</i>
2.1 Hardware.....	17
2.1.1 Microprocesador.....	17
2.1.2 Emisor y receptor de ultrasonidos	19
2.1.3 Actuador	20
2.1.4 Placa de circuito impreso (PCB)	21
2.2 Software	22
2.2.1 C	22
2.2.2 KiCad	24
<i>Capítulo 3. Estado del Arte.....</i>	<i>27</i>
3.1 Sensores de ultrasonidos	28
3.2 Sensores de infrarrojos.....	31
3.3 Sensores láser	32
3.4 Conclusiones	33
<i>Capítulo 4. Definición del Trabajo</i>	<i>37</i>
4.1 Justificación.....	37

4.2	Objetivos	37
4.2.1	Desarrollo del sistema de medida.....	38
4.2.2	Creación de dos señales.....	38
4.2.3	Cálculo del tiempo	39
4.2.4	Sistema de aviso	40
4.2.5	Diseño de la placa de circuito impreso (PCB)	40
4.3	Metodología	40
4.4	Escenario de uso.....	41
4.5	Impacto medioambiental.....	44
4.6	Impacto sobre el caballo.....	45
Capítulo 5. Sistema Desarrollado.....		47
5.1	Sensores de ultrasonidos	47
5.1.1	Los circuitos montados.....	47
5.1.2	Las señales de las pruebas.....	50
5.2	Comunicación entre los sensores	52
5.2.1	Rebote de la señal en una pared	52
5.2.2	Rebote de la señal en el otro caballo y regreso al punto de partida	53
5.2.3	Confusión y/o mezcla de las señales	53
5.2.4	Pérdida de la señal.....	53
5.3	Medida del tiempo.....	54
5.4	Actuador.....	56
5.4.1	El circuito montado	56
5.5	Alimentación.....	57
Capítulo 6. Diseño de la Placa de Circuito Impreso		59
6.1	Diseño de los circuitos	60
6.2	Conexión de los sensores y vibradores	61
6.3	Creación de las huellas de los componentes	62
6.4	Generación de la lista de conexiones	62
6.5	Diseño de la placa	62
6.6	Producción de la placa	64
6.7	Aspecto final	65
Capítulo 7. Código del Sistema		67

7.1 Puertos usados en este proyecto	67
7.2 Microprocesador del caballo del invidente	67
7.3 Microprocesador del caballo precedente.....	71
7.4 Combinación de los proyectos	73
Capítulo 8. Análisis de Resultados.....	77
8.1 Funcionamiento del Sistema	77
8.2 Pruebas realizadas	78
8.3 Aspectos a mejorar.....	80
Capítulo 9. Conclusiones y Trabajos Futuros.....	83
9.1 Conclusiones principales.....	83
9.2 Trabajos futuros	84
9.2.1 Monitorización del trabajo del animal.....	84
9.2.2 Aplicación para teléfono móvil.....	85
9.2.3 Conexión entre distintos jinetes invidentes	87
Capítulo 10. Objetivos del Desarrollo Sostenible	89
10.1 Introducción	89
10.2 Relación con el proyecto realizado	90
10.2.1 Deporte y discapacidad.....	91
Bibliografía y referencias	95
ANEXO I: Estudio Económico	99
Coste de los componentes	99
Coste del proyectista	100
Coste de los equipos del laboratorio.....	101
Conclusiones	102
ANEXO II: Guía de instalación.....	103
KiCad.....	103
MPLAB X IDE	108
Indicaciones Finales.....	111
ANEXO III: Código del Sistema Desarrollado	113
Microprocesador del caballo del usuario	114

<i>config.h</i>	114
<i>config.c</i>	114
<i>emisor_delante.h</i>	115
<i>emisor_delante.c</i>	116
<i>receptor_delante.h</i>	117
<i>receptor_delante.c</i>	117
<i>tiempo.h</i>	118
<i>tiempo.c</i>	118
<i>vibrador.h</i>	118
<i>vibrador.c</i>	119
<i>main.c</i>	120
Microprocesador del caballo precedente	123
<i>config.h</i>	123
<i>config.c</i>	123
<i>receptor_trasero.h</i>	124
<i>receptor_trasero.c</i>	125
<i>emisor_trasero.h</i>	125
<i>emisor_trasero.c</i>	126
<i>main.c</i>	127
Microprocesador para todos los sensores	129
<i>config.h</i>	129
<i>config.c</i>	129
<i>receptor_delante.h</i>	130
<i>receptor_delante.c</i>	131
<i>receptor_trasero.h</i>	131
<i>receptor_trasero.c</i>	132
<i>emisor_delante.h</i>	132
<i>emisor_delante.c</i>	133
<i>emisor_trasero.h</i>	134
<i>emisor_trasero.c</i>	134
<i>vibrador.h</i>	135
<i>vibrador.c</i>	136
<i>tiempo.h</i>	137

<i>tiempo.c</i>	137
<i>main.c</i>	137
<i>ANEXO IV: Diseño de la Placa</i>	143
<i>ANEXO V: Datasheets de los Componentes</i>	153

Índice de Figuras

Figura 1: Microprocesador dsPIC33FJ32MC202 [2]	18
Figura 2: Emisor y receptor de ultrasonidos [3]	19
Figura 3: Actuador [4]	20
Figura 4: Placa de circuito impreso [5]	21
Figura 5: Logo de MPLAB X IDE [8]	23
Figura 6: Logo del programa [10]	25
Figura 7: Sensores de ultrasonidos de coche aparcando [13]	28
Figura 8: Ejemplo de circuitos (emisor y receptor) usados de base [14]	29
Figura 9: Ejemplos de circuitos (emisor y receptor) usados de base [15]	31
Figura 10: Sensores de infrarrojos [17]	32
Figura 11: Medida de distancia con sensores láser [18]	33
Figura 12: Esquema del funcionamiento	34
Figura 13: Ejemplo de señal PWM	39
Figura 14: Pista de doma cercada [19]	42
Figura 15: Medidas aproximadas del animal, tomadas por Paula Melero [1]	43
Figura 16: Circuito de los emisores de ultrasonidos	48
Figura 17: Circuito de los receptores de ultrasonidos	49
Figura 18: Sensores enfrentados	50
Figura 19: Medida del tiempo entre la emisión de los pulsos y su recepción	51
Figura 20: Direccionalidades de los componentes [21]	52
Figura 21: Relación entre el tiempo y la distancia	55
Figura 22: Circuito montado para el actuador	57
Figura 23: Ejemplo de portapilas [22]	57
Figura 24: Proceso seguido para desarrollar la placa [23]	60
Figura 25: Una de las primeras versiones de la placa en el proceso de diseño	63
Figura 26: Elementos pendientes de conectar	64
Figura 27: Conexiones finales de la placa	64
Figura 28: Placa final del proyecto anterior [1]	65

Figura 29: Aspecto final del diseño del proyecto anterior [1]	66
Figura 30: Diagrama de estados del microprocesador del invidente	68
Figura 31: Diagrama de estados del microprocesador auxiliar	71
Figura 32: Posiciones orientativas de los cuatro sensores	77
Figura 33: Esquema de las pruebas entre dos circuitos	79
Figura 34: Ejemplo de pantalla de agenda virtual	85
Figura 35: Ejemplo pantalla de una alumna	86
Figura 36: Varios caballos conectados entre sí	87
Figura 37: Objetivos del desarrollo sostenible [25]	89
Figura 38: Ciclista ciega y su guía [27]	91
Figura 39: Postura correcta a la hora de realizar doma [29]	93
Figura 40: Página principal de KiCad	104
Figura 41: Ventana para elegir el sistema operativo	105
Figura 42: Ventana para descargar el programa	106
Figura 43: Ventana de instalación	107
Figura 44: Página principal de MPLAB-X-IDE	108
Figura 45: Opciones de descarga de MPLAB-X-IDE	109
Figura 46: Ventana de instalación de MPLAB-X-IDE	110
Figura 47: Confirmación de la instalación de MPLAB X IDE	110
Figura 48: Acuerdo de licencia MPLAB X IDE	111

Índice de Tablas

Tabla 1: Ejemplos de distancias y su relación con el tiempo	55
Tabla 2: Estimación del coste de los componentes	99
Tabla 3: Estimación del coste del proyectista	101
Tabla 4: Amortización de los equipos a lo largo del proyecto	102

Capítulo 1. INTRODUCCIÓN

Este proyecto es la continuación del proyecto de Miguel Ángel Sánchez Sánchez [1], llevado a cabo durante el curso 2017-2018. En ese proyecto se continúa con el desarrollo de un medidor de distancia por ultrasonidos.

El objetivo en ambos proyectos es diseñar un equipo de apoyo a personas invidentes para la práctica de la equitación en pista, en concreto para la doma. Miguel Ángel Sánchez se centró en desarrollar sensores que midieran la distancia que hay entre el jinete y las paredes laterales.

En este proyecto se desarrolla un sensor para medir la distancia entre el caballo del usuario y el precedente. Como en la doma los ejercicios se realizan dentro de un recinto cerrado, y en este caso específico también en grupos de mínimo dos personas, el sistema completo se encarga de medir las distancias entre el caballo del usuario y las paredes de la pista y el caballo del usuario y el caballo precedente.

En el proyecto de Miguel Ángel Sánchez los sensores utilizados fueron sensores de proximidad de aparcamiento de vehículos a los lados del animal para medir la distancia con las paredes. En este trabajo se continúa con el mismo sistema, empleando también sensores de ultrasonidos, aunque distintos a los de Miguel Ángel Sánchez, para medir la distancia con el caballo precedente al del invidente. Para que la comunicación se dé de manera efectiva, se sitúa un sensor en la parte delantera del caballo del usuario y otro en la parte trasera del caballo precedente y ambos se comunican entre sí.

1.1 MOTIVACIÓN DEL PROYECTO

La motivación para crear este sistema nace de la necesidad de ayudar a personas en riesgo de exclusión social por motivo de su discapacidad. Con ayuda de los sensores, los invidentes

pueden practicar la equitación de manera adaptada a sus necesidades, favoreciendo así su inclusión social.

Esta motivación está relacionada también con los Objetivos del Desarrollo Sostenible, en concreto con el número diez, como se explicará más adelante. Este objetivo busca reducir las desigualdades, que se dan por sexo, religión, raza, discapacidad o cualquier otro motivo.

En el caso de esta discapacidad, se busca conseguir una mayor inclusión social, y por ende una menor discriminación, al favorecer que cualquier persona pueda realizar cualquier actividad en el ámbito deportivo.

Además, este deporte trae múltiples ventajas para el usuario. A nivel psicológico mejora la autoestima y el estado de ánimo del jinete y a nivel físico, desarrolla la musculatura y el equilibrio del usuario. Al igual que cualquier deporte, el practicarlo beneficia al usuario mejorando su estado físico y su estado anímico.

1.2 RECURSOS POR EMPLEAR

Este proyecto se divide en dos grandes partes: software y hardware. A continuación, se resume qué abarca cada parte y se añade una lista de los materiales, tanto físicos como aplicaciones de ordenador, utilizados a la hora de realizar el proyecto. En el Capítulo 2, se detalla el funcionamiento de los componentes más importantes.

1.2.1 HARDWARE

Este sistema cuenta con dos microprocesadores, uno situado en el caballo delantero y otro situado en el caballo del usuario. Cada microprocesador se encuentra incorporado en una placa de circuito impreso (PCB) donde están los circuitos de los sensores de ultrasonidos. Además, la PCB del caballo del invidente cuenta con el circuito que realizó Miguel Ángel Sánchez para la medida de las distancias desde el caballo hasta las paredes laterales del recinto.

Una vez se realizó el sistema con el sensor delantero y los laterales, se añadió también el sensor trasero en la misma PCB de manera que todos los sensores desarrollados a lo largo de este proyecto y sus predecesores se unifican en una sola placa y dependen de un solo microprocesador. De esta forma, podrían conectarse varios caballos en serie de distintos usuarios, lo cual favorecería que hubiera clases más numerosas.

Las partes más destacables a nivel hardware son los sensores de ultrasonidos, el actuador y el microcontrolador. Se detalla su funcionamiento en el Capítulo 2.

1.2.2 SOFTWARE

La programación de los dos microprocesadores utilizados en este proyecto se hace en C, por ser unos de los lenguajes de programación más extendidos y conocidos. Además, en los proyectos anteriores los sensores laterales y el actuador también se programaron en C por lo que utilizar el mismo lenguaje de programación facilita la fusión de los dos proyectos. Para programar los microprocesadores se ha usado el programa MPLAB X IDE, que es un programa gratuito disponible para MacOS, Linux y Windows.

Además, para el diseño de la placa de circuito impreso se usa el paquete de software libre KiCad. Esto permite crear de manera sencilla el esquema de un circuito y convertirlo a una placa de circuito impreso.

En esta PCB se unen en un solo proyecto todos los sensores desarrollados, para que el sistema final diseñado cuente con la medida de todas las distancias pertinentes para que se incremente la seguridad del usuario mientras practica la equitación.

1.2.3 HERRAMIENTAS EN EL LABORATORIO

A la hora de desarrollar el proyecto, se trabajó en uno de los laboratorios electrónicos de ICAI. En este, se usaron distintas herramientas: un generador para alimentar los circuitos, un osciloscopio para ver las ondas de ultrasonidos con las que se trabaja y cables y sondas para poder conectarlo todo.

1.2.4 RESUMEN DE LAS HERRAMIENTAS

En este apartado se enumeran todas las herramientas utilizadas, tanto a nivel de programas de ordenador, como los componentes de hardware y los elementos utilizados en el laboratorio a la hora de desarrollar el proyecto. Estos últimos se usan para trabajar en pasos intermedios, pero no se necesitan para que funcione el sistema final.

- MPLAB-X-IDE
- KiCad
- Microprocesadores dsPIC33FJ32MC202
- Sensores de ultrasonidos CEBEK C-7210
- Elementos pasivos (resistencias, condensadores)
- Amplificador operacional MCP6002
- Transistor 2N3905
- Diodos shottky
- Diodos rectificadores
- Vibrador Z7AL2B1692082
- PCB
- Varios metros de cable
- Herramientas del laboratorio
 - Osciloscopio *Tektronix TDS 210*
 - Fuente de alimentación *KAISE DF1731SB5A*
 - Cables y sondas varios para los distintos ensayos

Además de estos componentes, a la hora de diseñar y montar la PCB se añaden los componentes que se utilizaron en el proyecto anterior para la medida de las distancias entre el caballo y las paredes laterales.

Estos componentes se añaden utilizando los mismos circuitos que desarrolló Miguel Ángel Sánchez y se colocan en los mismos pines del microprocesador, de forma que se modifica su proyecto lo mínimo posible y solo a nivel software.

Los programas que se modifican solo sufren cambios a nivel de organización o para añadir más variables en ellos, pero no cambian su estructura ni su funcionalidad. De esta manera, el trabajo que se desarrolló anteriormente es totalmente compatible con el desarrollado en este proyecto y la PCB abarca los dos de manera conjunta.

1.3 OBJETIVOS DEL PROYECTO

Los objetivos fijados a la hora de realizar este proyecto se han marcado de manera escalonada, por lo que se han ido consiguiendo uno detrás de otro. Estos objetivos son:

- Desarrollo del sistema de medida de distancia con el caballo precedente usando ultrasonidos.
- En este sistema de ultrasonidos, creación de dos señales distintas para que los sensores no las confundan. Investigación de la posible confusión de las señales y, si se da el caso, codificación de las señales para su diferenciación.
- Cálculo del tiempo entre la emisión y la recepción en el caballo del invidente.
- Implementación de un sistema de aviso al jinete, que servirá para indicar si este debe variar la distancia que mantiene con el caballo precedente.
- Combinar el diseño de estos sensores con el de los sensores anteriormente desarrollados en una placa de circuito impreso conjunta.

El resultado final cuando se acaba este proyecto es una placa de circuito impreso, dependiente de un microprocesador. Esta placa se coloca en el caballo del invidente y se encarga de controlar las distancias a las que se encuentra el usuario de los obstáculos para garantizar su seguridad.

En esta placa se encuentran:

- Un sensor a cada lado que mide la distancia con las paredes del recinto. Estos sensores se comunican con el microprocesador, permitiéndole calcular la distancia que hay entre el caballo y las paredes laterales.

- Dos actuadores, uno en cada pierna del jinete, que le informan sobre cómo de cerca o de lejos se encuentra del recinto de la pista.
- Un sensor en la parte delantera del caballo que se comunica con el caballo precedente y sirve para medir la distancia entre ambos. Además, cuentan con un vibrador que comunica al invidente si está demasiado lejos o demasiado cerca del caballo precedente.
- Un sensor en la parte trasera del caballo, que se comunica con el caballo posterior y sirve para que este otro caballo mida la distancia entre ambos. Sería un paso intermedio para que el caballo posterior controle la distancia que están guardando los dos animales en todo momento.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

A continuación, se comentan las tecnologías que se utilizan en este Trabajo de Fin de Grado. Se han dividido en dos grandes familias, hardware y software, ya que el sistema desarrollado combina ambas.

En este capítulo solo se mencionan los componentes de los sensores desarrollados para medir la distancia entre el caballo del usuario y el precedente, aunque en la PCB se combine este proyecto con los anteriores.

Los componentes utilizados en los otros proyectos han sido descritos en las memorias de estos, por lo que comentarlos aquí sería redundante.

2.1 *HARDWARE*

Los cuatro componentes a destacar a nivel hardware son los microprocesadores, los sensores de ultrasonidos, el actuador utilizado y la placa de circuito impreso (PCB).

En el montaje de los circuitos también se utilizarán amplificadores operacionales, divisores de tensión y más componentes de uso generalizado, estos se mencionarán en el Capítulo 5. al explicar los circuitos montados pero su funcionamiento es el usual por lo que no merecen una explicación detallada.

2.1.1 MICROPROCESADOR

El microprocesador es un circuito integrado cuya función es procesar datos y controlar distintos dispositivos exteriores a él mismo, llamados periféricos, además de realizar operaciones lógico-matemáticas.

Para la realización del proyecto se utiliza el microprocesador dsPIC33FJ32MC202, de 16 bits, que es el mismo que se utilizó en el proyecto antes mencionado de Miguel Ángel Sánchez.

Como se puede ver en la Figura 1, el microprocesador cuenta con 28 pines, de los que seis son canales para entradas analógicas. La mayoría de los pines pueden usarse de entradas o salidas, teniendo en cuenta que las entradas toleran hasta 5 voltios y que las salidas serán de máximo 3 voltios.

La intensidad por estos pines será del orden de 10 mA. Las entradas al microprocesador pueden ser digitales ('0', '1') o analógicas (un valor de tensión que se convertirá en un valor entre '0' y '1023').

En el caso de este proyecto de fin de grado la señal que reciben los receptores de ultrasonidos es analógica por lo que se utilizará de entrada en cada montaje uno de los seis canales que tiene el microprocesador. Además, se programará un módulo de conversión analógica digital para que el microprocesador pueda trabajar con la señal recibida.



Figura 1: Microprocesador dsPIC33FJ32MC202 [2]

Los periféricos que controla este microprocesador en el diseño final son el emisor y receptor de ultrasonidos delanteros, el emisor y receptor de ultrasonidos traseros, el actuador que avisa de la distancia con el caballo precedente, los dos sensores para medir las distancias laterales con el recinto y los dos actuadores para indicar la distancia con las paredes.

Durante el desarrollo de esta memoria se describen los diversos ficheros que se programan en el microcontrolador y cómo se van a usar las salidas de este para que los componentes interactúen con él.

2.1.2 EMISOR Y RECEPTOR DE ULTRASONIDOS

Para medir la distancia entre los dos caballos se va a mandar una señal de ultrasonidos desde el caballo del jinete invidente al caballo de la posición delantera, que tendrá un receptor.

Cuando este caballo reciba la señal del primero mandará otra señal distinta de vuelta. Finalmente, cuando esta señal llegue de vuelta al primer microprocesador se usará el tiempo que ha tardado en llegar la segunda señal desde que se emitió la primera para calcular la distancia entre los caballos.

En caso de que esta distancia fuera incorrecta se avisaría al usuario con el actuador, haciendo que este vibre con mayor o menor intensidad en función de la posición del jinete.

Para que cada pareja de receptor-emisor no confunda la señal que envía con la señal que debe recibir y para evitar pérdida de señales por el camino, se envían dos señales distintas, a la misma frecuencia, usando 2 cápsulas separadas emisor-receptor como las de la Figura 2 y no un sistema que ya tenga conectados todos los componentes entre sí.

Las señales se consideran distintas porque se emiten en dos partes del proceso de medida distintas, de manera que nunca están emitiéndose a la vez.



Figura 2: Emisor y receptor de ultrasonidos [3]

En el proyecto anterior también se emplearon sensores de ultrasonidos para medir la distancia, pero eran los mismos que emitían la señal los que la recibían de vuelta, rebotando

la señal a mitad de este proceso con las paredes del recinto. En este caso no contaban con otro sensor intermedio que recibiera la señal y mandara otra distinta de vuelta.

Para medir las distancias laterales este sistema es suficiente porque las paredes no van a desplazarse de su sitio ni a realizar movimientos extraños, ya que son objetos de construcción, la señal podrá ir, rebotar y volver sin problemas y no se perderá por el camino.

En el proyecto actual no se puede usar el mismo sensor debido a que el caballo del invidente podría mover la cabeza o el caballo precedente podría mover la cola y podría perderse la señal. Si la señal se perdiese, podría llevar a que la distancia no se midiera correctamente y el usuario podría estar a una distancia no adecuada y no saberlo.

Añadiendo un sensor intermedio se reducen las posibilidades de que la señal se pierda por el camino, por lo que se obtiene una medida más segura. Por eso, los sensores de ultrasonidos que se van a utilizar en este proyecto son las cápsulas de ultrasonidos de Cebek C-7210 en los que puede controlarse cada cápsula de manera independiente.

2.1.3 ACTUADOR

Para indicar al jinete si debe aumentar la separación frontal o disminuirla, se usará un actuador vibrador. No se pueden emplear actuadores sonoros ya que estos podrían molestar o distraer al animal durante la clase y reaccionaría de manera impredecible, pudiendo ser peligroso para el usuario.



Figura 3: Actuador [4]

Los actuadores que indican la separación que guarda el caballo con las paredes del recinto van colocados uno en cada pierna del usuario. Como se observa en la Figura 3, estos actuadores son de pequeño tamaño así que para avisar de la separación frontal pueden colocarse en la parte delantera del jinete, alrededor de la muñeca derecha o cerca del puño, de manera que se pueda diferenciar si la vibración indica que se ha separado de la pared o del caballo delantero.

El circuito desarrollado para incorporar este actuador está hecho de manera que el ciclo de trabajo del actuador no permita que este supere la tensión máxima que soporta.

2.1.4 PLACA DE CIRCUITO IMPRESO (PCB)

Una placa de circuito impreso es aquella en la que los componentes y las uniones entre ellos (conectores) están dentro de la misma estructura mecánica. Como se observa en la Figura 4, las conexiones entre los distintos componentes de los circuitos se hacen con un filamento dorado de cobre, que se encuentra entre capas de material aislante no conductor.



Figura 4: Placa de circuito impreso [5]

Una vez impresa la placa, el resto de los componentes se añaden en la capa superficial, bien de manera automática (máquinas pick and place que colocan los componentes) o a mano (con un soldador de estaño).

En la mayoría de los diseños electrónicos se utilizan placas de circuito impreso ya que permiten realizar diseños más complejos y pequeños de manera más rápida. Además, permite la automatización del proceso, lo cual vuelve más rentable una producción a gran escala.

En este proyecto se realiza una PCB combinando los circuitos diseñados de manera específica para los sensores que detectan el caballo precedente, los sensores para el caballo posterior y los sensores realizados por Miguel Ángel Sánchez en el proyecto anterior.

Al convertir los circuitos diseñados en una PCB se reduce mucho el tamaño del diseño, lo cual hace que sea más fácil colocarlo en el caballo y en el jinete y transportarlo, haciendo que este sistema sea menos incómodo de añadir a la equipación del jinete que si fueran los circuitos desarrollados en primera instancia sin la PCB.

2.2 SOFTWARE

En este apartado se explican los aspectos más relevantes del software del sistema desarrollado. En el “ANEXO II: Guía de instalación”, se explica cómo instalar los programas que se han utilizado a lo largo del trabajo y que se consideran imprescindibles para seguir avanzando en proyectos posteriores.

2.2.1 C

Como se indicó más atrás, la programación de los microprocesadores se hace en C. Es un lenguaje de gran facilidad para escribir códigos sencillos y compactos [7]. Fue creado por Dennis Ritchie, lanzándolo en 1972 y es compatible con Linux, Windows y Unix y con prácticamente todos los sistemas operativos que existen actualmente.

El proyecto se ha programado utilizando un ordenador con sistema operativo macOS Catalina, y al no ser compatible la aplicación donde se programa, MPLAB-IDE, con el sistema operativo del ordenador utilizado durante este proyecto se ha realizado una partición del disco en la que se ha instalado Windows 10.

Aunque el programa en sí es compatible con macOS, no es compatible, al menos en el periodo de tiempo durante el que se ha realizado este trabajo, con la versión Catalina de macOS, que se lanzó a finales de 2019.

El logo del programa utilizado se muestra en la Figura 5.

El código se compone de un fichero principal del programa, llamado `main`, y varios ficheros adicionales a los que se accede desde el principal. Estos ficheros adicionales se encargan cada uno de uno de los componentes del sistema en específico, pero todos pueden interactuar entre sí con ayuda del principal. De esta forma, todas las funciones dependen del `main` que se encarga de coordinar las distintas funcionalidades de todos los componentes del proyecto.



Figura 5: Logo de MPLAB X IDE [8]

Por lo tanto, se tiene una función que se encarga de cada emisor de ultrasonidos, otra de cada receptor, otra del actuador y otra del cálculo del tiempo.

Como en el caballo del invidente y el precedente las señales que esperan y emiten los receptores y emisores funcionan de manera distinta, si se colocan los sensores por separado

habrá dos main distintos en los dos microprocesadores que se utilizan, cada uno con sus programas adicionales específicos.

En el caso de colocar todos los sensores juntos hay un solo programa principal que fusiona las medidas de las distancias laterales, precedente y posterior para que, usando pines distintos, se pueda controlar el sistema completo.

En este caso, se adaptan los códigos desarrollados para programar dos microprocesadores por separado y también, aunque en menor medida, los programas desarrollados por Miguel Ángel Sánchez para su proyecto.

Las partes y el funcionamiento específicos del código se explican en el Capítulo 7.

2.2.2 KiCAD

Este programa de código libre puede ejecutarse en Linux, Windows y Mac OSX. También está disponible para un gran número de distribuciones de Linux. Sirve para crear esquemas electrónicos y circuitos impresos [9].

Además, una vez creado el esquema se puede comprobar el resultado mediante una de las herramientas que incluye el software y que indica si el diseño de la placa es correcto.

En este proyecto se utiliza para diseñar la placa en la que irán los circuitos con los que trabajan todos los sensores desarrollados. En esta placa se han acoplado todos los circuitos desarrollados durante este proyecto y sus predecesores con todos los sensores y actuadores necesarios para su correcto funcionamiento.

Al ser una PCB, se reduce el tamaño que tendría el sistema final respecto a los mismos circuitos desarrollados de manera independiente en protoboards. Cuanto más compacto sea el sistema, más fácil será usarlo y transportarlo para el usuario y más se extenderá su uso, llegando a un mayor número de jinetes invidentes.

La placa se alimenta usando una batería de pilas de 1.5 V conectadas en serie, obteniendo una tensión final de 12 V. De esta manera, el sistema desarrollado es independiente y no necesita conectarse a otra fuente de alimentación más allá de las ocho pilas.

El logo del software se puede observar en la Figura 6. Es un programa que se puede descargar de manera gratuita y que cuenta con infinidad de tutoriales en la red por lo que es relativamente sencillo aprender a usarlo.

Además, en la misma página en la que se descarga el programa se puede acceder también a tutoriales, librerías con los símbolos para desarrollar los circuitos, librerías con las huellas de un gran número de componentes, ejemplos desarrollados en 3D y múltiples recursos más. Su uso está ampliamente extendido lo cual facilita encontrar información de calidad sobre su uso y sus funciones.



Figura 6: Logo del programa [10]

Capítulo 3. ESTADO DEL ARTE

Hoy en día las personas discapacitadas cuentan con adaptaciones para realizar cualquier actividad de la vida cotidiana. En concreto para las personas invidentes existen desde teléfonos móviles, donde todas las interacciones van acompañas de indicaciones de voz para ayudar al usuario [11], hasta escáneres que les leen documentos y libros [12]. También cuentan con ayudas para moverse por la calle, cocinar y cómo no, para practicar deporte.

En la mayoría de las disciplinas deportivas los discapacitados van acompañados de otra persona que les sirve de guía. Esta persona realiza la misma disciplina deportiva que ellos, acompañándolos a su ritmo para guiarles. Por ejemplo, en el caso del ciclismo, comparten una bicicleta de dos plazas en la que el guía va delante y el deportista discapacitado detrás y en el caso del atletismo, dependiendo de la categoría de la competición, pueden ir unidos entre sí con una cinta a la altura del muslo y corren la carrera juntos o pueden ir unidos por una cinta que sujetan con las manos.

En la equitación no es posible que esta persona los acompañe montada sobre el mismo caballo, ya que el sobrepeso podría lesionar al animal. Tampoco puede ir en el suelo andando a la par del binomio (conjunto del jinete y el caballo), porque desde ese ángulo y distancia no es una verdadera ayuda para el jinete, además de que le costaría mantenerse siempre a una distancia del caballo constante ya que tendría que avanzar al mismo ritmo que este.

Por ello, se decidió desarrollar un sensor que midiese la distancia que guarda el jinete con las paredes del recinto y con el caballo delantero, que en este caso le serviría de guía.

Hasta el momento, existen multitud de sensores para medir la distancia entre dos objetos. La mayoría de estos sirven para medir distancias entre dos objetos que están quietos, y en algunos casos que se mueven a baja velocidad.

Tres claros ejemplos de sensores de medida de distancia son los sensores de ultrasonidos, los sensores de infrarrojos y los sensores por láser, de los cuales se explica a continuación

su funcionamiento, y los problemas que podrían darse al querer usarlos para el propósito de este trabajo.

3.1 *SENSORES DE ULTRASONIDOS*

La manera más efectiva de medir la distancia es usar sensores de ultrasonidos. La mayoría de los sensores de este tipo son similares a los de coches. Como se observa en la Figura 7, el coche emite una onda de ultrasonidos que impacta contra los objetos de alrededor del vehículo para después regresar al sensor. En función del tiempo que tardan estas ondas en salir y volver al sensor, se calcula la distancia a la que están los objetos y se avisa con señales acústicas de la misma al conductor.

Estos sensores son los que usó Miguel Ángel Sánchez en su proyecto de fin de carrera para detectar la distancia a la que se encuentran las paredes del recinto dónde se realiza el deporte. En el caso de la medida de distancia con el caballo delantero no pueden emplearse ya que el caballo podría mover la cabeza o modificar la inclinación de esta y la señal se perdería. Para hacer más segura la medida de la distancia en este proyecto se optó por dos transmisores y dos receptores, de manera que se añade un paso intermedio a la medida.

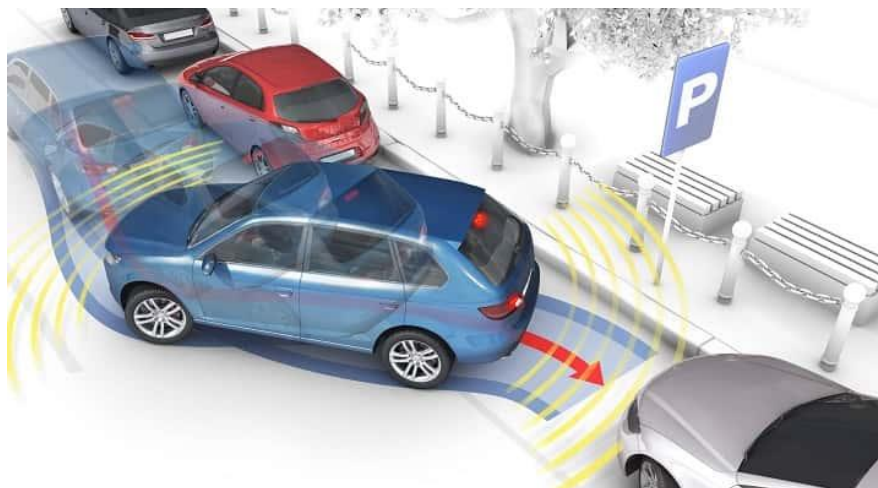


Figura 7: Sensores de ultrasonidos de coche aparcando [13]

En primer lugar, se emite una señal desde el sensor situado en la cabeza del caballo del jinete invidente. Esta señal se recibe en el sensor situado en la grupa del caballo precedente.

Estos sensores también podrían situarse sobre el cuerpo de los jinetes, por ejemplo, sus cascos, siempre y cuando se mantengan al mismo nivel durante toda la práctica de la equitación.

Cuando esta señal es recibida, desde el sensor del caballo precedente se manda de vuelta una señal distinta al caballo que comenzó la comunicación.

Por último, cuando esta señal se recibe, se mide el tiempo que se ha tardado desde la emisión de la primera señal hasta la recepción de la segunda y se calcula la distancia entre los animales. En caso de ser necesario, se comunica al invidente que se acerque o se aleje más del caballo precedente para guardar una distancia adecuada.

Este tipo de sensores no son tan habituales ya que normalmente cuando se trabaja con una pareja de emisor y receptor se hace en conjunto, es decir, se emite una señal y se recibe en su receptor pareja.

La información disponible de los mismos es escasa, y se centra en el funcionamiento de los sensores de ultrasonidos de Cebek C-7210 utilizados en este proyecto.

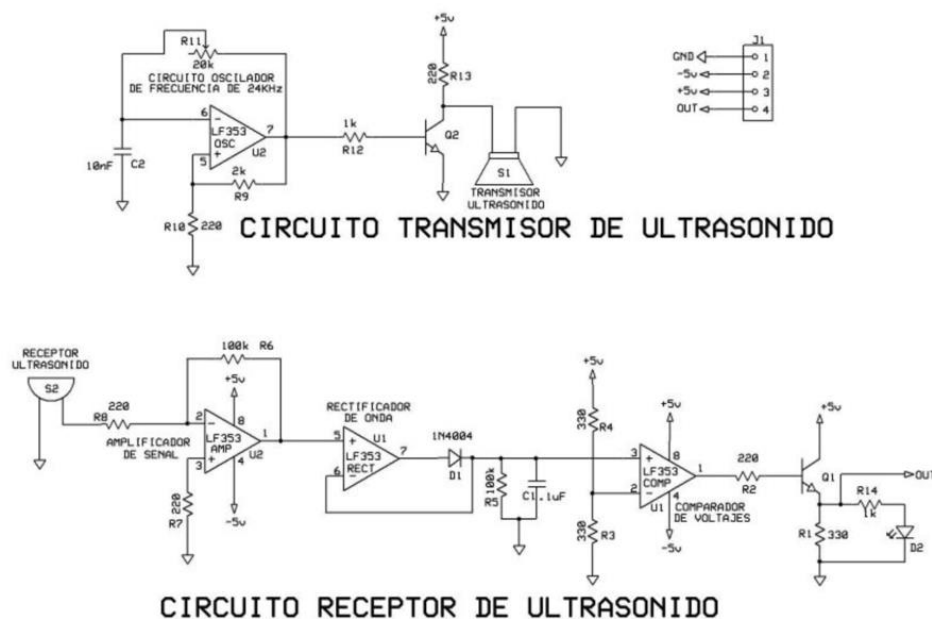


Figura 8: Ejemplo de circuitos (emisor y receptor) usados de base [14]

Investigando en diversos foros de electrónica [14] se encontraron dos ejemplos de circuitos montados con cápsulas individuales de ultrasonidos. Con estos ejemplos de circuitos se analizó el funcionamiento de los sensores y se adaptó para su posterior uso, sirviendo de ejemplo para entender el funcionamiento de los sensores de ultrasonidos.

Los circuitos de la Figura 8 cuentan con un oscilador en el emisor, para crear una onda senoidal que se transmite como onda de ultrasonidos mediante el emisor de ultrasonidos. En este emisor se transforma la corriente en energía mecánica, que activa los mecanismos del emisor y crea la onda de ultrasonidos.

En el circuito receptor hay un amplificador, un detector de pico y un comparador de voltaje, los cuales hacen que la onda que llegue a la salida total de ese circuito sea una onda diferente a la que llegó al receptor, convirtiendo el tren de pulsos que llega al sensor en una onda cuadrada con la que puede trabajar el microprocesador.

Los circuitos de la Figura 9 son más sencillos, cuentan solo con condensadores, resistencias, inductancias, diodos y un transistor. En este caso el circuito no funcionaba para el propósito del proyecto ya que la señal que llegaba al emisor no se transmitía bien y en el receptor apenas se llegaba a tener una salida apreciable.

En este diseño, no es tan importante la amplitud de la señal recibida como el retraso desde que se emite la señal hasta que se recibe la siguiente por lo que se amplía lo máximo posible para asegurar que se detecta cuándo llega esa señal.

Si bien es verdad que era un circuito más simple, la tensión a la que se alimentaba el receptor era muy pequeña por lo que la onda creada también tenía una amplitud pequeña y se perdía más por el camino.

El diseño estaba hecho para que funcionase al igual que los sensores de aparcamiento de los coches, emitiendo en un lado una señal y esperando la llegada de la señal rebotada en el objeto. Además, medía pequeñas distancias por lo que no era muy adaptable para el propósito de este proyecto.

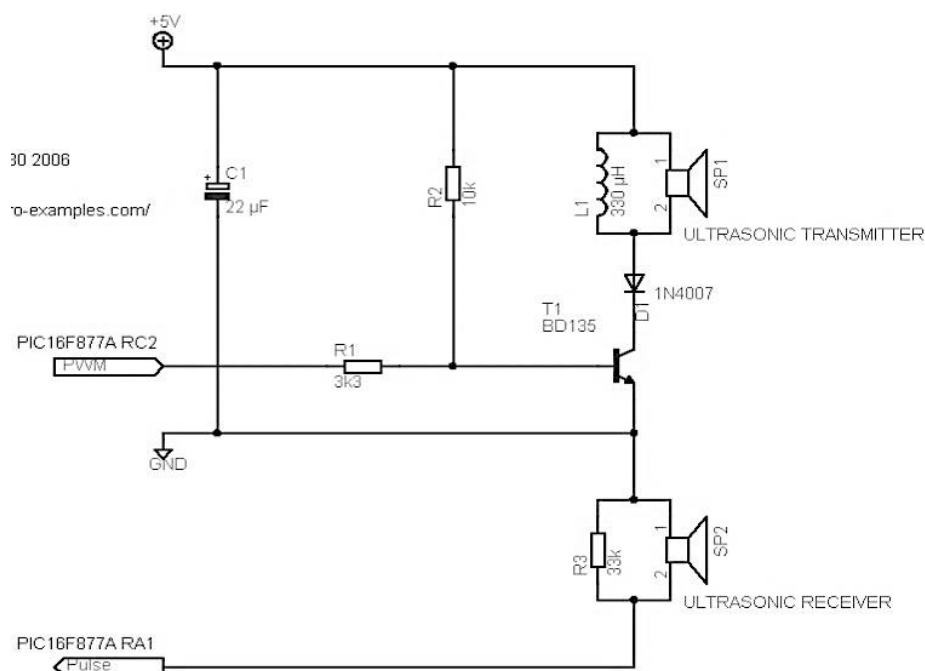


Figura 9: Ejemplos de circuitos (emisor y receptor) usados de base [15]

Aunque este circuito se estudió, se descartó a la hora de montar el diseño final y se optó por desarrollar un circuito más similar al de la Figura 8, aunque adaptándolo para que funcionase como se esperaba en el sistema desarrollado.

3.2 *SENSORES DE INFRARROJOS*

También existen sensores para medir distancia por infrarrojos. Suelen ser de menor alcance, entre 30 y 60 centímetros por lo que no son útiles para el propósito de este proyecto. Además, si algún objeto se situase entre los dos sensores, por ejemplo, la cola del caballo precedente, la medida que se obtendría no sería la correcta.

Estos sensores emiten un haz de luz infrarroja que se refleja en el objeto del que se quiere saber la distancia, calculándola por triangulación del haz de luz [16]. Por ello, miden desde que se emite el haz de luz hasta que este se corta, normalmente porque se ha llegado al objeto del que se quiere medir la separación, aunque podría ser de un objeto que se cruce en el camino del haz de luz dando lugar a una medida errónea.

Otro problema de este tipo de sensores, además del explicado en el párrafo anterior, es que la respuesta que se obtiene no es lineal, por lo que la medida no es precisa.

Podría tenerse un cambio significativo en el voltaje del sensor y la respuesta obtenida por el sensor sería similar a una de menor voltaje. Por ello, no podría diferenciarse correctamente y de manera precisa cuál es la distancia entre los dos animales.

En la Figura 10 puede observarse una pareja de estos pequeños sensores.



Figura 10: Sensores de infrarrojos [17]

3.3 SENSORES LÁSER

En este tipo de sensores, se calcula la distancia en función del tiempo que tarda el láser en ir, rebotar en la superficie del objeto y regresar al emisor. De esta forma, su funcionamiento es similar al de las ondas de ultrasonidos, diferenciándose en el tipo de señal que emite y parametriza.

Estos sensores son muy precisos, pero se descartaron para realizar este proyecto porque también son muy direccionales y una pequeña variación en la posición del jinete o de los caballos, imposibilitaría la medida. Estas variaciones de posición se darán durante la práctica

del ejercicio por las oscilaciones en el movimiento de los materiales por lo que no es aconsejable este tipo de sensores para este proyecto.

En la Figura 11 puede observarse un ejemplo de aplicación para este tipo de sensores, la construcción. En este sector se utiliza para medidas puntuales y que necesitan precisión, como podría ser el largo de una habitación.



Figura 11: Medida de distancia con sensores láser [18]

3.4 CONCLUSIONES

Si bien es cierto que existen múltiples maneras de medir la distancia entre dos puntos, algunas son más adecuadas para este proyecto que otras.

La medida de distancia mediante infrarrojos no es muy adecuada para este proyecto ya que cualquier objeto que se colase de manera inesperada en el haz de luz falsearía la medida. Por ejemplo, la cola del caballo al moverse podría hacer que el sensor pensara que los caballos están más cerca entre sí de lo que de verdad están, indicándole de manera errónea al usuario que debe corregir su posición.

Se ha descartado el uso de sensores de infrarrojos en este proyecto, además de por la posibilidad de que la medida se falsease por objetos indeseados, porque es un sistema más sensible que podría verse dañado con los movimientos bruscos del animal, porque sirve para medir cortas distancias y porque a la larga sería más caro mantenerlo y repararlo.

También se ha descartado medir la distancia con sensores con láser ya que las pequeñas oscilaciones del movimiento natural de los caballos harían que los sensores no funcionaran correctamente.

En definitiva, la tecnología más adecuada de las explicadas en este capítulo sería la medición de distancias por ultrasonidos, siempre y cuando se adapten los sistemas ya existentes para hacer la transmisión de la señal más segura y minimizar las posibilidades de que se pierda la señal con la que se trabaja.

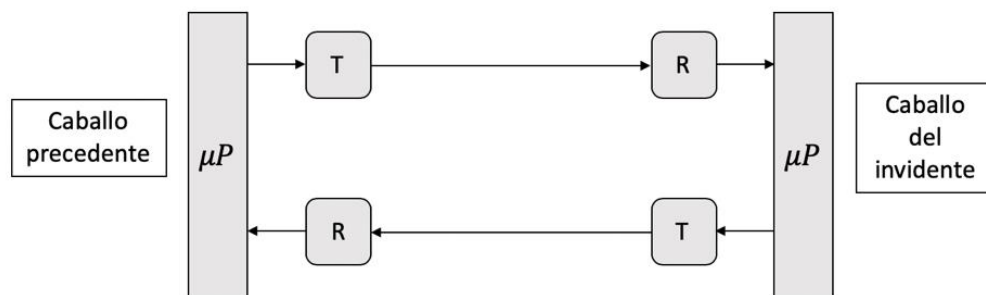


Figura 12: Esquema del funcionamiento

Para adaptar esta transmisión de señales, simplemente es necesario meter un paso intermedio en el circuito, de manera que no solamente se transmite una señal de ultrasonidos sino dos. Se emite una primera señal desde el caballo del usuario al caballo precedente. Cuando este segundo caballo recibe la señal, envía otra de vuelta. Por último, cuando se recibe la segunda señal en el caballo del usuario se mide la distancia entre ambos animales en función del tiempo que ha pasado en ese microprocesador desde que se emitió la primera señal hasta que se recibió la segunda. Este modo de funcionamiento se ilustra en la Figura 12.

En este diseño la medida importante es la del tiempo de retraso entre las dos señales con las que trabaja el microprocesador del caballo del invidente, y no tanto la amplitud de estas señales. Por ello, se procura emitir la primera señal con la mayor amplitud posible, para que llegue con una amplitud considerable al segundo microprocesador para que pueda ser detectada con facilidad. De la misma manera, la señal que se devuelve tiene también una amplitud considerable.

Finalmente, cuando la señal regresa al primer microprocesador se amplifica y se filtra, para poder trabajar con ella con mayor facilidad.

En resumen, la opción más adecuada para la medida que se requiere es utilizar sensores de ultrasonidos, añadiendo un paso intermedio en el caballo precedente.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Después de haber analizado el problema de este colectivo y de estudiar que en el mercado no existen soluciones acertadas para el mismo, se llega a la conclusión de que es necesario desarrollar un sistema de medición de distancias adaptado, partiendo de alguno de los circuitos de medida de distancias por ultrasonidos explicado anteriormente.

Se necesita un sistema sencillo de utilizar, a la vez que efectivo y preciso, para conseguir que la experiencia del usuario sea lo más placentera y segura posible. Con el sistema desarrollado a lo largo de estos proyectos se consigue que el invidente sea consciente en todo momento de su posición en la pista, respecto a las paredes y al caballo precedente. De esta forma, se proporciona al usuario la seguridad y tranquilidad de que su posición está controlada en todo momento asegurándole que mantendrá una distancia segura con respecto a los otros objetos de la pista.

El único requerimiento para el usuario es que se coloquen los sensores sobre el caballo que vaya a utilizarse y los actuadores en su cuerpo para que reciba bien las indicaciones del microprocesador, pero en partes distintas de su cuerpo para poder diferenciar su procedencia.

4.2 OBJETIVOS

Como se ha mencionado en el Capítulo 1. los objetivos del proyecto son:

- Desarrollo del sistema de medida de distancia con el caballo precedente usando ultrasonidos.
- En este sistema de ultrasonidos, creación de dos señales distintas para que los sensores no las confundan. Investigación de la posible confusión de las señales y, si se da el caso, codificación de las señales para su diferenciación.

- Cálculo del tiempo entre la emisión y la recepción en el caballo del invidente.
- Implementación de un sistema de aviso al jinete, que servirá para indicar si este debe variar la distancia que mantiene con el caballo precedente.
- Combinar el diseño de estos sensores con el de los sensores anteriormente desarrollados en una placa de circuito impreso conjunta.

Cada objetivo del proyecto se ha ido trabajando como una parte aislada, pero perteneciente al conjunto del sistema y partiendo de la base de que el punto anterior ya está desarrollado. De esta forma, a la hora de desarrollar el cálculo de la distancia, ya se dispone de los dos conjuntos emisor-receptor de ultrasonidos y de las dos señales de ultrasonidos que se transmiten de lado a lado. A medida que se consiguió un objetivo, se avanzó de manera gradual un paso más hacia el final del proyecto.

4.2.1 DESARROLLO DEL SISTEMA DE MEDIDA

En este objetivo se desarrolla un solo sensor de ultrasonidos, parecido al usado en el proyecto anterior por Miguel Ángel Sánchez, pero utilizando dos cápsulas de ultrasonidos en lugar de un solo componente.

Una vez montado un circuito con un emisor y uno con un receptor, se realizan pruebas para ajustarlos y mejorar su funcionamiento en conjunto. Después, se estudia la precisión de la señal medida, realizando medidas de varias distancias conocidas de antemano y viendo cómo varía la amplitud de la señal recibida en función de la distancia entre sensores.

Al finalizar este paso, se dispone de un sensor de ultrasonidos, con una cápsula emisora y una cápsula receptora, capaz de medir distancias de manera precisa hasta dos metros y medio aproximadamente.

4.2.2 CREACIÓN DE DOS SEÑALES

A continuación, se monta otro circuito igual al anterior, que ya se ha comprobado que puede emitir y recibir señales de ultrasonidos útiles para medir distancia. Se sitúa una pareja de emisor y receptor frente a la otra y se estudia cómo se reciben las señales emitidas por un

lado en el contrario y viceversa. De esta forma, se establece la comunicación entre los dos sensores que se van a colocar uno en cada caballo.

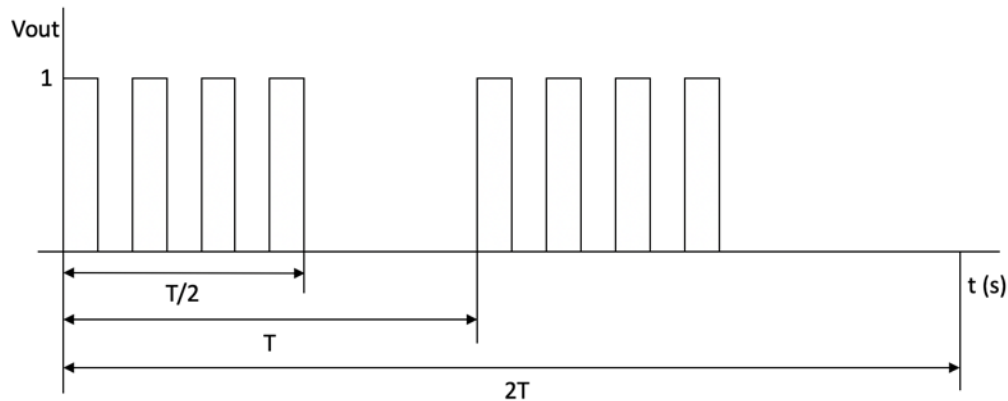


Figura 13: Ejemplo de señal PWM

Para ver cómo interactúan los circuitos, la señal de prueba que se envía de un lado a otro es una señal cuadrada con varios pulsos en '1' y el resto, la mayoría de ellos, en '0', de periodo

$$T = \frac{1}{40kHz} = 0,000025 \text{ segundos} \rightarrow 25\mu s, \text{ ya que las cápsulas funcionan a } 40 \text{ kHz.}$$

A esta frecuencia trabajan ambos sensores de ultrasonidos, de forma que, aunque las dos señales se emiten en tiempos distintos, tienen la misma forma y periodo.

En la Figura 13 se muestra un ejemplo de señal PWM en la que, durante medio periodo, la señal de salida oscila entre cero y uno y, durante el otro medio, permanece a cero. Al tratarse de una onda periódica, en el siguiente periodo la señal se reproduce de igual manera que durante el anterior. Las señales que se utilizan en el proyecto se emiten durante dos periodos con varios pulsos entre cero y uno y a continuación la señal se queda a cero para no confundir al receptor del otro lado.

4.2.3 CÁLCULO DEL TIEMPO

Cuando los dos circuitos son capaces de comunicarse entre sí, y se puede medir el tiempo que pasa desde que se emite en un microprocesador y hasta que se recibe una señal de vuelta,

se programa el cálculo de la distancia en el microcontrolador del caballo del invidente. Esta distancia depende del tiempo entre señales y de la velocidad del sonido en el aire.

4.2.4 SISTEMA DE AVISO

Después, una vez se tiene ya calculada la distancia en función del tiempo medido, hay que avisar al usuario en caso de ser necesario. Para conseguir realizar este objetivo simplemente se programa cuándo y cómo funcionará el actuador, de manera que vibrará con menor intensidad si se encuentra lejos del caballo precedente y con mayor intensidad si se encuentra demasiado cerca.

Para que el sistema no sea demasiado estricto, se acepta que la distancia es correcta mientras el caballo se encuentre dentro de un rango de distancias no muy alejadas de la distancia objetivo de 1.5 metros.

4.2.5 DISEÑO DE LA PLACA DE CIRCUITO IMPRESO (PCB)

Por último, se combina la parte del sistema diseñada en este proyecto de fin de grado con la diseñada en los proyectos anteriores. Para ello se diseña una placa de circuito impreso usando el programa KiCad.

Esta placa cuenta con los circuitos necesarios para situar los sensores que miden la distancia hacia las paredes, una pareja emisor/receptor en la parte delantera del caballo y otra pareja emisor/receptor en la parte trasera del caballo.

En resumen, se unen todos los sensores que se han desarrollado a lo largo de los dos proyectos de manera que con una sola placa y un solo microprocesador puede controlarse todo.

4.3 METODOLOGÍA

El proceso se ha dividido en cinco partes principales:

1. **Estudio del funcionamiento de los sensores:** Se investigó el funcionamiento del emisor y del receptor de manera separada ya que la mayoría de los sensores de ultrasonidos que se comercializan se venden juntos, de manera que su funcionamiento ya está estipulado de una manera concreta.
2. **Acople de dos parejas de sensores de ultrasonidos:** A continuación, se montaron dos circuitos de ultrasonidos y se enfrentaron, ajustándolos para que el receptor de un lado recibiera la señal emitida en el lado contrario.
3. **Programación de los microprocesadores:** Después, se programaron los dos microprocesadores, realizando el cálculo de distancia en el microprocesador del caballo del invidente y fijando cuándo serán emitidas las señales de cada receptor.
4. **Diseño de la PCB:** Se diseña una placa que fusione los sistemas de medida de distancia anteriores con este.
5. **Pruebas y ajuste del sistema:** Una vez montada la PCB se ajusta el sistema para que la medida de la distancia sea precisa y para que los distintos sensores funcionen de manera compenetrada.

Para poder realizar cada objetivo es necesario haber conseguido el anterior por lo que, a la hora de trabajar, se ha conseguido avanzar de manera gradual en el desarrollo del proyecto dividiéndolo en pequeños bloques.

De esta manera, a la hora de intentar conseguir nuevos avances se parte de la base de que lo anterior ya es correcto y sirve como pilar para el siguiente paso. Así, se simplifica el trabajo y se divide en pequeños bloques fáciles de desarrollar, que forman parte importante de todo el sistema final.

4.4 ESCENARIO DE USO

El sistema está diseñado para utilizarse para la práctica de la equitación en pista, en concreto de la doma. Cuando se practica esta disciplina se hace dentro de un recinto cerrado, de sesenta metros de largo por veinte metros de ancho.

Esta pista cuenta con una cerca alrededor que delimita el espacio, y que sirve de referencia para delimitar la distancia a la que se encuentra el caballo de los límites de la pista. Puede observarse un ejemplo en la Figura 14.



Figura 14: Pista de doma cercada [19]

La pista es rectangular por lo que el caballo avanza en línea recta mayoritariamente, a excepción de cuando dobla en las esquinas para cambiar de lado del recinto. Por ello, habría que realizar de manera diferente, con un sistema más depurado, la medida de la distancia entre el caballo principal y las paredes laterales y entre el caballo principal y el caballo delantero cuando el usuario se encuentre girando en una esquina.

Al ser el tiempo que el caballo se encuentra haciendo una esquina una pequeña fracción de tiempo comparada con el tiempo que avanza en línea recta, en este proyecto no se particulariza el funcionamiento del sistema para este caso.

De todas formas, cuando se practica la equitación, en concreto al practicar doma, el caballo ha de incurvarse en las esquinas, de manera que su cuerpo se redondee con el camino que sigue. Si el jinete realiza esta flexión de manera correcta, los sensores no van a dejar de comunicarse entre sí nunca.

La medida de la distancia lateral se realiza respecto a la cerca que delimita el recinto, dejando que la señal rebote en esa pared y vuelva al sensor. La medida de la distancia entre el caballo del usuario y el caballo precedente se hace manteniendo entre ambos caballos el mismo ritmo

de manera que si el jinete se ha separado, aumentando el paso puede recuperar su posición original y si se ha acercado demasiado disminuyéndolo puede recuperarla también.

El sistema en sí se sitúa sobre la montura del animal a utilizar, que en este caso pertenece a la categoría de pony C, debido a su altura que hasta la cruz es de 1.40 metros. Estos ponis son los segundos de mayor altura, dentro de la clasificación global. Los siguientes más altos, antes de considerarse caballos, son los ponis D que tienen una altura de hasta 1.48 metros medidos desde el suelo hasta la cruz. En la Figura 15 se muestran las medidas tomadas por Paula Melero del pony concreto para el que se desarrolla el sistema.

Estas medidas son importantes sobre todo a la hora de colocar los emisores y receptores de ultrasonidos. Si no estuviesen nivelados entre el caballo del usuario y el caballo precedente el sistema no funcionaría correctamente ya que las señales no llegarían de un lado a otro y no se recibirían.

También son importantes para calcular la longitud de los cables necesaria para conectar los diversos sensores y actuadores al microprocesador de la PCB. Estos cables no pueden quedar demasiado tensos ya que con cualquier movimiento del animal se romperían o desconectarían.

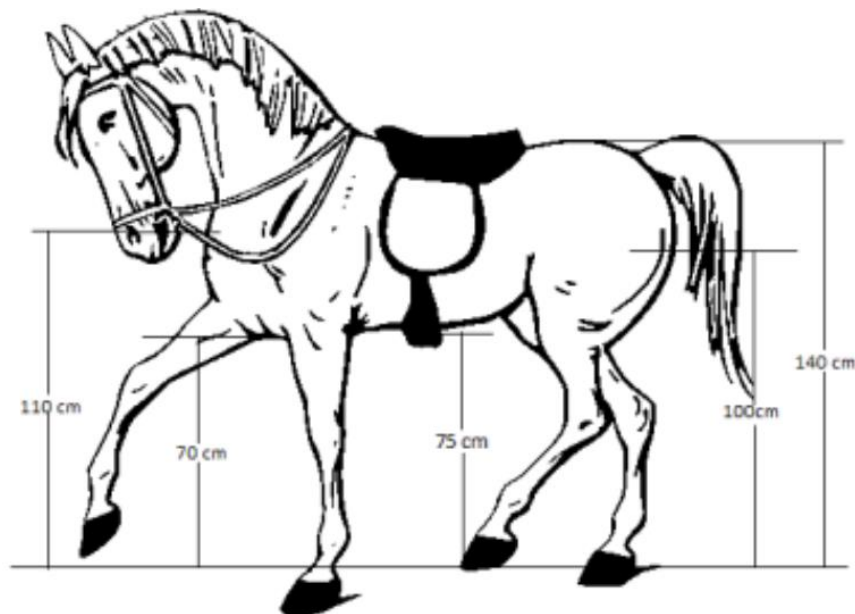


Figura 15: Medidas aproximadas del animal, tomadas por Paula Melero [1]

Estas medidas fueron realizadas por Paula al inicio de su proyecto en el Club Hípico Las Nieves, situado en la localidad de la sierra madrileña de Cerceda.

Aún así, el sistema podría adaptarse para colocarse en otros caballos de otras medidas con facilidad, cambiando sutilmente la orientación de los sensores y la longitud de los cables que los unen a la PCB, tal y como se hizo en el siguiente proyecto en el que las pruebas se realizaron utilizando un caballo.

4.5 IMPACTO MEDIOAMBIENTAL

A la hora de desarrollar cualquier proyecto es importante tener en cuenta el coste total del proyecto. Este coste se compone en su mayor parte por el coste de los materiales y la mano de obra necesarios para fabricarlo y programarlo, pero también por el coste que tiene este proyecto para el medioambiente. Por ello, en este apartado se va a describir el impacto medioambiental que genera este sistema desarrollado.

Al analizar las distintas partes que componen este proyecto nos encontramos principalmente con el software y el hardware. El software no genera en sí impacto medioambiental, ya que son programas que se están ejecutando dentro de componentes físicos pero que no son tangibles por ellos mismos.

En cuanto al hardware, mientras se esté utilizando el proyecto este no genera residuos ni emisiones nocivas para el medioambiente ya que sus componentes son electrónicos y no se involucran en ningún proceso de combustión de ningún tipo ni en otros procesos que generen residuos nocivos para el medioambiente.

Por lo tanto, el impacto ambiental que produce este sistema se daría al desecharlo. La mayoría de los componentes de los circuitos pueden reciclarse de manera sencilla y reutilizarse en otros proyectos, dándoles así una segunda vida a los mismos. Estos componentes que pueden reciclarse no generan apenas impacto ambiental. El problema serían las pilas que se utilizan para alimentar el sistema. Actualmente para alimentar el sistema completo se utilizan pilas alcalinas de 1.5 V.

La vida útil de estas pilas varía dependiendo del consumo del dispositivo, del uso que se le dé al mismo y de su conservación. Es un periodo que puede oscilar desde apenas semanas hasta meses o incluso años.

Cuando estas pilas se agotan lo habitual es desecharlas, llevándolas a un contenedor especial dedicado a su recogida. De ese contenedor pasan a puntos de reciclaje donde se aprovechan y reciclan para reutilizarse algunas de sus materias primas.

Si, por el motivo que fuese, esta pila no es tratada correctamente genera, un gran impacto para el medio ambiente ya que está llena de metales y sustancias químicas tóxicas, como mercurio o cadmio [20].

Una sola pila alcalina usada y mal desechada puede contaminar cantidades abundantes de agua, causando a las personas que la consuman diversos problemas, como por ejemplo de riñón o de pulmón, además también pueden contaminar el terreno donde sean desechadas y causar problemas en la fauna y la flora del entorno.

En definitiva, este sistema desarrollado produce poco impacto medioambiental por sí mismo, pudiendo contaminar por medio de las pilas que le suplen energía. Esta contaminación puede evitarse siempre y cuando se desechen correctamente las baterías con las que se alimenta.

4.6 IMPACTO SOBRE EL CABALLO

Los caballos pueden escuchar sonidos que se encuentren entre 55 Hz y 33.5 kHz aproximadamente. Este rango es muy superior al de los seres humanos que se encuentra entre 20 Hz y 20 kHz.

Como este sistema trabaja a 40 kHz no es de esperar que el caballo lo detecte y, si lo hiciera, sería de manera muy tenue, por lo que no va a molestarle ni a causarle ningún daño en su sistema auditivo. En el caso de que el caballo se notase molesto por las vibraciones de los sensores podrían modificarse los circuitos para trabajar a frecuencias superiores y así solucionar el problema de forma sencilla.

Capítulo 5. SISTEMA DESARROLLADO

En este capítulo se describe el funcionamiento del sistema desarrollado, separándolo en bloques que se explican en el orden en el que se desarrollaron.

En primer lugar, se analizó el funcionamiento de los sensores de ultrasonidos, seguido por la comunicación de dos parejas de sensores. Teniendo esta comunicación entre los sensores, se desarrolló la medida del tiempo entre la emisión y la recepción en la pareja de sensores del caballo del invidente para después poder usarla para calcular la distancia entre los dos caballos y compararla con la distancia de referencia. Por último, usando esta distancia calculada, se programó la interacción del actuador con el jinete.

5.1 SENSORES DE ULTRASONIDOS

En este proyecto se utilizan dos emisores y dos receptores de ultrasonidos, por lo que se montan dos sensores de ultrasonidos iguales.

Cada uno de ellos se situará en un caballo distinto, pero contarán con los mismos componentes y funcionarán de la misma manera, como se explicará a continuación. Lo que les diferencia es en qué parte del proceso de la medida de la distancia intervienen.

5.1.1 LOS CIRCUITOS MONTADOS

Para comenzar, se estudió el funcionamiento de una única pareja emisor-receptor de sensores de ultrasonidos. Para analizar el comportamiento de estos, se montaron dos circuitos, el circuito de la Figura 16 para el emisor y el circuito de la Figura 17 para el receptor y se realizaron diversas pruebas con el sistema montado.

Estos circuitos se conectaron a un osciloscopio para poder observar bien las señales de ultrasonidos con las que se trabaja.

En el circuito del emisor, entre la señal enviada por el microprocesador y la que recibe el transmisor se encuentra el transistor Q1.

La tensión que recibe esta cápsula de ultrasonido es cercana a 5 V lo cual se encuentra por debajo de su límite superior de trabajo, 20 Vpp. La tensión que recibe este transmisor se convierte en energía cinética, que mueve los componentes de la cápsula y produce una onda de ultrasonidos, que como era de esperar es inaudible para los caballos y para los humanos.

El circuito correspondiente al receptor de ultrasonidos cuenta con un amplificador, en este caso el MCP6002, como se observa en la Figura 17.

Este amplificador operacional se encarga de aumentar la tensión de la señal que se recibe que, al haber sido transmitida por aire y durante cierta distancia, pierde parte de la amplitud con la que salió del primer microprocesador.

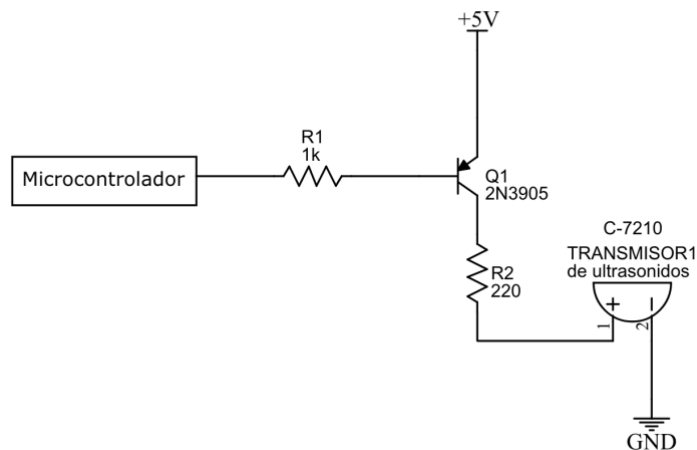


Figura 16: Circuito de los emisores de ultrasonidos

La onda después de pasar por esta etapa tendrá un valor máximo de 5 V, que es la tensión de alimentación positiva del operacional, para que pueda ser lo suficientemente amplia como para trabajar con ella, pero no demasiado como para dañar al microprocesador cuando llegue a este.

El microcontrolador funciona a 3.3 V, pero algunas de las entradas son tolerantes a 5 V. Por ello, los pines utilizados de entradas al microcontrolador son tolerantes a 5 V y que es la tensión máxima que tendrá la señal que recibirán.

Después de la etapa amplificadora, la señal pasa por un detector de envolvente que se queda solo con los valores máximos de la onda recibida y no con la parte de transición de la misma. De esta manera, se pasa de una señal muy oscilatoria a una más suavizada y fácil de manejar, que simplifica el trabajo posterior que se realiza con ella.

Al ser una onda muy oscilatoria, necesita el filtro para obtener solo los valores importantes de la señal. Por último, se ha incorporado un circuito de protección formado por dos diodos shottky para asegurar que la tensión de entrada en el microprocesador es igual o menor a la tensión máxima que soporta (5 V).

Si la tensión que llega a este divisor es mayor que la esperada, el diodo superior conducirá y desviará la tensión para que no pueda dañarse la entrada al microprocesador. De esta forma se protege el sistema de posibles sobretensiones.

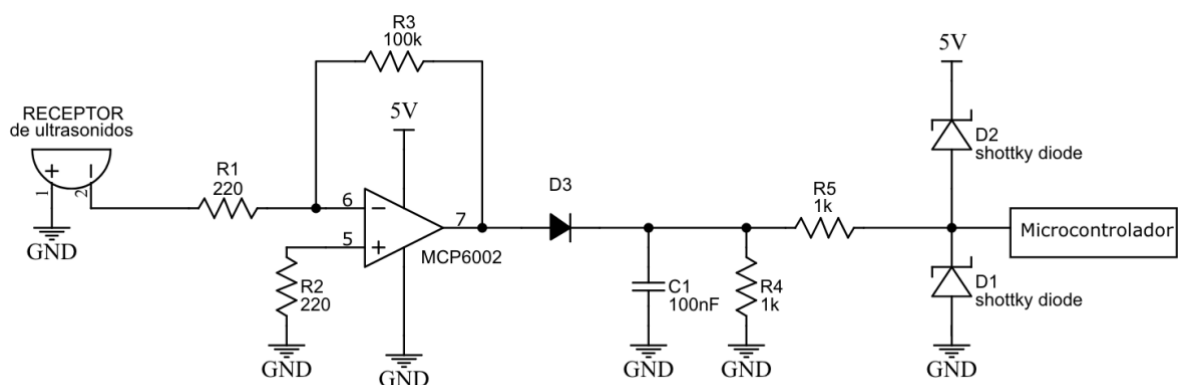


Figura 17: Circuito de los receptores de ultrasonidos

Una vez montados los circuitos, se situó un sensor enfrente del otro, separados por una distancia variable, que era medida en cada prueba que se realizaba como se observa en la Figura 18.

Al variar la distancia entre estos circuitos en cada ensayo y compararla con la medida de tiempo obtenida, se comprobó que el funcionamiento del conjunto era correcto, dando lugar a una desviación muy pequeña.

Esta comprobación se hizo con las dos parejas de emisor y receptor que se usan en el proyecto para comprobar que ninguna de las cápsulas estaba dañada y que todas funcionaban de la forma esperada.

En las diversas pruebas que se realizaron se comprobó también si el objeto en el que rebotaba la señal influía en la amplitud de la señal recibida, haciendo que rebotase en la pared, en la mano o en libros, pero la precisión del sistema era siempre la misma. De esta forma se comprobó que la precisión del sistema es muy buena, sea cual sea el medio al que tiene que llegar la señal para rebotar.

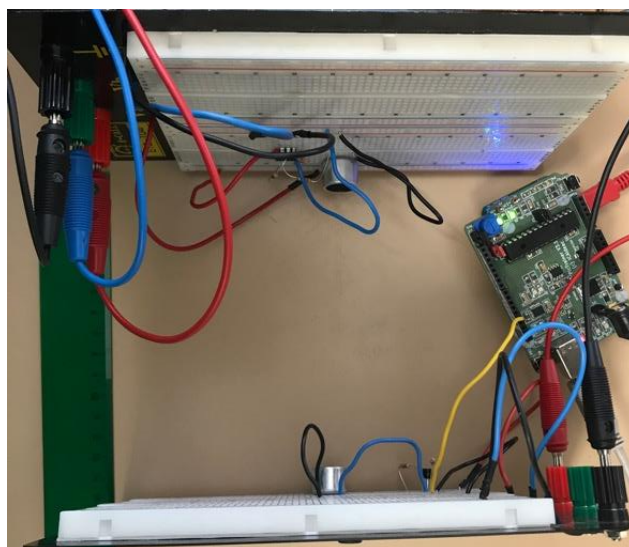


Figura 18: Sensores enfrentados

5.1.2 LAS SEÑALES DE LAS PRUEBAS

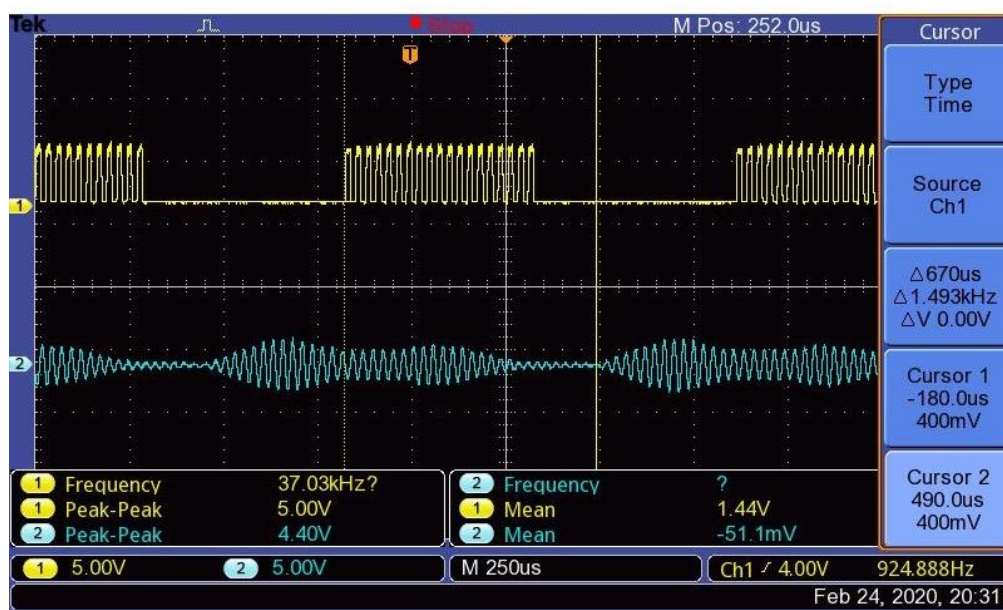
Como se observa a continuación en estas pruebas el microprocesador envía una onda cuadrada de varios pulsos entre cero y cinco voltios, lo que se corresponde con la onda amarilla. Esta señal alimenta el emisor, que convierte la energía eléctrica del circuito en energía mecánica que sirve para accionar su mecanismo y crear la señal acústica de

ultrasonidos de 40 kHz. La separación que se deja desde los diversos pulsos en la onda emitida, hasta que se vuelva a emitir en la misma, se hace para poder distinguir cuándo llega la señal al otro lado y no confundir distintas recepciones.

La onda azul de la Figura 19 se corresponde con la señal final que recibe el microprocesador desde el receptor. Esta señal es la onda final obtenida después de pasar por la etapa de amplificación, el filtro y el circuito de protección (limitador) explicado en el Capítulo 5. Cuando esta señal llega al microcontrolador, se estudia el tiempo que ha empleado la onda en ir y volver hasta el mismo midiendo la separación en el osciloscopio entre ambas señales.

El retraso de tiempo que se observa desde que empieza la onda emitida (amarilla), hasta que comienza la onda recibida (azul) se corresponde con el retardo que hay desde que la onda se emite en un extremo hasta que el otro lo recibe. Con esta medida temporal se puede calcular la distancia que separa ambos sensores.

Al aplicar el detector de pico al circuito, en lugar de simplemente un filtro, la onda que se obtendría sería solamente la forma exterior de la onda completa, sin las oscilaciones que se producen en la Figura 19.



TBS 1072B-EDU - 13:35:27 24/02/2020

Figura 19: Medida del tiempo entre la emisión de los pulsos y su recepción

5.2 COMUNICACIÓN ENTRE LOS SENSORES

La señal que se emite en ambos lados es la misma, a la misma frecuencia. Se trata de una señal cuadrada de varios pulsos, como se ha explicado más atrás. En el Capítulo 7. se ampliará la información sobre estas señales emitidas.

Pueden darse varios problemas a la hora de trabajar con ultrasonidos. A continuación, se van a describir los tres problemas que podrían darse y se explicará por qué no suceden en este sistema. Todas estas opciones han sido estudiadas para descartarlas y/o prevenirlas.

5.2.1 REBOTE DE LA SEÑAL EN UNA PARED

Podría darse el caso de que las señales emitidas salieran perpendiculares al caballo, de manera que se dirijan hacia la pared en vez de hacia el otro animal. Si bien es cierto que los componentes de ultrasonidos con los que se trabajan pueden emitir y recibir a 90° de su eje principal en ambas direcciones, la señal se emite en esa dirección con tan poca intensidad, que a su vuelta es prácticamente imperceptible.

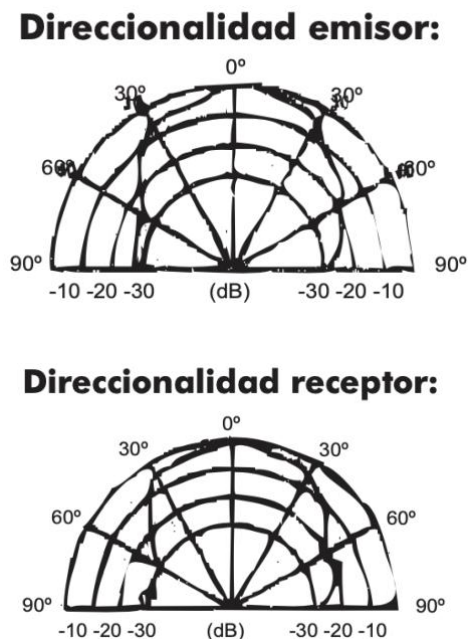


Figura 20: Direccionalidades de los componentes [21]

En la Figura 20 pueden observarse las direccionalidades de los componentes utilizados, destacando la disminución de los decibelios cuanto mayor es el ángulo de apertura.

5.2.2 REBOTE DE LA SEÑAL EN EL OTRO CABALLO Y REGRESO AL PUNTO DE PARTIDA

Al ser sensores de ultrasonidos tienen un alcance limitado, en este caso según lo indicado por el fabricante de cuatro metros en el caso de trabajar con señales de 20 voltios pico. Esto quiere decir que en el caso de contar con una señal de veinte voltios podría medir distancias de cuatro metros.

En el caso de este proyecto el voltaje utilizado es menor, por lo que la distancia máxima posible que se puede medir es menor. Los caballos se encuentran a un metro y medio aproximadamente entre sí, por lo que, si la señal tuviera que ir, rebotar y regresar debería viajar durante al menos tres metros, lo que la convertiría en una señal altamente debilitada.

Por ende, no se va a dar el caso en el que la señal emitida en un lado regresa a la vez que el otro microprocesador emite su señal.

5.2.3 CONFUSIÓN Y/O MEZCLA DE LAS SEÑALES

Como las señales no van a estar emitiéndose a la vez, es imposible que se mezclen entre sí y se confundan o distorsionen de cualquier manera por la presencia de otras ondas de las mismas características.

Si se hubieran emitido a la vez podría haberse dado el caso en el que las señales se mezclasen, pero al desarrollar la idea de este proyecto se decidió que no se emitiese a la vez para evitar esto mismo.

5.2.4 PÉRDIDA DE LA SEÑAL

En el caso de que los sensores se desalinearan, la señal seguiría recibándose debido a la alta direccionalidad con la que cuentan tanto el emisor como el receptor.

Aún así, si de manera puntual se perdiera alguna señal y no se produjera la comunicación correcta de la misma, el microprocesador del caballo del invidente está programado para volver a empezar el ciclo de emisiones y recepciones de señales si pasan diez milisegundos sin recibirse ninguna señal en el microprocesador del invidente después de haber emitido la suya.

Esto se hace con una señal que va contando el tiempo que transcurre desde que se emitió y mientras que no se recibe nada. Esta cuenta se resetea si se recibe una señal o si tiene que volver a emitirse la primera señal porque se ha perdido la anterior.

De esta manera se asegura que siempre va a haber un respaldo para la primera señal enviada.

5.3 MEDIDA DEL TIEMPO

Para calcular la distancia entre la emisión y la recepción de los ultrasonidos, en función del tiempo medido entre estas dos señales, se necesita saber que la velocidad del sonido en el aire es $v_{sonido,aire} = 343.2 \text{ m/s}$.

La relación entre este tiempo y la velocidad del sonido en el aire es lineal, como se observa en la Figura 21.

Por lo tanto, a mayor tiempo transcurrido entre ambas señales, mayor separación habrá entre los sensores de ultrasonidos. Esta relación lineal se rige por la Ecuación 1. En esta ecuación se multiplica por un medio ya que el tiempo es de ida y vuelta y se necesita solo el de ida para no obtener el doble de la distancia real.

Ecuación 1: Cálculo de la distancia en función del tiempo

$$Distancia [m] = 0.5 \cdot v_{sonido,aire} \left[\frac{m}{s} \right] \cdot \Delta tiempo [s] = 0.5 \cdot 343.2 \cdot \Delta tiempo$$

La distancia idónea es un metro y medio, pero se ha programado el microcontrolador de manera que cualquier distancia que esté en el rango de entre 1.3 metros y 1.7 metros será aceptada como buena.

Si solo se hubiese aceptado la distancia de 1.5 metros el sistema habría sido demasiado restrictivo y habría exigido al usuario una precisión en su posición inalcanzable en cualquier caso. Además, el sistema habría hecho que el actuador avisase todo el rato al usuario de que variase su distancia por lo que podría llegar a ser molesto para el jinete. Para prevenir esto se acepta el rango de distancias anteriormente mencionado.

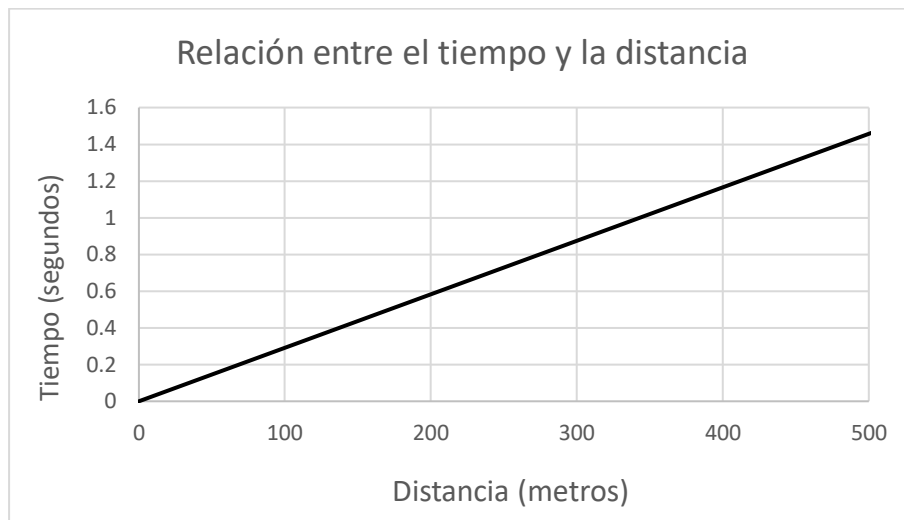


Figura 21: Relación entre el tiempo y la distancia

Es importante tener en cuenta que estos sensores de ultrasonidos miden como mucho 4 metros si trabajan con la tensión máxima que soportan por lo que nunca se trabajaría con tiempos mayores de 11.68 milisegundos en este caso. Como ejemplo, pueden verse en la Tabla 1 algunas distancias medidas y el tiempo de retraso que conllevan.

Tabla 1: Ejemplos de distancias y su relación con el tiempo

Tiempo de retraso	Distancia entre caballos
1 μ s	0.03432 cm
10 μ s	3.432 cm
1 ms	34.32 cm
10 ms	3.432 m
1 s	343.2 m

5.4 ACTUADOR

El actuador que se emplea es un vibrador, que aumenta la vibración si la distancia a la que se encuentran los caballos es demasiado cercana y la disminuye si se encuentran demasiado lejos. Se ha elegido el mismo actuador que se usó en el proyecto de fin de grado de Miguel Ángel Sánchez, para mantener la coherencia entre los actuadores.

En función de la intensidad de la vibración se distingue la posición del jinete respecto al caballo que le precede. Además, los actuadores que indican las distancias laterales se sitúan en las piernas del jinete por lo que este nuevo actuador puede situarse en la mano y no causa confusión con los otros dos. También podría situarse en el vientre del jinete, para lo que se necesitarían menos cables.

La intensidad del actuador se controla variando el ciclo de trabajo de este, pero manteniendo dentro de un límite para no superar la tensión máxima de alimentación que soporta el vibrador.

5.4.1 EL CIRCUITO MONTADO

La tensión máxima a la que pueden funcionar estos vibradores es de 3 V y necesitan ser accionados por una PWM, que en función de su ciclo de trabajo creará una vibración de mayor o menor intensidad en cada caso.

Esta intensidad se regula mediante la programación del microprocesador, en concreto de la parte encargada del vibrador. Como este ciclo de trabajo depende de la posición, podría darse el caso de que fuese mayor que el soportado por el componente por lo que se ha programado el actuador para que en el caso de que el ciclo de trabajo que se le asigne sea superior a su máximo lo ignore y use su ciclo de trabajo máximo.

En el circuito de la Figura 22 se coloca un diodo en paralelo con el actuador para evitar que la fuerza electromotriz inducida por el motor del vibrador afecte al resto del circuito, este diodo se conoce como diodo de libre circulación. Como se ha comentado antes, la tensión

máxima que soportan es de 3 V, pero se alimentan 12 V, controlando con la PWM el ciclo de trabajo para no sobrepasar la tensión máxima que soportan.

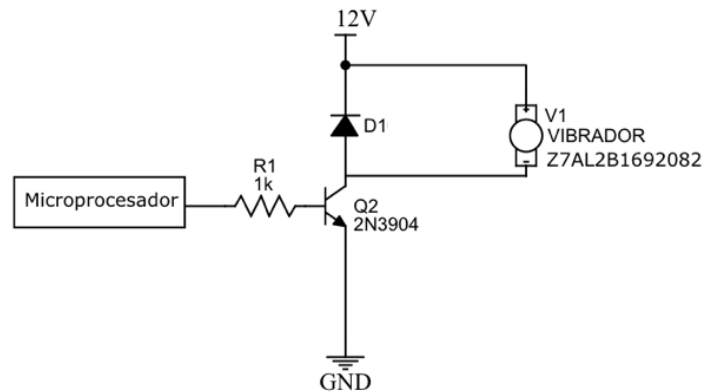


Figura 22: Circuito montado para el actuador

5.5 ALIMENTACIÓN

Al igual que en el proyecto anterior, este sistema se alimenta mediante ocho pilas de 1.5 V en serie, de manera que se obtienen al final 12 V. También se cuenta con una toma de 3 V, que adaptó en el sistema anterior Miguel Ángel Sánchez.



Figura 23: Ejemplo de portapilas [22]

Debido a esta tensión de alimentación, los circuitos diseñados en primer lugar con los sensores de ultrasonidos tuvieron que ser adaptados para su correcto acople con el sistema anterior. En primera instancia se usaba otra tensión de alimentación en el amplificador operacional del circuito del receptor de ultrasonidos.

Además, se pensaba alimentar el operacional de manera simétrica para que su V_{cc} tuviera el mismo módulo que su $-V_{cc}$. No ha sido posible hacerlo así por la alimentación disponible para el sistema, pero no ocasiona ningún problema o cambio más allá de tener que adaptar los circuitos principales.

El problema de esta fuente de alimentación es que ocupa demasiado espacio en el diseño, como puede observarse en la Figura 23. La placa de circuito impreso está optimizada para tener menor tamaño y por ende ocupar menos espacio, pero estas pilas no. Podrían sustituirse por otras de menor tamaño, por ejemplo, una batería tipo LiPo o pilas AAA, en proyectos posteriores.

Capítulo 6. DISEÑO DE LA PLACA DE CIRCUITO IMPRESO

En este capítulo se diseña una placa de circuito impreso (PCB) en la que se ponen en común los circuitos desarrollados en este proyecto con los desarrollados por Miguel Ángel Sánchez en el proyecto anterior.

De esta manera, la PCB cuenta con los circuitos necesarios para el funcionamiento de los sensores laterales, la pareja emisor/receptor que se colocará en la parte delantera del caballo y la pareja emisor/receptor que se colocará en la parte trasera del mismo.

Para el diseño de esta PCB se parte del prototipo final desarrollado por Miguel Ángel Sánchez, de manera que a esa PCB se le añaden los nuevos circuitos desarrollados en este trabajo, acoplándolos y redistribuyendo los componentes en la placa para que el tamaño de esta sea el menor posible.

Algunos de estos circuitos desarrollados en los cursos anteriores han sido adaptados ya que su diseño no era del todo correcto. Estas adaptaciones son mínimas (cambios de algunos divisores de tensión que no proporcionaban la tensión esperada y cambio del valor de las tensiones de dos pines que estaban intercambiadas entre sí y por ello mal indicadas) por lo que no afectan al desarrollo del proyecto. Por otra parte, es necesario destacar que en los planos desarrollados en el proyecto anterior se indica que se cambia el pin RB2 por el RB11, por lo que este debe ser cambiado en los códigos del sistema.

El proceso seguido para desarrollar esta placa se muestra en la Figura 24. En primer lugar, se descargaron las librerías necesarias para tener todos los componentes de los circuitos. Aquellos componentes que no se encuentran en librerías ya existentes fueron diseñados.

En segundo lugar, se creó un plano con los circuitos de todo el proyecto. En este plano se corroboró que todas las conexiones eran correctas y no había ningún punto suelto. A continuación, se crearon las huellas de los componentes y la lista de las conexiones. Por último, se diseñó la placa en sí, consiguiendo los circuitos diseñados.

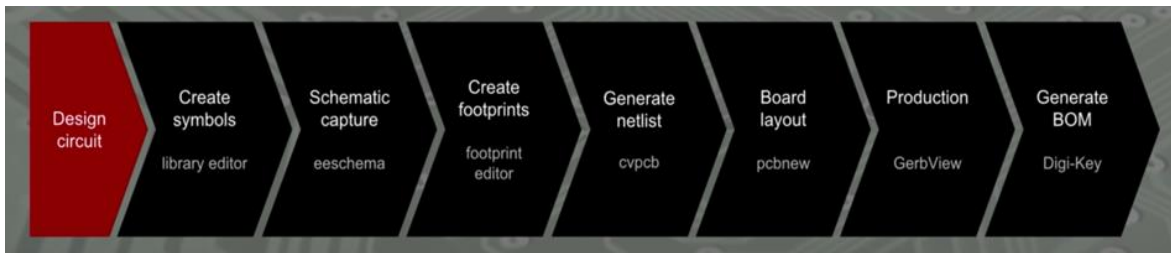


Figura 24: Proceso seguido para desarrollar la placa [23]

6.1 DISEÑO DE LOS CIRCUITOS

En primer lugar, se desarrolló el diseño de los circuitos que iban a incluirse en la PCB. Se diseñaron tanto las conexiones de los distintos componentes entre sí como las conexiones de estos circuitos y componentes con el microprocesador.

Para ello, se recuperaron los circuitos desarrollados por Miguel Ángel Sánchez en el proyecto anterior y se acoplaron con los desarrollados en este proyecto.

Estos circuitos anteriores no sufren modificaciones en cuanto a sus conexiones con el microprocesador ya que no necesitan usar los mismos pines del microprocesador que se utilizan en este proyecto.

Sí sufren alguna modificación, como se indicó anteriormente, sobre alguno de los divisores de tensión que no estaban diseñados de manera correcta y con respecto a las patillas del chip CD4010 ya que la alimentación que tenían dos de ellas no era la correcta.

También se conservan las anotaciones realizadas por Miguel Ángel en su plano, referentes a la naturaleza de los componentes utilizados y a aspectos para tener en cuenta a la hora de soldar los componentes.

Estos circuitos se capturan usando la aplicación *Eeschema* de KiCad. Para incluir los componentes en el esquema se han descargado varias librerías de componentes de KiCad, disponibles en su página web oficial. En estas librerías se encontraban ya diseñados distintos

componentes como por ejemplo resistencias, potenciómetros, fuentes de tensión y corriente, condensadores y diodos de varios tipos.

Otros componentes, como el emisor y el receptor de ultrasonidos o el microprocesador se han diseñado de manera independiente ya que no se encontraban disponibles en estas librerías descargadas. Para diseñarlos se ha utilizado la aplicación *Symbol Editor* de KiCad.

Estos diseños pueden observarse en el primer plano disponible en el “ANEXO IV: Diseño de la Placa”.

6.2 CONEXIÓN DE LOS SENSORES Y VIBRADORES

Para conectar los sensores y vibradores del sistema completo se han usado dos conectores DB9, de manera que dos conectores hembra van unidos a la PCB y dos conectores macho van unidos a los diversos componentes.

En una pareja de conectores se encuentran los tres sensores del proyecto anterior y los dos vibradores de este y en la otra pareja los dos emisores, los dos receptores y el vibrador de este proyecto.

De esta forma, los sensores y vibradores pueden unirse mediante cables a la caja donde se encuentra el microprocesador y colocarse en su posición correspondiente sobre el caballo.

Las conexiones entre los DB9 y los sensores y vibradores pueden observarse en el segundo plano disponible en el “ANEXO IV: Diseño de la Placa”.

Aun siendo los elegidos para este proyecto, hay mejores opciones que conectores de tipo DB9 como por ejemplo los conectores circulares estándar de Amphenol [31].

Los conectores de Amphenol aportan robustez al sistema ya que al ir roscados no se pueden desconectar de manera accidental durante el desarrollo de la actividad, al contrario que los DB9 que son más sensibles y pueden soltarse con mayor facilidad si se les da un golpe o sufren un movimiento brusco.

Estos nuevos conectores se descartaron por la dificultad a la hora de colocarlos (la mayoría de los conectores exigían colocarse de forma vertical y es imposible por cómo está diseñada la caja dentro de la cual va la PCB) y por su elevado precio, que se salía del presupuesto final pensado para este sistema.

6.3 CREACIÓN DE LAS HUELLAS DE LOS COMPONENTES

A continuación, se crearon las huellas de los componentes que no estaban disponibles en librerías preexistentes. Para crear estas huellas se usó *Footprint Editor* de KiCad.

Es un proceso algo tedioso por lo que se recomienda buscar librerías que ya cuenten con las huellas de los componentes que van a utilizarse de antemano.

6.4 GENERACIÓN DE LA LISTA DE CONEXIONES

Para generar la lista de conexiones se asoció cada componente del esquema con su huella correspondiente, de manera que se le añade a cada componente del circuito sus medidas reales y sus pines en las posiciones correspondientes, para después poder diseñar la placa conociendo las proporciones reales de cada parte.

6.5 DISEÑO DE LA PLACA

Después se hizo el diseño de la placa, situando todas las huellas de todos los componentes en la misma. Estos componentes estaban unidos entre sí por líneas de color blanco, llamadas *ratnest*.

Estas líneas representan la conexión que existe entre las rutas que más tarde se situarán en la placa y los componentes. Muchas de estas líneas se cruzan entre sí, aunque se intenta que sea el menor número posible para facilitar el trabajo posterior a la hora de colocar las rutas.

En la Figura 25 puede observarse la placa que se ha diseñado para este proyecto en uno de los pasos intermedios de su diseño. Como se ha explicado anteriormente, unas líneas blancas conectan todos los componentes entre sí.

Una vez se tuvieron todos los componentes situados, separados entre sí y con el menor número de cruces posibles entre las líneas blancas, se procedió a conectar los elementos. En la Figura 26 puede verse parte de la lista de elementos que había que conectar.

También puede observarse en la Figura 26 que se había realizado una comprobación de los elementos situados en el plano y que no había ningún error porque estuvieran demasiado cerca unos de otros. En el caso de que hubieran estado demasiado cerca entre sí o si hubiera dos componentes solapándose darían lugar a cortocircuitos y la placa no funcionaría de la manera deseada.

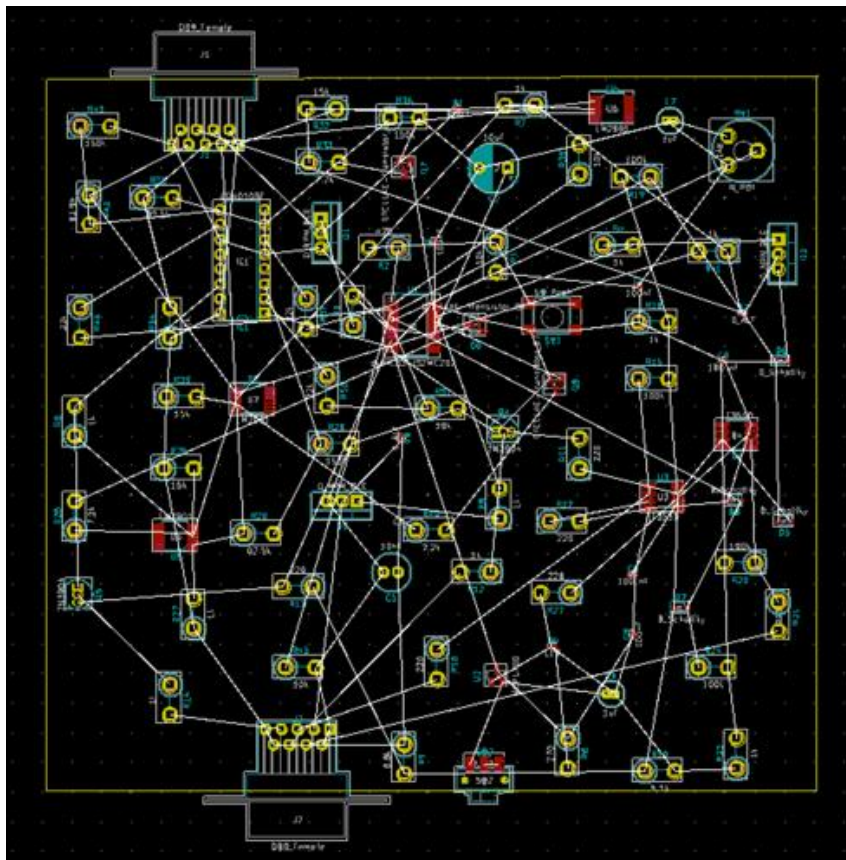


Figura 25: Una de las primeras versiones de la placa en el proceso de diseño

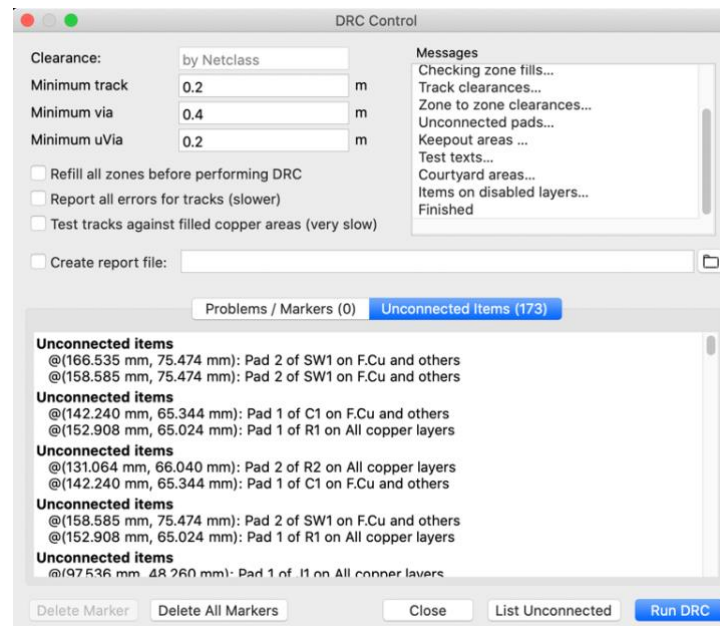


Figura 26: Elementos pendientes de conectar

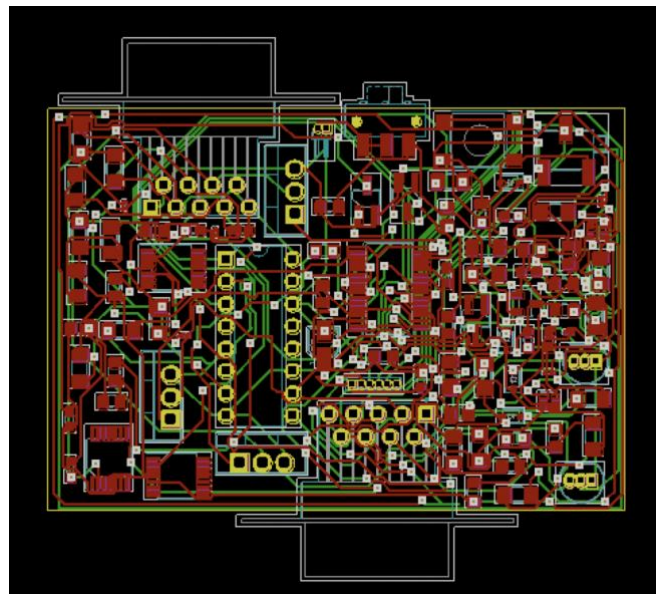


Figura 27: Conexiones finales de la placa

6.6 PRODUCCIÓN DE LA PLACA

Para la producción de la placa se adjuntan en el “ANEXO IV: Diseño de la Placa” los planos necesarios para la misma.

Hay dos tipos de planos, los que muestran diferentes partes del conjunto de manera individual y los que muestran todo el conjunto completo.

Los componentes se diferencian en dos grupos principalmente, los que se sitúan en la placa por orificio pasante y los que se sitúan soldándose a la cara superior de la PCB por lo que hay dos planos diferenciados en los que se separan unos de otros.

También se adjuntan dos planos mostrando las conexiones que se realizan en la cara superior y las que se realizan en la cara inferior de la PCB.

6.7 ASPECTO FINAL

Una vez se ha diseñado la placa de circuito impreso, con todos los circuitos y componentes necesarios, al mandarla a fabricar se obtendría una placa similar a la de la Figura 28, aunque de mayor tamaño ya que tiene más componentes y conectores que esta, que es la placa final del proyecto anterior. Durante el desarrollo de este proyecto no se ha podido fabricar la placa por el cierre de las instalaciones de la universidad y las restricciones laborales derivadas del estado de alarma provocado por el COVID-19.



Figura 28: Placa final del proyecto anterior [1]

Finalmente, se introduce la placa junto con la petaca de las pilas que sirven de alimentación en una caja metálica para que el sistema no sufra golpes o daños durante su uso.

De esta manera, el usuario solo tiene acceso a los conectores DB9, el botón de encendido y apagado y el de reseteo del sistema. A estos conectores DB9 se le unen los componentes mediante cables largos que permiten que se sitúen los sensores y vibradores en la posición para la cual han sido diseñados.

Puede observarse en la Figura 29 el aspecto que tendría el sistema final, con la caja metálica y los cables para unir los componentes necesarios.

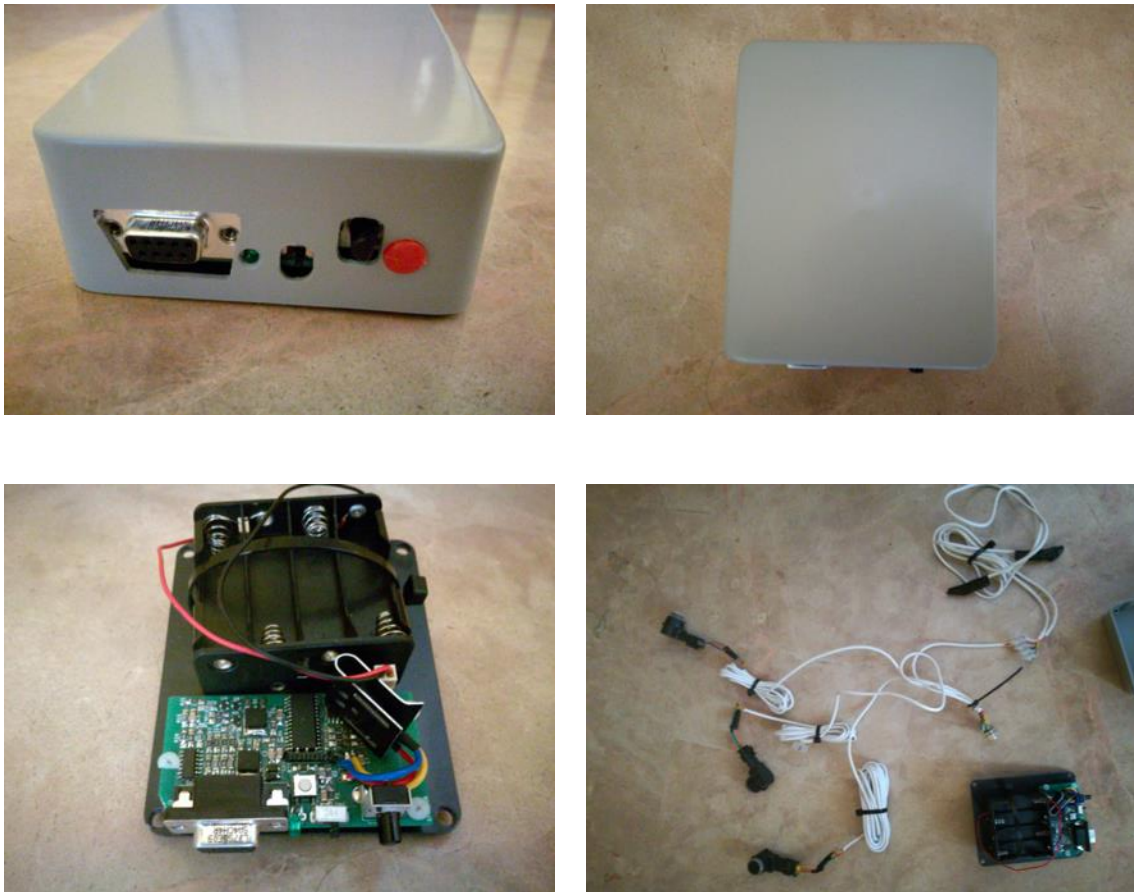


Figura 29: Aspecto final del diseño del proyecto anterior [1]

Capítulo 7. CÓDIGO DEL SISTEMA

En este capítulo van a explicarse los aspectos más destacables del código del sistema. Al trabajar con dos microprocesadores, han de programarse los dos por lo que se va a describir el funcionamiento de cada código de manera separada, usando en cada parte un diagrama de estados para explicar la estructuración de las actividades que realiza cada microprocesador.

Finalmente, también se unifican todos los códigos en un solo microprocesador, para unificar todo el sistema en una sola PCB.

7.1 PUERTOS USADOS EN ESTE PROYECTO

En este proyecto, los pines utilizados para conectar los distintos componentes a los microprocesadores son:

- **Emisor delantero:** RB4
- **Emisor trasero:** RB5
- **Receptor delantero:** RB0 (AN2)
- **Receptor trasero:** RB1 (AN3)
- **Actuador:** RB9 (PWM2L1)

7.2 MICROPROCESADOR DEL CABALLO DEL INVIDENTE

Este microprocesador se encarga, además de emitir y recibir una señal, de la medida de la distancia entre ambos caballos. Por ello, su diagrama de estados cuenta con tres estados distintos.

En primer lugar, como se puede leer en la Figura 30, se emite la primera señal para que sea recibida en el otro lado del sistema. A continuación, cuando se recibe la señal que llega de vuelta, se pasa a calcular el tiempo transcurrido desde que se ha emitido la primera señal

hasta que se ha recibido la segunda. Finalmente, antes de regresar al estado principal de emitir la señal, se indica con el actuador si es necesario corregir la distancia entre los caballos o no.

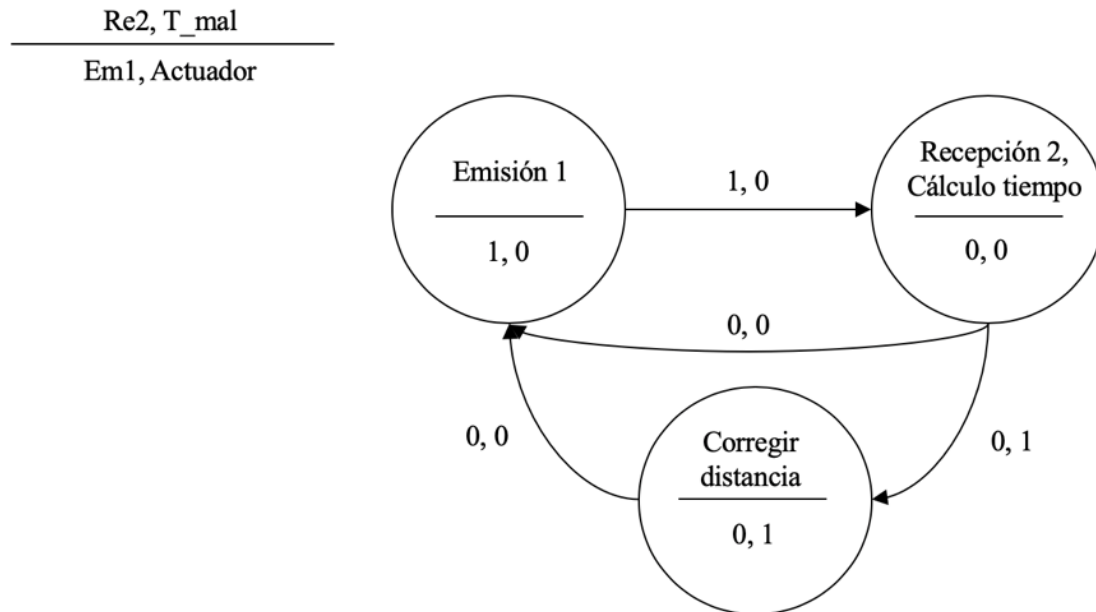


Figura 30: Diagrama de estados del microprocesador del invidente

Además, si pasaran más de diez milisegundos sin recibirse una señal de vuelta se emitiría de nuevo la primera señal, suponiendo que la anterior se ha perdido por el camino. No es de esperar que se pierda ninguna señal, pero al tratarse de ondas con un periodo de veinticinco microsegundos, dejar un margen de seguridad de diez milisegundos antes de emitir de nuevo garantiza que la medida va a seguir siendo precisa y aporta más seguridad al sistema.

En este estado del sistema, el microprocesador pasará más tiempo en el segundo estado, esperando a recibir la señal desde el otro lado. El cálculo del tiempo se realiza también de manera rápida en este estado.

A la hora de seleccionar los puertos de entrada y salida que se van a usar para el sensor desarrollado en este trabajo, se tienen en cuenta los puertos que usó Miguel Ángel Sánchez para su proyecto, de manera que no se solapen. Como ambos proyectos se van a fusionar en

la PCB no pueden usarse los mismos puertos, no sería coherente y el sistema final no funcionaría de manera correcta.

En el programa principal de este microprocesador, primero se pone el vibrador a cero, para que no interfiera con el resto de los componentes. A continuación, se envía la primera señal y se comienza a comprobar si ha llegado alguna señal al receptor. Cuando llega la señal, se calcula la distancia entre los dos caballos en función del tiempo (en milisegundos) que ha pasado desde que se emitió la primera señal hasta que ha llegado la más reciente.

```
int calcular_distancia(int tiempo) {  
    int distancia_cm = 0;  
    distancia_cm = 340*tiempo/100000/2; //Distancia en centímetros  
    return distancia_cm;  
}
```

Extracto de código 1: Cálculo de la distancia. Extracto de tiempo.c

Una vez se ha calculado la distancia a la que se encuentran ambos caballos, se avisa al usuario si es necesario. La distancia será considerada correcta si se encuentra entre 1.3 y 1.7 metros para que las señales mandadas al usuario no sean confusas, como se explicó más atrás. De esta forma se proporciona al jinete un rango de distancias en el que su posición se considera acertada y no se le informa de que debe cambiarla.

Si el jinete no está dentro de ese rango de distancias, se avisa al usuario mediante vibraciones de distinta intensidad. El programa que se encarga de las acciones del actuador es una modificación del programa que usó Miguel Ángel Sánchez en su proyecto, de manera que se han adaptado varias funciones para su uso en este caso.

El vibrador se encuentra en el pin RB9 por lo que depende de la señal PWM2L1. Las funciones de este actuador se programan usando el módulo PWM del microprocesador.

En caso de que se tenga que avisar al jinete con vibraciones, durante 2 segundos la acción que realizará el microprocesador es la de avisar al jinete, sin tomar medidas. Es decir, durante dos segundos solo actuará el vibrador para que el usuario tenga tiempo de reaccionar a la

señal y procesar cómo debe cambiar su posición. Una vez acabado este aviso se retoma la medición de tiempos y el cálculo de la distancia con el otro caballo.

Además, se ha añadido un contador para un `tiempo_seguro`. Este contador se encarga de asegurar que hay intercambio de señales antes de los 10 milisegundos (3.14 metros). Si no se produce la llegada de ninguna señal en este tiempo, se asume que la señal anterior se ha perdido y se envía otra para que el ciclo comience de nuevo.

Esto sirve para hacer que la comunicación sea más segura de forma que, en el caso de que alguna señal se perdiese, el microprocesador no se encuentre en estado de espera siempre sin llegar a recibir nada.

```
if(tiempo_seguro>10000){  
    enviar_senal_delante();  
    tiempo = 0;  
    tiempo_seguro = 0;  
}
```

Extracto de código 2: Extracto del main.c

Como se observa en el Extracto de código 2, si se supera este tiempo límite y se envía de nuevo la señal, se ponen a cero los valores de `tiempo` y `tiempo_seguro`. La primera variable se encarga de contar el tiempo que pasa desde que se emite la primera señal hasta que se recibe la segunda en este microprocesador.

Una vez se ha medido ese tiempo, se usa su valor para calcular la distancia entre los caballos. La segunda señal es la que lleva la cuenta de seguridad para cerciorarse de que no se pierden las señales.

Ambas señales se incrementan usando un contador de 1 microsegundo. Esto facilita el cálculo de la distancia. Por ello, el valor límite de `tiempo_seguro` es 10.000 microsegundos, es decir, 10 milisegundos. Una vez se ha alcanzado este tiempo, se envía nuevamente la primera señal, y se ponen a cero ambas cuentas. Así, se vuelve a tomar

la medida del tiempo que ha pasado desde que se ha emitido hasta que se ha recibido alguna señal en este microprocesador.

7.3 MICROPROCESADOR DEL CABALLO PRECEDENTE

Este sistema se sitúa en la parte trasera del caballo precedente y sirve como paso intermedio en la transmisión de señales entre los dos caballos. Así, aporta seguridad al sistema completo, asegurando que la señal no se debilitará demasiado ni se perderá al rebotar, sino que volverá a emitirse desde este lado a su máxima amplitud y bien direccionada.

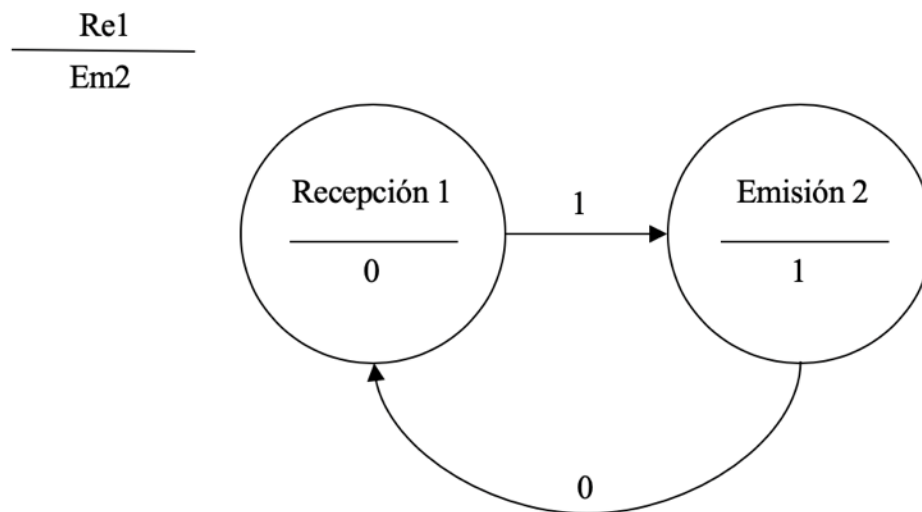


Figura 31: Diagrama de estados del microprocesador auxiliar

Este microprocesador se dedica a tareas sencillas dentro del sistema completo. Simplemente debe enviar una señal, compuesta por varios pulsos y seguida por un periodo a '0', cuando recibe otra señal.

Como se puede observar en la Figura 31, este microprocesador solo tiene dos estados. Uno en el que espera recibir algún dato desde el otro lado y otro en el que, después de recibir esos datos, responde brevemente. En caso de no recibir ninguna señal permanecería todo el tiempo en el estado "Recepción 1", llamado así porque recibe la primera señal de las dos que

se emiten en el sistema. Cuando se produjera la recepción, pasaría a emitir y luego volvería a esperar otra señal nueva.

La señal que se va a recibir es una señal analógica. Esta onda analógica, después de pasar por el módulo AD del microprocesador, se recibe como un entero entre 0 y 1023. Por esto al programar el módulo AD del microprocesador se fija un valor como límite inferior, a partir del cual se considera que se está recibiendo la señal del otro microprocesador y no ruido. En este caso ese valor umbral se ha fijado en 200.

Todas las señales que tengan un valor mayor de 200 se consideran enviadas desde el otro lado y desencadenan la emisión de la señal de vuelta. Si las señales que se reciben son menores, se consideran ruido y se descartan por lo que no producen ninguna emisión nueva desde este lado del sistema.

```
while(1) {  
    if(get_ad_detras(3) >= 200) {  
        init_emisor_detras();  
    }  
}
```

Extracto de código 3: Extracto del main.c

Una vez se ha recibido la señal desde el otro microprocesador, se devuelve otra similar a la que se mandó desde el microprocesador del caballo del invidente. Es una señal PWM, que durante dos ciclos de trabajo (de duración la mitad del periodo), crea pulsos entre '0' y '1'.

```
void __attribute__((interrupt, no_auto_psv)) _TlInterrupt(void) {  
    if(contador2 < 50) { //El periodo de 40 kHz es 25 us, 50 son dos ciclos  
        IFS0bits.T1IF = 0;  
  
        if(contador < DC) {  
            PORTB |= 0x20;  
        } else {  
  
            PORTB &= 0xDF;  
  
            if(contador > (periodo - 1)) {  
                contador = 0;  
            }  
        }  
    }  
}
```

```
    }  
    }  
    contador++;  
    contador2++;  
}  
  
if(contador2 > 50){  
    contador2 = 0;  
    contador = 0;  
}  
}
```

Extracto de código 4: Extracto del emisor_trasero.c

Como se puede observar en el Extracto de código 4 mientras el tiempo sea menor que el del ciclo de trabajo (DC) y que el periodo de emisión de la señal ($50\ \mu\text{s}$), la señal de salida se alterna entre cero y uno cada vez que se cuenta un microsegundo. De esta manera se crean los pulsos que son detectados en el microprocesador del otro lado.

7.4 COMBINACIÓN DE LOS PROYECTOS

Dando por hecho que en un futuro se situarán varios caballos unos detrás de otros, de manera que en uno mismo haya un sensor en la parte delantera y otro en la trasera, se ha desarrollado también un código conjunto en el que se encuentran tanto las funciones del antes llamado “microprocesador del caballo del invidente” como las funciones del “microprocesador del caballo delantero”.

En este código se añade además el realizado por Miguel Ángel Sánchez, de manera que pueden controlarse todos los sensores desarrollados (delantero, trasero y laterales). En el anexo se añaden en el `main` las partes del código desarrolladas por Miguel Ángel Sánchez que requieren adaptaciones y en un apartado distinto los programas `tmr.c` y `tmr.h` que han sido adaptados para poder acoplar de manera correcta los diseños, el resto de los programas continúan igual que los que fueron anexados en la memoria de su Proyecto Fin de Grado [1].

Los cambios más relevantes a la hora de juntar los dos proyectos son referentes a la organización del conjunto. Se han cambiado nombres de archivos y de funciones para que se diferencien mejor de otros similares, además de reorganizar los timers para sacarles más provecho. De esta manera, se reaprovechan recursos del proyecto de Miguel Ángel Sánchez para medir el tiempo que se tarda desde que la señal se emite hasta que se recibe alguna en el receptor que se encarga de medir la distancia con el caballo precedente.

Cuando se trabaja solo con uno de los sensores no es necesario economizar el uso de los timers pero al unir los proyectos sí, ya que solo se dispone de tres timers. De esta manera, se utiliza un timer para el emisor delantero, otro para el emisor trasero y otro se separa en un archivo independiente (`tmr.c`) para darle varios usos. Este archivo es una variación del archivo con el mismo nombre utilizado en el proyecto de Miguel Ángel Sánchez.

Así, los timers quedan distribuidos de la siguiente forma:

- **Timer 1:** Archivo `tmr.c`
- **Timer 2:** Archivo `emisor_delante.c`
- **Timer 3:** Archivo `emisor_trasero.c`

Además, en el programa principal del proyecto (`main.c`) se han fusionado las distintas partes, cada una referente a una zona del caballo. Estas partes se han separado con comentarios para mejorar el entendimiento de código, como puede observarse en el Extracto de código 5. De esta forma con un solo microprocesador se puede interaccionar con todos los sensores desarrollados a lo largo de este proyecto y sus predecesores.

```
while(1) {  
  
    //Sensor delantero  
    set_dc_vibrador_trasero(0);  
    if(get_ad_delante(2)>200){  
        distancia = calcular_distancia(tiempo);  
        if(distancia>17){  
            set_dc_pwm(5);  
            wait(2000);  
        }  
        if(distancia<13){  
            set_dc_pwm(25);  
        }  
    }  
}
```

```
        wait(2000);
    }
    tiempo = 0;
    tiempo_seguro = 0;
    enviar_senal_delante();
}

if(tiempo_seguro>10000){
    enviar_senal_delante();
    tiempo = 0;
    tiempo_seguro = 0;
}

//Sensor trasero
if(get_ad_trasero(3) >= 200){
    init_emisor_trasero();
}

//Sensores laterales
potenciometro = get_ad();
set_dc_pwm(1,0);//parar conmutaciones del pwm para los sensores
set_dc_pwm(2,0);

for(i = 0; i < 2; i++){
    dist[i] = rollingAverage(get_sensor(i+1),i+1);
    pwm_dc[i] = map(dist[i]);
    if(pwm_dc[i] < 0)
        pwm_dc[i] = 0;
    wait(2);
}
set_dc_pwm(1, (int)pwm_dc[0]*potenciometro/0x3FF);
set_dc_pwm(2, (int)pwm_dc[1]*potenciometro/0x3FF);
for(i = 0; i<1000;i++){
    wait(1000);}}
```

Extracto de código 5: Extracto del programa main.c final

En conclusión, se crea un conjunto de funciones coordinadas entre sí que se encargan de medir todas las distancias que influyen en la posición del jinete dentro del recinto y respecto a los otros jinetes que haya en la pista con él en ese momento para asegurar que se encuentra en el lugar adecuado siempre.

Así, puede realizar este deporte de manera más segura y también más libre que si dependiera de un guía que le acompañase a pie todo el rato. También proporciona más seguridad al monitor, que ya no tiene que estar todo el rato pendiente de la posición de este alumno en concreto.

En resumen, los puertos y timers utilizados en un microprocesador para poder usar todos los sensores a la vez son:

- RA0 (AN0)
- RA1 (AN1)
- RB0 (AN2)
- RB1 (AN3)
- RB2
- RB3
- RB4
- RB5
- RB9 (PWM2L1)
- RB11
- RB12
- RB13 (PWM1L2)
- RB14
- RB15 (PWM1L1)
- Timer 1
- Timer 2
- Timer 3

Capítulo 8. ANÁLISIS DE RESULTADOS

En este capítulo se analizan los resultados obtenidos del proyecto. Primero, se resume cuál es el funcionamiento final del sistema. Después, se mencionan las pruebas realizadas con el sistema para comprobar el funcionamiento de este. Por último, se resumen algunos aspectos que podrían mejorarse en proyectos futuros.

8.1 FUNCIONAMIENTO DEL SISTEMA

En el sistema final, el caballo cuenta con cuatro sensores y tres actuadores para medir todas las distancias necesarias para asegurar la correcta posición del jinete en el camino. Los tres actuadores tienen funciones distintas, sirviendo dos para indicar la distancia a las paredes laterales y uno para indicar la posición con respecto al caballo precedente.

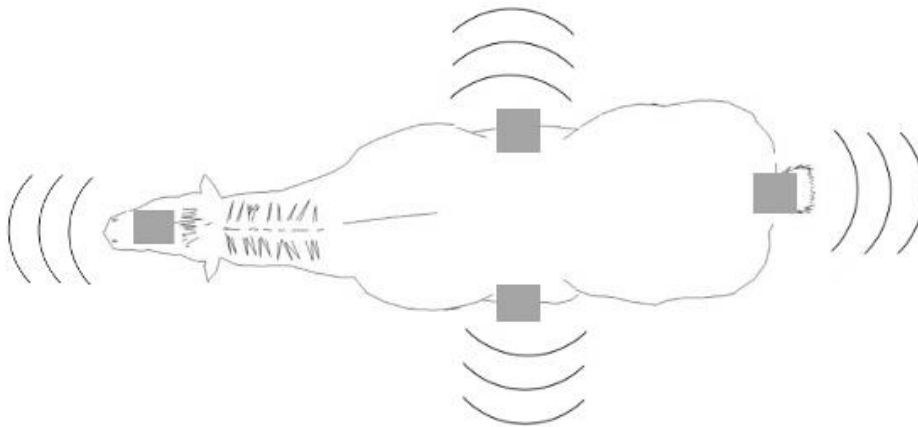


Figura 32: Posiciones orientativas de los cuatro sensores

En la Figura 32 se muestra un esquema orientativo de cómo quedarían situados los cuatro sensores, todos ellos siendo sensores de ultrasonidos. De esta manera se controlan los cuatro flancos del caballo y se asegura que la posición del jinete es la correcta durante el desarrollo de la actividad.

Los tres sensores funcionan de manera similar, vibrando en función de la corrección de la posición que haya que hacer. Por lo tanto, los actuadores vibrarán más si el animal se encuentra demasiado cerca del elemento de guía y vibrarán menos si se encuentra demasiado alejado del mismo.

Estos vibradores se sitúan en distintas partes del cuerpo del jinete para que pueda relacionar cada vibración con la medida adecuada. Los sensores que miden la distancia con las paredes laterales van situados cada uno en una pierna del jinete. El vibrador derivado de la medida de la distancia con el caballo precedente se sitúa cerca de la mano del jinete. Así, vibrará una parte u otra del cuerpo del jinete en función de en qué dirección debe moverse el jinete.

Todos los componentes del sistema están coordinados por el programa principal del microprocesador situado en la PCB, que cuenta con una función específica dedicada a cada componente o parte del sistema.

8.2 PRUEBAS REALIZADAS

Las pruebas se han ido realizando a medida que se ha desarrollado el proyecto, de manera que se ha comprobado que cada etapa funciona antes de pasar a montar la siguiente.

En primer lugar, se comprobó el funcionamiento de una pareja emisor/receptor en solitario mediante un osciloscopio unido al circuito con diversas sondas. Con estas sondas se controló la señal de salida y la señal de entrada y se midió el retraso en tiempo desde que se emite la primera hasta que se recibe la segunda. También se estudió como variaba la amplitud de la señal recibida respecto a la emitida para poder ajustar la amplificación necesaria en el circuito y poder detectar la señal sin confusiones.

A continuación, se comprobó la comunicación entre dos parejas de emisores/receptores, de manera que la señal viajaba de un lado al contrario y después regresaba. Este proceso también se realizó con osciloscopio y sondas, situando los circuitos a diferentes distancias entre sí. En la Figura 33 se ilustra el esquema montado para las pruebas y cuál era la distancia medida para tomarla de referencia. De esta manera se comprobó que los circuitos

funcionaban en el rango de distancias correcto y no solo en una distancia concreta. Para estas pruebas se emitieron señales desde ambos lados alternadamente para poder detectarlas al otro lado sin que se mezclaran entre sí.

Finalmente se comprobó que las distancias calculadas eran correctas, conectando el microprocesador al ordenador y sacando por pantalla la distancia calculada. Esta distancia se comparaba con la distancia de separación que había entre los sensores, que era medida de manera manual con una regla. El rango de desviación entre la medida calculada y la medida real era del orden de medio centímetro, lo cual para la distancia que quiere medirse (1.5 metros) no resulta relevante.

Esta parte final de sacar el resultado por pantalla no se incluye en los anexos ya que no es relevante ni necesario para el montaje final del sistema completo. Sólo se utilizó como comprobación adicional de que todo funcionaba correctamente.

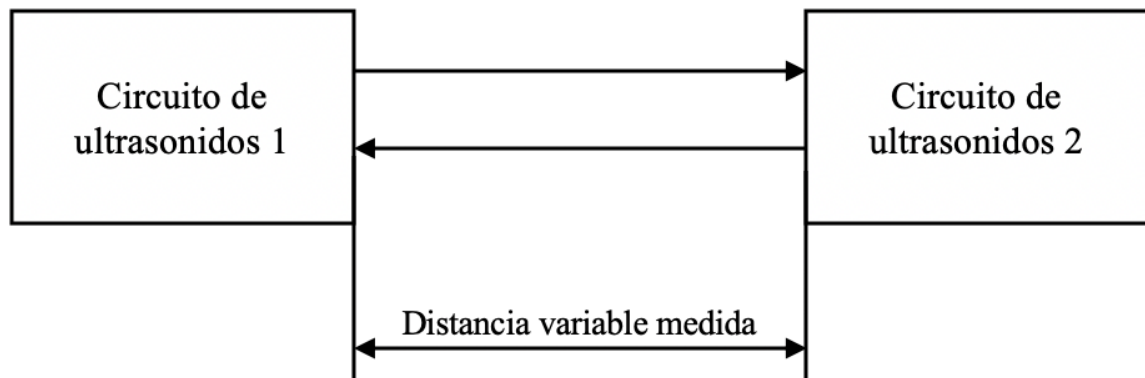


Figura 33: Esquema de las pruebas entre dos circuitos

No se han realizado pruebas con el diseño final completo por complicaciones externas al proyecto, derivadas del cierre de las instalaciones de la universidad durante un cierto periodo de tiempo.

En resumen, se espera que el sistema funcione de la manera esperada y pueda coordinar las funciones de todos los componentes en un solo microprocesador y todos los circuitos en una

sola PCB resultando un diseño bastante compacto. Esto se comprobará haciendo las pruebas pertinentes al sistema, cuando sea posible hacerlas.

8.3 ASPECTOS A MEJORAR

Aunque se van a mencionar varios aspectos a mejorar en proyectos posteriores, son aspectos que no afectan a la funcionalidad del sistema completo desarrollado hasta el momento, sino que suponen una mejora en la calidad de este. De esta forma, el sistema puede funcionar sin estos avances, pero desarrollarlos mejoraría el acabado final y favorecería una experiencia más satisfactoria para el jinete a la hora de practicar este deporte.

En primer lugar, este sistema, al igual que el del proyecto anterior, se alimenta mediante ocho pilas alcalinas de un voltio y medio, que proporcionan una tensión total de doce voltios. A la hora de montar la PCB los circuitos originales sufren ciertas adaptaciones ya que estaban diseñados para alimentarse con otros voltajes distintos.

Por ejemplo, el amplificador operacional iba a alimentarse a ± 5 V de manera simétrica, por lo que la señal recibida saturaría en 5 V y -5 V al amplificarse. Al modificar el diseño anterior, se acabó alimentando este operacional a 5 V y 0 V y por ende la señal satura a 5 V o 0 V, dependiendo de la señal de entrada.

No es un cambio significativo en el diseño del circuito ni afecta a todo el desarrollo del circuito, pero hace que el rango en el que se amplifica esta señal sea menor del que se pensaba utilizar.

En proyectos futuros podría trabajarse para sustituir este tipo de alimentación por otra más eficiente y de menor tamaño, como por ejemplo baterías LiPo recargables o pilas AAA, que facilitase el disminuir el tamaño total del sistema desarrollado.

Existen diversos modelos de pilas de botón que proporcionan hasta 3 V de tensión y distintas corrientes, por lo que no sería complicado adaptar el sistema a este tipo de alimentación.

Podría mejorarse el sistema de aviso al usuario de manera que no se utilicen siempre vibraciones y permitiendo que las distintas señales que recibe el jinete puedan diferenciarse mejor entre sí. Si se siguieran usando las vibraciones, podría sustituirse la vibración continuada por patrones de vibración que simplificasen el entendimiento del mensaje.

Por ejemplo, en cuanto a otro tipo de avisos para el jinete, podría conectarse un auricular al sistema de manera que al jinete se le indique con frases sencillas y de manera sonora qué distancia debe variar y cómo.

También, podría diseñarse otra PCB optimizada en la que se disminuyera el tamaño de la placa, haciendo así que el sistema fuera más compacto. Aunque se disminuye el tamaño del diseño al montar los circuitos en una PCB, el sistema final sigue teniendo un tamaño considerablemente grande.

El tamaño de la PCB unido a que esta se encuentra situada dentro de una caja metálica para proteger los circuitos hace que el sistema tenga un mayor tamaño del que debería. Como esta placa se va a situar en la montura, podría ser incómodo para el usuario.

Por último, si se cambiase la alimentación del sistema, pero no quisiera usarse pilas de botón ni otras pilas similares, podría sustituirse por una batería que permita la carga de esta. Podría ser similar a la de un teléfono móvil, y se recargaría conectándolo a la red eléctrica usando un cargador adaptado a su voltaje. Suelen tener un tamaño pequeño ya que están optimizadas para que quepan dentro de dispositivos móviles y, además, como solo se les daría el uso de alimentar este sistema, durarían bastante tiempo sin necesitar recargarlas. Es un sistema de alimentación más efectivo que las pilas que necesitan cambiarse cuando se agotan y si no se cuidan bien durante algún tiempo pueden segregar sustancias contaminantes.

Capítulo 9. CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se analizan las conclusiones principales sacadas de este Trabajo de Fin de Grado, además de distintos avances que podrían realizarse en proyectos futuros para mejorar el sistema ya existente.

9.1 CONCLUSIONES PRINCIPALES

En este proyecto se ha conseguido desarrollar un sistema que mide la distancia entre el caballo del invidente y el caballo delantero y fusionarlo con un proyecto anterior que medía la distancia con las paredes laterales del recinto. Estos dos proyectos se han fusionado en una sola PCB que hace que funcionen de manera conjunta. Si la distancia con los objetos de alrededor es excesiva o escasa se avisa al jinete con una señal en forma de vibración para que corrija el error.

Al mantener una distancia adecuada con los otros caballos de la pista se aumenta la seguridad del jinete. Si estuviese demasiado cerca podría ocasionar una pelea entre los dos caballos, donde el delantero le daría una coza al trasero (el del usuario), y si estuviese demasiado lejos, el animal perdería la referencia delantera y podría salirse del camino estipulado y acabar en cualquier lugar aleatorio que el caballo considerase apropiado.

La distancia comúnmente aceptada en este deporte es aproximadamente un metro y medio ya que es lo suficientemente amplia como para que la mayoría de los caballos no alcancen a los que vayan detrás de ellos de una coza y lo suficientemente cercana como para que el caballo de detrás quiera seguir al precedente.

Los objetivos principales de este Trabajo de Fin de Grado eran desarrollar el sistema de medición de distancia con el caballo precedente y avisar al jinete en el caso de que no guardase una distancia adecuada, además de fusionar los códigos de todos los sensores en un microprocesador y sus circuitos en una PCB, por lo que se han conseguido con éxito.

9.2 TRABAJOS FUTUROS

En función de las necesidades y motivaciones del usuario y del profesor podrían realizarse diversos avances, que favorecerían la obtención de información del esfuerzo del caballo y del jinete para su posterior estudio y la organización de las clases para favorecer su gestión por parte del profesorado. En este apartado se proponen tres posibles avances que le darían valor añadido al diseño final desarrollado durante el proyecto.

9.2.1 MONITORIZACIÓN DEL TRABAJO DEL ANIMAL

En trabajos futuros relacionados con este proyecto de fin de grado podría desarrollarse un sistema de sensores que obtenga datos sobre el estado del caballo. Se colocarían en el cuerpo del animal y recogerían los datos sobre el estado biológico del mismo a lo largo del desarrollo de la clase.

Sería interesante saber sus pulsaciones y ritmo respiratorio para saber si el animal está tranquilo, asustado o cansado. Con estos datos podría incluso anticiparse un movimiento brusco del animal si se asustase y el incidente estaría prevenido ante estas posibles situaciones.

También serviría para llevar un registro de las sesiones de trabajo del caballo y de la intensidad de estas y así evitar que trabaje de más y se agote. Podría planificarse de manera más sencilla los días que le toca ejercitarse y cuántas horas en función de su morfología y metabolismo, en adición de su forma muscular. Además, se podría saber durante cuánto tiempo puede trabajar el caballo a máximo rendimiento para sacarle el máximo partido a la clase.

Todos estos datos podrían almacenarse en una agenda virtual para el profesor como la de la Figura 34, que podría llevarla descargada como una aplicación en el móvil. En esta agenda tendría distintas pantallas en función del caballo que fuera.

En cada pantalla se anotarían los datos pertinentes según el animal, como notas importantes sobre su estado, clases que va a tener ese día, objetivos marcados para la doma del animal durante ese mes y las clases que ya ha tenido ese día.

Tendría en el mismo lugar la ficha de todos los caballos con los que cuenta y los datos de estos, lo cual facilitaría su eficiencia en el trabajo. De esta manera, podría pasar fácilmente de los datos de un caballo a los datos de otro y todo estaría al alcance de su mano.

Organización del día (Nombre Caballo) - 30/03/2020

NOTAS IMPORTANTES

- Revisar cojera pierna izquierda
- Herrar patas delanteras

CLASES DEL DÍA

10:00 Clase de iniciación (2 personas)

11:30 Clase de salto (4 personas)

13:00 Clase de equitación adaptada (1 persona + guía)

18:00 Ruta a la playa (6 personas)

CALENDARIO MENSUAL

APRIL 2014

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

OBJETIVOS MENSUALES

- Mejorar incurvación en las esquinas
- Aumentar respuesta a las ayudas

CLASES DADAS HOY

- 9:00 – 1 hora
- 10:00 – 30 minutos
- 11:30 – 30 minutos

Figura 34: Ejemplo de pantalla de agenda virtual

9.2.2 APLICACIÓN PARA TELÉFONO MÓVIL

Por otro lado, podría desarrollarse una aplicación para el móvil que permitiera al profesor saber la distancia media que lleva el jinete respecto al caballo delantero y las paredes. Estos

datos los obtendría la aplicación de los sensores puestos sobre los caballos, de manera que solo habría que desarrollar la comunicación entre la aplicación y los sensores y la aplicación en sí, reutilizando el sistema de sensores desarrollado en este proyecto. También podría informarle de la velocidad media que mantiene y de los cambios bruscos de la misma, debidos a frenazos o acelerones en gráficas como la mostrada en la Figura 35. Para que la app pudiera ser usada por los jinetes, tendría que adaptarse para que fuera accesible para personas invidentes. Esto podría hacerse añadiendo diversos comandos que permitiesen que el móvil leyera en voz alta al usuario los datos mostrados por pantalla.



Figura 35: Ejemplo pantalla de una alumna

Esto serviría para el aprendizaje del jinete, ya que podría monitorizar sus fallos de ritmo y trabajarlos. Corregir estos fallos le permitiría desarrollarse como deportista y aumentar su nivel en esta disciplina deportiva.

También sería útil para que el monitor pudiera saber cómo está trabajando el jinete actualmente, para así planificar clases posteriores y ayudarle a su desarrollo deportivo, además de ver sus avances a lo largo de los entrenamientos.

Sería algo parecido a un diario virtual sobre las mejoras del jinete que a su vez serviría como agenda o diario personal de sus logros en este deporte. En definitiva, una manera sencilla de llevar el seguimiento de los avances en el deporte de la doma de todos los alumnos de un mismo centro hípico.

En esta aplicación cada alumno tendría un perfil con sus datos y el profesor a su vez tendría acceso a todos los perfiles de sus alumnos, de manera que podría acceder fácilmente a toda la información disponible en la plataforma.

9.2.3 CONEXIÓN ENTRE DISTINTOS JINETES INIDENTES

Por último, podría adaptarse el sistema para que varios caballos contaran con los sensores y se comunicaran entre sí, de manera que hubiera en la misma clase varios jinetes inidentes que hicieran doma a la vez.

En la Figura 36 se muestra un esquema de cómo se comunicarían los sensores que miden la distancia entre los caballos.



Figura 36: Varios caballos conectados entre sí

Estos sensores comunicados entre sí se asegurarían de que cada jinete guardase la distancia adecuada con los siguientes, por lo que en un mismo caballo habría sensores a los lados del animal, en la cabeza y en la cola de este y se tendrían controladas todas las distancias.

En este proyecto se ha diseñado la PCB para que puedan colocarse todos los sensores en la misma, además del código que permite que funcionen a la vez todos los sensores, el delantero, el trasero y los laterales.

En el futuro podría optimizarse este sistema para que fuera más sencillo y preciso, reduciendo su tamaño y mejorando su eficiencia. Cuanto más pequeño sea el sistema final, más sencillo será para el usuario llevarlo y más cómodo será su uso.

Si este proceso se optimiza lo suficiente, podrían tenerse coordinados en una sola clase a un caballo guía y múltiples caballos con jinetes invidentes, cada uno con sus sensores, trabajando de forma conjunta.

Capítulo 10. OBJETIVOS DEL DESARROLLO SOSTENIBLE

10.1 INTRODUCCIÓN

En este capítulo se va a estudiar la relación que tiene el proyecto realizado durante este Trabajo de Fin de Grado con los Objetivos de Desarrollo Sostenible de Naciones Unidas fijados para el año 2030.



Figura 37: Objetivos del desarrollo sostenible [25]

Como se observa en la Figura 37 los Objetivos de Desarrollo Sostenible son 17. Estos objetivos pretenden conseguirse con ayuda de todos los países del mundo para mejorar la vida de todas las personas. Algunos de los objetivos son para ayudar al medioambiente, otros para lograr la paz entre los distintos países y otros centrados en la vida de las personas, intentando solucionar aspectos como el hambre en el mundo, la educación de calidad y la disponibilidad de trabajos dignos y crecimiento económico para todos.

Fueron establecidos por los Estados Miembros de la ONU en 2015 para erradicar la pobreza, proteger el planeta y asegurar la paz. La fecha límite para alcanzarlos es 2030. Son objetivos relacionados entre sí, ya que avanzar hacia el alcance del uno hace que se avance en el de otros a la par, buscando así el equilibrio entre la sostenibilidad medio ambiental, económica y social.

Cada objetivo cuenta con varias metas que son aspiraciones a nivel mundial, por lo que pueden variar en cada país para respetar las políticas y prioridades de este. El propósito de las metas es servir de guía y que se fijen en cada país metas particulares nacionales que favorezcan la consecución de los Objetivos del Desarrollo Sostenible, dentro de las capacidades de cada territorio. En definitiva, estas metas son pequeños objetivos dentro del objetivo en sí, de manera que a medida que se van cumpliendo se está más cerca de alcanzar el objetivo completo.

Toda la información sobre los ODS y sus metas se encuentra disponible en la página web de la ONU.

10.2 RELACIÓN CON EL PROYECTO REALIZADO

El Trabajo de Fin de Grado realizado durante este curso guarda una estrecha relación con estos objetivos, en concreto con el Objetivo de Desarrollo Sostenible número 10: reducción de las desigualdades en y entre países. Entre las múltiples metas que quiere conseguir este objetivo se encuentra la de “potenciar y promover la inclusión social, económica y política de todas las personas, independientemente de su edad, sexo, discapacidad, raza, etnia, origen, religión, situación económica o cualquier otra condición”, como indica la ONU en su página web [26].

Por ello, al favorecer la inclusión social de personas con discapacidad en actividades cotidianas para el resto de las personas se persigue alcanzar esta meta.

La ceguera es una discapacidad que afecta a una pequeña parte de la sociedad, pero en gran medida. Dicho trastorno hace que no sea posible realizar actividades comunes entre el resto

de las personas, ya que dependen en su mayor parte del sentido de la vista, como conducir o pintar.

Este colectivo ya cuenta con soluciones para algunos de los problemas, como por ejemplo la lectura en braille o aplicaciones que les ayudan a usar el teléfono móvil leyéndoles los mensajes en pantalla y describiendo las imágenes o dispositivos que responden a comandos realizados por voz, pero en el ámbito deportivo se encuentran muy limitados.

Las soluciones que se ofrecen actualmente suelen ser restrictivas ya que requieren la presencia de un guía en la mayoría de los casos.

10.2.1 DEPORTE Y DISCAPACIDAD

Un gran número de disciplinas deportivas cuentan con facilidades para las personas con discapacidades, dependiendo de cuál sea esta y en qué grado se tenga, para ayudarles a disfrutar de la práctica deportiva adaptada a sus necesidades. En concreto, cuando la discapacidad es visual, los deportistas cuentan con personas guías que los acompañan durante el desarrollo de la actividad. Por ejemplo, como se puede ver en la Figura 38, en las pruebas de ciclismo cuentan con bicicletas de dos plazas y el guía se sitúa en la delantera para dirigir la marcha.



Figura 38: Ciclista ciega y su guía [27]

Aunque la equitación puede parecer un deporte poco adaptable a las diferentes discapacidades que puede tener una persona, cuenta con disciplinas de equitación adaptada para personas con necesidades especiales. Estas ayudas permiten realizar este deporte de forma segura y controlada. Con esta práctica, personas con parálisis musculares o problemas de desarrollo pueden aprender esta afición y, por tanto, también pueden hacerlo personas con ceguera si se les proporcionan los medios adecuados para ello.

En el caso de discapacidades de tipo muscular, como por ejemplo parálisis o falta de desarrollo de alguna zona, más que equitación adaptada se realiza equinoterapia. Esta práctica consiste en sesiones de fisioterapia a lomos de un caballo, de manera que el movimiento de este favorece el movimiento del alumno y desarrolla su musculatura sin suponerle un gran esfuerzo. Se han conseguido grandes avances con niños y adultos que contaban con grandes pérdidas de musculatura y han podido recuperarla lentamente con la práctica de este deporte.

El problema de este tipo de terapias es que no suele ser económica y, en familias en las que estas personas con necesidades especiales ya acuden a multitud de terapias de alto precio, no suele plantearse la opción de probar esta nueva forma de terapia. Aún así, existen muchas fundaciones que recaudan dinero con otras actividades hípicas para disminuir el precio de las sesiones de equinoterapia y hacerlas más asequibles para las personas que realmente las necesitan.

Para las personas que sufren algún tipo de discapacidad motora, la equitación cuenta ya con las adaptaciones pertinentes para que puedan practicarla, pero en el caso de la invidencia, todavía no se han incorporado los medios suficientes como para que sea una práctica extendida. Por eso se ha desarrollado este sistema de ayuda al usuario, que interactúa de manera simple con el mismo para que pueda disfrutar de este deporte sin problemas.

En definitiva, la práctica de la equitación es beneficiosa para las personas, tengan o no cualquier tipo de discapacidad. El contacto con el animal ayuda al usuario a favorecer aspectos psicológicos mejorando la autoestima, la concentración, aumentando la confianza,

disminuyendo la ansiedad y creando una sensación de bienestar general. Se establece una conexión emocional entre el animal y el jinete que aporta múltiples beneficios al usuario.

A nivel físico, se mejora el tono muscular, la motricidad, se aumenta el equilibrio y se ayuda a movilizar la columna y las caderas, tonificando la musculatura de la zona de una forma más específica que con otros deportes [28].

Por ello, esta disciplina deportiva mejora el nivel de vida de la persona que la práctica y mejora su forma física en general, lo cual es beneficiosa para la salud de esta.

La postura que hay que adoptar a la hora de montar a caballo fortalece también la espalda y ayuda a prevenir dolores en la misma. Esta postura se muestra en la Figura 39, donde puede verse a una amazona montando a caballo.



Figura 39: Postura correcta a la hora de realizar doma [29]

Como puede observarse, la espalda debe mantenerse erguida y recta y los codos tienen que estar flexionados de tal manera que, entre la mano, la rienda y la boca del caballo se produzca una línea recta. Además, las piernas deben caer relajadas a los lados de la barriga del animal, haciendo que los talones estén a un nivel un poco más bajo que el resto del pie.

Con el dispositivo con sensores realizado en este proyecto se pretende acercar la disciplina deportiva de la hípica a los invidentes para que puedan disfrutar de este deporte como el resto de los usuarios. Para que puedan montar a caballo con libertad, sin depender de un guía, el sistema que se ha desarrollado cuenta con sensores colocados en el animal que le comunican al usuario cómo y dónde situarse para seguir el camino.

Ayudando a estas personas a practicar el deporte de la equitación se potencia la inclusión social de los mismos, y con ello se favorece el cumplimiento del objetivo diez de los Objetivos de Desarrollo Sostenible de la ONU.

BIBLIOGRAFÍA Y REFERENCIAS

- [1] Proyecto Fin de Grado: Sistema de asistencia a invidentes para práctica de la equitación en pista. Miguel Ángel Sánchez Sánchez. (2019). Última consulta: 4 de abril de 2020, desde: <https://repositorio.comillas.edu/xmlui/handle/11531/31913>
- [2] dsPIC33FJ32MC202. Último acceso: 4 de abril de 2020, desde <https://www.microchip.com/wwwproducts/en/dsPIC33FJ32MC202>
- [3] Cebek C-7210 pareja emisor-receptor ultrasonidos. Último acceso: 4 de abril de 2020, desde <https://fadisel.com/es/sensor/160-pareja-de-emisor-yreceptor-de-ultrasonidos-cebek.html>
- [4] Z7AL2B1692082. Último acceso: 4 de abril de 2020, desde <https://www.vibration-motor.com/products/cylindrical-vibratormotors/encapsulated-vibration-motors/z7al2b1692082>
- [5] Printed PCB Circuit Board Customed Printed Circuit Boards. Último acceso: 4 de abril de 2020, desde: <http://www.ourpcbte.com/products-detail/printed-pcb-circuit-board/>
- [6] ¿Qué es una PCB o tarjeta de circuito impreso? Último acceso: 4 de abril de 2020, desde: <https://www.altium.com/es/solution/what-is-a-pcb/>
- [7] C (lenguaje de programación). Último acceso: 4 de abril de 2020, desde: [https://www.ecured.cu/C_\(lenguaje_de_programaci%C3%B3n\)](https://www.ecured.cu/C_(lenguaje_de_programaci%C3%B3n))
- [8] MPLAB X IDE. Último acceso: 14 de abril de 2020, desde: <https://www.microchip.com/mplab/mplab-x-ide>
- [9] KiCad. Último acceso: 4 de abril de 2020, desde: <https://www.ecured.cu/Kicad>
- [10] KiCad: una herramienta open source para el diseño de circuitos. Publicado el 28 de julio de 2014. Último acceso 4 de abril de 2020, desde: <https://oshl.edu.umh.es/2014/07/28/kicad-una-herramienta-open-source-para-el-diseno-de-circuitos/>
- [11] El primer «smartphone» para ciegos ya está aquí. ABC, Septiembre 2017. Último acceso: 4 de abril de 2020, desde <http://blogthinkbig.com/big-data-emprendedor/>

- [12] 5 tecnologías que pueden mejorar la vida de las personas con discapacidad visual. Julieta Schulkin, 30 de junio de 2018. Último acceso: 4 de abril de 2020, desde <https://www.infobae.com/tecno/2018/06/30/5-tecnologias-que-pueden-mejorar-la-vida-de-las-personas-con-discapacidad-visual/>
- [13] Sensores estacionamiento, dispositivos seguridad y precaución. Último acceso: 4 de abril de 2020, desde: <https://www.servitronic.eu/instalar-sensores-estacionamiento/>
- [14] Circuito de ultrasonidos. Último acceso: 4 de abril de 2020, desde: <https://www.forosdeelectronica.com/threads/circuito-de-ultrasonidos.7499/>
- [15] Un buscador de rango ultrasónico barato. Último acceso: 4 de abril de 2020, desde: http://translate.google.com/translate?hl=en&ie=UTF8&prev=_t&rurl=translate.google.com&sl=en&tl=es&u=http://www.micro-examples.com/public/microex-navig/doc/090-ultrasonic-ranger.html
- [16] Sensores de distancia infrarrojos para Arduino. Último acceso: 4 de abril de 2020, desde: <http://proyectosconarduino.com/sensores/sensores-de-distancia-infrarojos-para-arduino/>
- [17] Fotocélula Haz pasante, Infrarrojo. Último acceso: 1 de junio de 2020, desde: [Enlace a RS](#)
- [18] Bosch Glm 120 Medidor Láser. Último acceso: 4 de abril de 2020, desde: https://articulo.mercadolibre.com.mx/MLM-666135902-bosch-glm-120-medidor-laser-glm120-telemetro-distanciometro-_JM?quantity=1
- [19] Hípica Las Cadenas. Último acceso: 1 de junio de 2020, desde: <http://www.hipicalascadenas.com/home/instalaciones>
- [20] ¿Dónde tirar pilas y baterías? Último acceso: 15 de abril de 2020, desde: <https://ecoembesdudasreciclaje.es/donde-tirar-pilas-y-baterias/>
- [21] Emisor y receptor de ultrasonidos. Último acceso: 4 de abril de 2020, desde: <https://www.electan.com/datasheets/cebek/CE-C7210.pdf>
- [22] Portapilas 4 pilas AAA LR03 con interruptor y tapa. Último acceso: 5 de abril de 2020, desde : <https://www.tiendatec.es/electronica/baterias-pilas-y-accesorios/880-portapilas-4-pilas-aaa-lr03-con-interruptor-y-tapa-8472496014397.html>

- [23] An Intro to KiCad – Part 1: How PCBs are made | DigiKey. Último acceso: 25 de abril de 2020, desde: <https://www.youtube.com/watch?v=vaCVh2SAZY4>
- [24] Pilas de botón de litio CR2354. Último acceso: 8 de abril de 2020, desde: <https://pilanet.es/pilas/pilas-boton/pilas-de-boton-de-litio/pila-de-boton-de-litio-cr2354.html>
- [25] Objetivos del Desarrollo Sostenible. Naciones Unidad. Último acceso: 4 de abril de 2020, desde: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [26] Objetivo 10: Reducir la desigualdad en y entre los países. Último acceso: 4 de abril de 2020, desde: <https://www.un.org/sustainabledevelopment/es/inequality/>
- [27] ‘Kuki’ López y Mayalen Noriega: un tándem único que sueña con Tokio 2020. Marca, marzo, 2019. Último acceso: 4 de abril 2020, desde: <https://www.marca.com/paralimpicos/2019/03/14/5c896625e5fdea4d368b4571.html>
- [28] Beneficios de montar a caballo para las personas con discapacidad. Actualizado el 29 de junio de 2018. Último acceso : 6 de abril de 2020, desde: <https://www.infosalus.com/asistencia/noticia-beneficios-montar-caballo-personas-discapacidad-20180629112102.html>
- [29] Mejorarás el contacto de tus manos, liberando las escapulas. Marc Elmeua. Publicado el 31 de enero de 2019. Último acceso: 6 de abril de 2020, desde: <https://www.equisens.es/fisio-del-jinete/mejoraras-el-contacto-de-tus-manos-liberando-las-escapulas/>
- [30] Componentes SMD (Montaje Superficial). Último acceso: 4 de junio de 2020, desde: <https://www.orgontec.com/smd-montaje-superficial>
- [31] Conector Circular Estándar Amphenol. Último acceso: 17 de junio de 2020, desde: <https://www.mouser.es/ProductDetail/Amphenol-SINE-Systems/RTS712ND4S03?qs=rI7uf1IzohQMneGoQELyaA%3D%3D>

ANEXO I: ESTUDIO ECONÓMICO

COSTE DE LOS COMPONENTES

A la hora de estimar el coste del sistema desarrollado hay que tener en cuenta varios aspectos, que tendrán distintos costes y distinta importancia en el presupuesto final del sistema. En primer lugar, los costes se pueden dividir en tres partes: coste de componentes y de fabricación de la placa, coste de los softwares usados para el desarrollo de la placa y de la programación y coste de la persona que se dedica a diseñar el sistema, programarlo, crearlo y montarlo.

Tabla 2: Estimación del coste de los componentes

Componentes			
Tipo	Número	Precio unitario	Precio total
Emisor ultrasonidos	2	6,72 €	13,44 €
Receptor ultrasonidos	2		
Microprocesador	2	3,28 €	6,56 €
Resistencia	14	0,30 €	4,20 €
Transistor	2	0,02 €	0,04 €
Diodos	4	0,04 €	0,16 €
Condensador	2	0,16 €	0,32 €
Amplificador operacional	2	0,78 €	1,56 €
Vibrador	1	3,59 €	3,59 €
Pila Alcalina	8	0,38 €	3,04 €
Portapilas	1	1,16 €	1,16 €
Fabricación PCB	1	23 €	23 €
Conectores DB9	2	1,69 €	3,38 €
Caja para el sistema	1	7,14 €	7,14 €
			67,59 €

En la Tabla 2 se puede observar la estimación de costes de los componentes y de la fabricación de la placa.

También habría que tener en cuenta que se necesita un programador para descargar los programas en el microprocesador y ejecutarlos, pero, al ser un útil que no va a quedarse permanentemente en el proyecto y que luego va a reutilizarse para seguir programando otros microprocesadores, el coste de ese programador en este proyecto sería mínimo comparado con el resto y no se ha tenido en cuenta.

Otro dato para tener en cuenta de estos componentes es que el emisor y el receptor de ultrasonidos solo se venden conjuntamente, al ser una pareja que trabajan de manera complementaria.

En cuanto al software utilizado, KiCad es un paquete de software libre por lo que no hay que pagar para utilizarlo. Para programar en C se pueden usar múltiples programas gratuitos, como por ejemplo MPLAB-IDE, que es el usado en este caso.

Además, a la hora de investigar sobre asuntos relacionados con el proyecto se ha utilizado el navegador Google Chrome, el cual se puede descargar de manera gratuita. Por lo tanto, la parte de software no añade ningún coste al proyecto.

COSTE DEL PROYECTISTA

Por último, estaría el coste de la persona que se encarga en el proyecto de idearlo, programarlo, fabricarlo y montarlo. Para este coste, se ha fijado un sueldo orientativo general por hora de trabajo y se han estimado las horas dedicadas a cada parte del proyecto, siendo las partes a las que se les ha dedicado más tiempo programar los microcontroladores y diseñar la PCB.

El importe total es orientativo y dependería del sueldo real del trabajador, que variará en función de los conocimientos de este, su experiencia, disponibilidad y otros factores ajenos a él como por ejemplo la urgencia de la realización del proyecto.

Este análisis se observa en la Tabla 3.

Tabla 3: Estimación del coste del proyectista

Mano de obra		
Descripción	Horas	Precio
Estudio TFGs anteriores	50	30 €/hora
Diseño electrónico	70	
Diseño PCB	100	
Programación	100	
Pruebas y depurado	30	
Documentación	70	
	420 horas	12.600 €

COSTE DE LOS EQUIPOS DEL LABORATORIO

También podría tenerse en cuenta el precio de los aparatos del laboratorio usados a la hora de desarrollar el proyecto como coste adicional a la hora de desarrollar el proyecto. Se han utilizado tanto osciloscopios como polímetros, fuentes de alimentación o generadores, pero, al ser equipamiento que se encuentra en la mayoría de los laboratorios electrónicos y usarse en la mayoría de los proyectos que se realizan en estos, no se consideran en esta estimación.

El precio de estos aparatos no es demasiado elevado y, al estar en continuo funcionamiento para los diferentes proyectos electrónicos que se realizan en los talleres, son muy amortizados a lo largo de su vida útil por lo que considerarlos no aumentaría en gran medida el presupuesto total del proyecto.

Aún así, se puede tener en cuenta la amortización de estos a lo largo de los años y cómo afecta esta durante el desarrollo del proyecto. Los equipos de los que se va a detallar la amortización son los utilizados durante el estudio del proyecto anterior y el desarrollo del actual. Para calcular el coste de estos equipos en el proyecto, se supone una amortización del 20% en los que se encuentran al laboratorio, ya que suele ser lo habitual en este tipo de instrumentación. En el ordenador, al ser personal y estar en continuo uso para otras labores, se supone una amortización mayor.

Tabla 4: Amortización de los equipos a lo largo del proyecto

Equipo	Cantidad	Uso [h]	Anual [h]	Amortización [%]	Precio [€]	Subtotal [€]
Osciloscopio	1	200	1200	25%	900	30,82
PicKit3	1	75	450	25%	50	0,64
Fuente de alimentación	1	250	1500	25%	200	8,56
Equipo de soldadura	1	10	60	25%	50	0,09
MacBook Pro (13", 2018)	1	252	756	33%	1600	45,57
						85,68

El coste total del uso de los equipos puede observarse en la Tabla 4.

CONCLUSIONES

Como puede observarse, la mayor parte del coste del proyecto se centra en los recursos humanos utilizados para desarrollar el proyecto. Al dedicar el resultado final a la ayuda a un colectivo vulnerable para facilitar su inclusión social, no es de extrañar que vaya a ser amortizado ampliamente durante su uso. Además, la mayoría de los costes relacionados con el proyectista solo habría que abonarlos una vez ya que si se quisieran fabricar más sistemas de ayuda a invidentes no haría falta estudiar de nuevo los proyectos anteriores ni desarrollar la electrónica ni dedicar tanto tiempo a la fabricación de la placa de circuito impreso de nuevo. Sería tan sencillo como volver a fabricar el diseño final que se esté utilizando.

Este proyecto social resulta altamente rentable teniendo en cuenta todos los beneficios que aporta a los individuos que lo necesitan y los bajos costes que reporta si se va a producir en masa. Además, cuenta con motivaciones de carácter social para todas las personas implicadas ya que se ayuda a personas con necesidades especiales que de otra manera no podrían disfrutar de este deporte.

Si se fabricase este sistema a gran escala, produciendo multitud de unidades de una sola vez, podría industrializarse el proceso y abaratar aún más los costes de la producción.

ANEXO II: GUÍA DE INSTALACIÓN

En este anexo se va a explicar el proceso de instalación de los programas utilizados para desarrollar las partes de la programación del microprocesador y el diseño de la PCB en el proyecto.

De esta manera, si se desea continuar en otros cursos desarrollando más programas para el sistema, pueden utilizarse las mismas plataformas que se han usado hasta el momento en este trabajo y sus predecesores.

KiCAD

Este es el programa utilizado para el diseño de la placa de circuito impreso y sus conexiones. Es un programa de software libre, disponible para Windows, OSX y Linux, además de muchas distribuciones de Linux.

En este caso va a explicarse el proceso de instalación para ordenadores con sistemas operativos OSX, en concreto macOS Catalina versión 10.15.4, pero este proceso varía muy poco entre las diferentes versiones de OSX y entre los distintos sistemas operativos. Es un proceso de instalación sencillo e intuitivo.

En primer lugar, se accede a la página web oficial de KiCad, donde se encuentran accesibles todos los recursos disponibles. Entre estos recursos se puede encontrar, además de los enlaces para descargar el software, también ayuda para comenzar a usar el programa, donde se explica qué son las librerías y cómo es el espacio de trabajo básico.

Además, pueden descargarse distintas librerías y se puede donar dinero para ayudar a los desarrollos y financiar en alguna medida parte del proyecto.

Esta página es: <https://www.kicad-pcb.org/>. Nada más entrar en ese enlace se puede observar la pantalla de la Figura 40, que tiene en la parte superior una barra en la que se muestran

todas las ventanas que pueden visitarse. Esta pestaña también ofrece información sobre las distintas publicaciones que se han realizado en su blog y los distintos editores que tienen para desarrollar los proyectos.

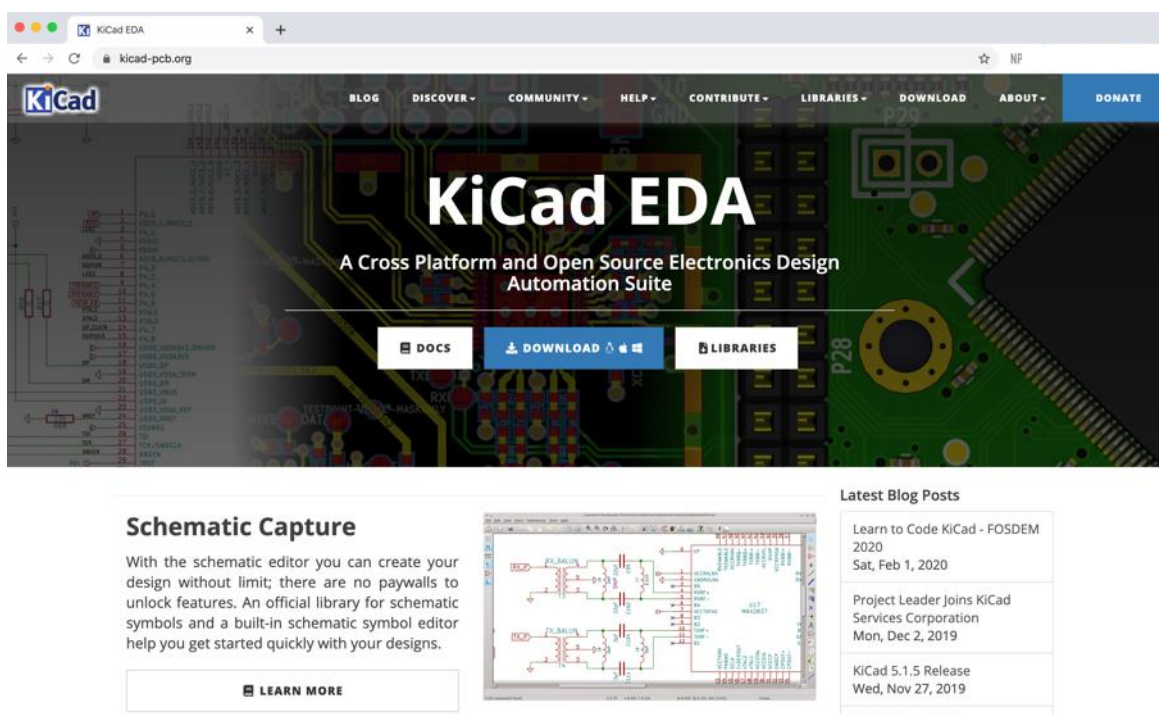


Figura 40: Página principal de KiCad

Estando en esta pestaña, se tiene que seleccionar la opción de descargar, que se encuentra en el medio de la pantalla. Al pinchar en este recuadro, se redirige al usuario automáticamente a la página de la Figura 41. En esta pantalla se proporciona al usuario una lista de todos los sistemas operativos para los cuales se encuentra disponible el sistema operativo.

La lista es muy amplia y cuenta con todo tipo de sistemas operativos, desde los más extendidos (Windows, Linux) hasta otras distribuciones de Linux más desconocidas por la mayoría de la sociedad (Debian).

Al ser de los programas que cuenta con recursos para un gran número de sistemas operativos, se convierte en la de las primeras opciones elegidas a la hora de desarrollar placas de circuito impreso.

Permite que un mismo diseño pueda actualizarse o trabajarse en distintos ordenadores con distintos sistemas operativos, de manera que no restringe al usuario a la hora de trabajar en equipos de trabajo o en proyectos con múltiples etapas distintas, desarrolladas cada una por una persona distinta.

La opción que se elige en este caso es el sistema operativo MacOS, como se indicó anteriormente. De esta manera, se vuelve a redirigir al usuario a una tercera pantalla, en la que se encuentran todos los ficheros disponibles para descargar el software. Esta pantalla puede observarse en la Figura 42.

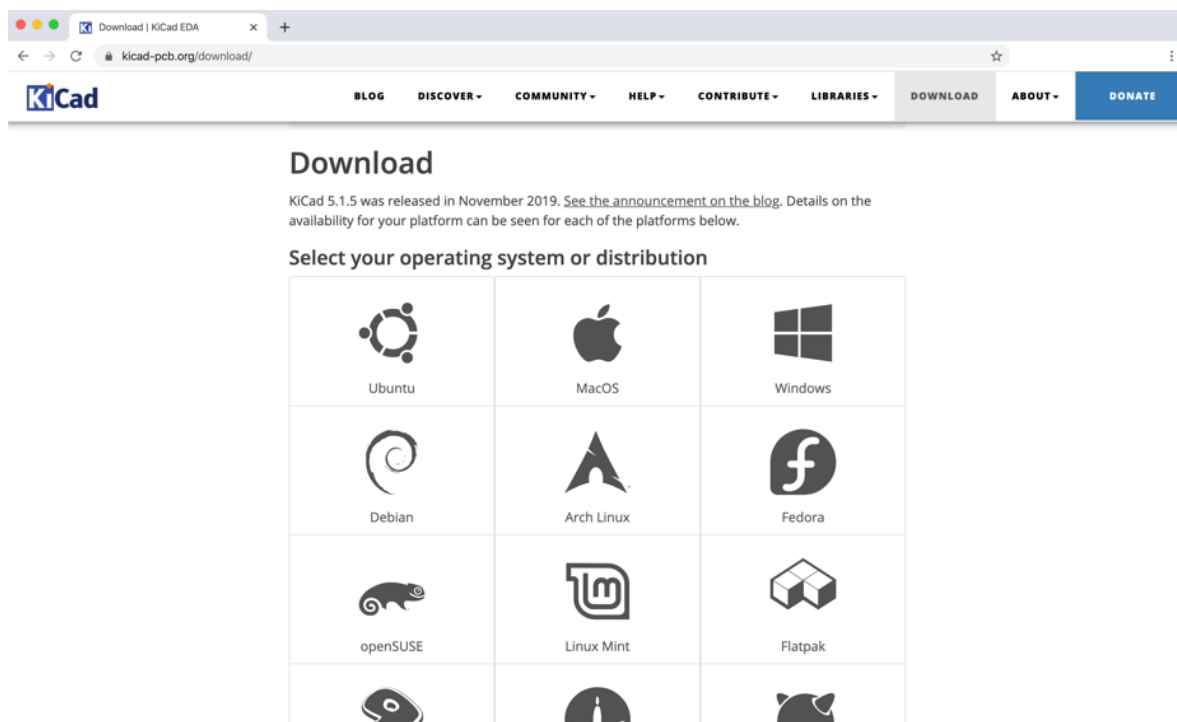


Figura 41: Ventana para elegir el sistema operativo

En esta tercera pantalla se ha seleccionado la opción de “CERN – Switzerland” para MacOS con versión posterior a la versión 10.14. Si se desciende por la misma página, pueden obtenerse versiones de KiCad desarrolladas para versiones posteriores de este sistema operativo. La versión más actual de KiCad, que es con la que se ha trabajado en este Trabajo de Fin de Grado, es la versión 5.1.5.

Automáticamente se descarga un archivo en el ordenador y se puede proceder a la instalación de este. Este archivo contiene el instalador para todas las partes de KiCad, desde plantillas hasta modelos 3D.

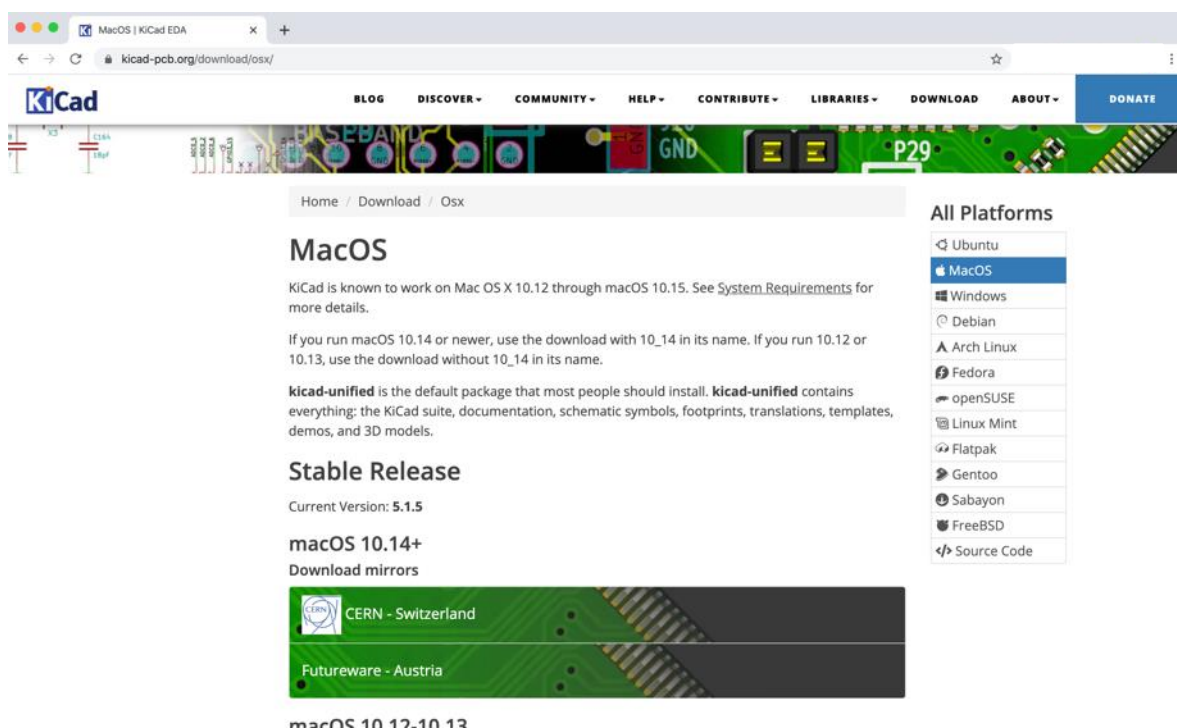


Figura 42: Ventana para descargar el programa

El archivo descargado tendrá un nombre igual o similar a “kicad-unified-5.1.5-0-10_14.dmg”. Este nombre variará en función de las versiones seleccionadas anteriormente.

Si se abre el archivo, después de una breve verificación por parte del ordenador de que está libre de elementos dañinos, aparece la pantalla de la Figura 43. En el archivo “README.txt” aparece toda la información necesaria referente a la instalación del programa.

En primer lugar, se describe brevemente en qué consiste KiCad. A continuación, se explica el proceso de instalación y el proceso de actualización.

Después, se explica el contenido de la carpeta “demos”, en la que hay varios proyectos de ejemplo que ofrecen como plantillas de manera gratuita para usarlos o adaptarlos.

Seguidamente explican las tres distintas versiones que existen para instalarse en MacOS, qué documentación existe sobre estos programas y dónde encontrarla, cómo indicar fallos y cómo desinstalarlo.

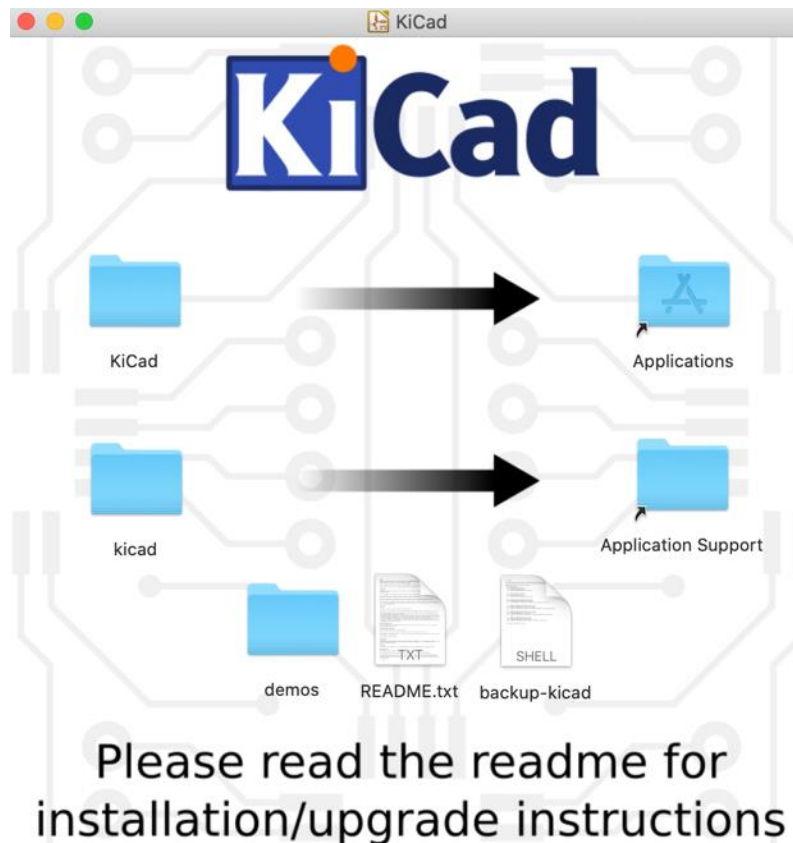


Figura 43: Ventana de instalación

Resumiendo este archivo, para instalar los programas necesarios solo hay que arrastrar la carpeta “KiCad” a la imagen de la carpeta “Applications” y la llamada “kicad” a “Application Support”.

Los programas finales se encontrarán en la ubicación de aplicaciones, que puede encontrarse usando el Finder.

Si es la primera vez que se abre la aplicación, el ordenador puede dar algún error indicando que no se puede abrir la aplicación porque se ha descargado de una fuente no segura (internet). Esto se soluciona haciendo clic derecho y seleccionando la opción abrir. Se

preguntará al usuario si realmente quiere abrirlo y después de indicar su conformidad no volverá a dar problemas.

MPLAB X IDE

La programación del microprocesador se ha realizado utilizando el programa MPLAB X IDE. Antes de comenzar a explicar el proceso de instalación del programa, se advierte de que, aunque se indica en su página oficial que el programa está disponible para Windows, MacOS y Linux, con la versión Catalina de MacOS a veces no es completamente compatible. El programa se instala, pero no es capaz de compilar ningún código y da errores continuamente.

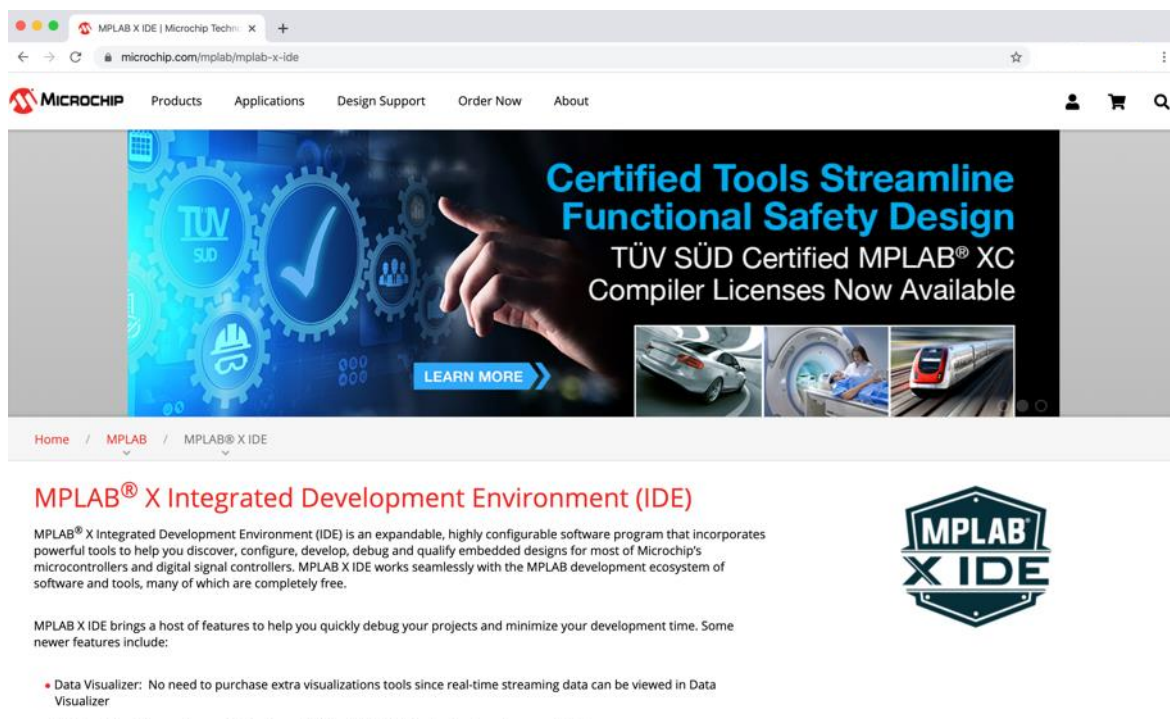


Figura 44: Página principal de MPLAB-X-IDE

Por ello, para realizar este proyecto en un ordenador con MacOS se recomienda hacer una partición del disco e instalar otro sistema operativo en la misma. De esta forma, el programa ya no daría problemas. Aún así, en la última actualización del programa disponible para descargar, los desarrolladores indican que han solucionado el problema.

Además, para poder compilar los programas se necesita un programador externo, que se vende por separado del microprocesador. El programador o depurador que se usa en este proyecto lleva incorporado el microprocesador PIC24FJ256GB106-1/PT.

Para descargar este programa, se debe entrar en el siguiente enlace: <https://www.microchip.com/mplab/mplab-x-ide>. La pantalla que aparece puede observarse en la Figura 44. Si se desciende en esta misma pantalla, se llega a la Figura 45.

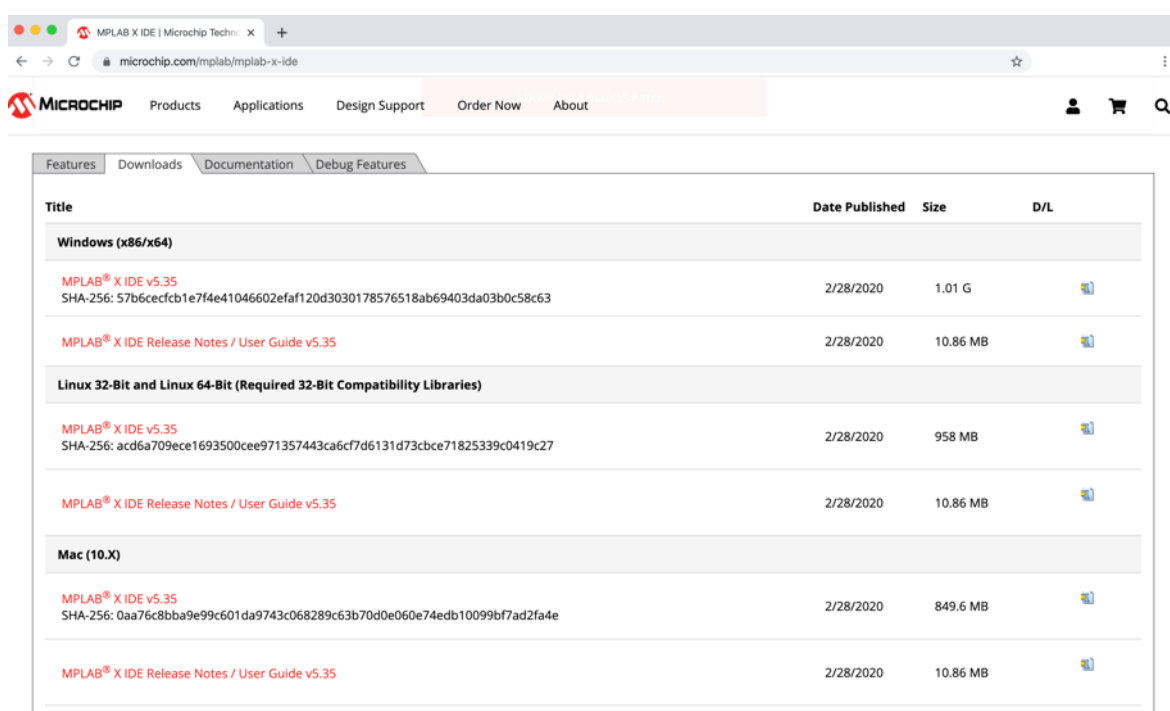


Figura 45: Opciones de descarga de MPLAB-X-IDE

En esta zona de la pantalla se encuentran los archivos que hay que descargar para que el programa funcione con los sistemas operativos indicados.

A continuación, se selecciona el archivo que se quiera descargar. Para realizar este anexo se ha descargado “MPLAB X IDE v5.35 para Mac (10.X)”. El otro archivo, “MPLAB X IDE Release Notes/ User Guide v5.35” contiene multitud de guías de usuario que pueden resultar muy útiles para sacarle el máximo provecho al software.

Cuando se termine la descarga, se tendrá un archivo de nombre “MPLABX-v5.35-osx-installer.dmg”. Al seleccionarlo, y después de una corta verificación por parte del ordenador, se obtiene la pantalla que se observa en la Figura 46.



Figura 46: Ventana de instalación de MPLAB-X-IDE

Al realizar doble clic sobre el icono que aparece en la pantalla aparece un aviso que indica que es una aplicación descargada de internet y pide al usuario verificar que quiere abrirla.



Figura 47: Confirmación de la instalación de MPLAB X IDE

Una vez se acepta este aviso, se debe introducir la clase de usuario para dar permiso para la instalación.

Después se abre el instalador final, que pide que se sigan unos pasos sencillos y se acepte el acuerdo de licencia de la Figura 48.

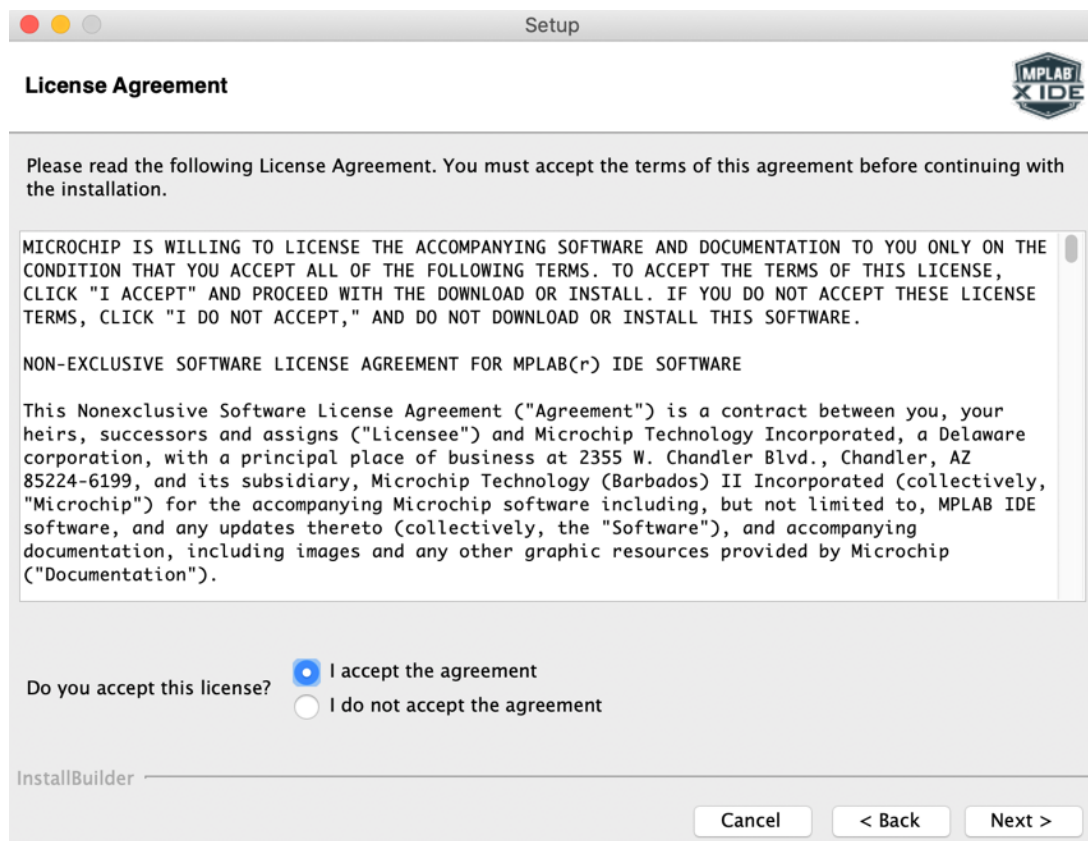


Figura 48: Acuerdo de licencia MPLAB X IDE

A partir de aquí es un proceso simple y que está guiado de manera sencilla por el instalador por lo que la instalación no debería suponer ningún problema para el usuario.

INDICACIONES FINALES

Para el desarrollo de este proyecto solo se han utilizado estos dos programas como software. Además de esto, sería muy recomendable disponer de un osciloscopio para poder realizar todas las pruebas con los sensores de ultrasonidos de manera efectiva ya que estas

herramientas sirven para saber la amplitud y frecuencia real de las ondas que se transmiten. En caso de que no se dispusiera de uno, se podría trabajar de igual manera con el sistema y desarrollar nuevos componentes, pero sería de gran dificultad comprobar que las ondas que se están transmitiendo y recibiendo son las esperadas.

ANEXO III: CÓDIGO DEL SISTEMA DESARROLLADO

En este anexo se adjunta el código completo del sistema desarrollado, que ya ha sido explicado en el Capítulo 7. Se divide en tres partes:

- En primer lugar, el código del microprocesador que va en el caballo del usuario invidente, en el caso de que este caballo solo contase con los sensores situados en la cabeza del caballo y el vibrador.
- En segundo lugar, el código del microprocesador del caballo delantero, en el caso de que este solo contase con los sensores situados en su grupa.
- Por último, el código de un microprocesador que cuenta con todos los desarrollados. Cuenta con los programas de los sensores delanteros y traseros (para comunicarse con dos caballos distintos, uno delante y otro detrás suya) y con los programas desarrollados por Miguel Ángel Sánchez para los sensores laterales. De estos últimos solo se añade en el anexo la parte referente al `main` y a los programas `tmr.c` y `tmr.h`, ya que el resto de los códigos de los sensores laterales fueron explicados y detallados por Miguel Ángel Sánchez en la memoria sobre su Proyecto de Fin de Grado y no han sido modificados en este proyecto.

El fichero `tmr.c` se ha copiado dentro del fichero del `main`, ya que interactúa de manera más directa. Introduciéndolo en este archivo se simplifican los procesos.

MICROPROCESADOR DEL CABALLO DEL USUARIO

CONFIG.H

```
// -----  
// ----- INTERFAZ DEL MODULO DE CONFIGURACIÓN -----  
// -----  
  
#ifndef _CONFIG_H  
#define _CONFIG_H  
  
// -----  
#include "p33FJ32MC202.h"  
  
// Frecuencia de operación del microprocesador (Hz)  
#define FCY 39613750  
  
// -----  
// ----- PROTOTIPOS DE LAS FUNCIONES PUBLICAS -----  
// -----  
  
void InicializarReloj(void);  
void RemapeaPerifericos(void);  
  
#endif
```

CONFIG.C

```
#include "p33FJ32MC202.h"  
#include "config.h"  
  
// -----  
// ----- FUNCIONES -----  
// -----  
  
/* Nombre: InicializarReloj  
 * Descripción: Inicializa el reloj interno FRC para que funcione con  
 * PLL.  
 * Argumentos: Ninguno  
 * Valor devuelto: Ninguno  
 */  
void InicializarReloj(void)  
{  
    // Configurar la frecuencia del oscilador FRC (FOSC), cuya frecuencia  
    // nominal (Fin) son 7.37 MHz, para que el microprocesador opere a 40 MIPS  
    // (FCY)  
    // FOSC = Fin * M/(N1*N2)      FCY = FOSC/2  
    // FOSC = 79.2275 MHz          FCY = 39.61375 MHz  
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1=2  
    PLLFBD             = 41; // Multiplicador del PLL: M=PLLFBD+2=43
```

```

        CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2

// Si hiciese falta llegar hasta las 40 MIPS de forma exacta, se podría
// hacer un ajuste fino de la frecuencia del reloj usando OSC_TUN para
// incrementar o decrementar la frecuencia a razón de OSC_TUN * 30 kHz.
// IMPORTANTE: Esta característica ha sido implementada para corregir
// derivas producidas a causa de la temperatura y no está garantizado que
// los incrementos sean de 30 kHz
/*
    OSC_TUN = 21; // Fin = 7.37 MHz + OSC_TUN * 30 kHz = 8 MHz
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1 = 2
    PLLFBD = 40; // Multiplicador del PLL: M = PLLFBD + 2 = 43
    CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2
*/

// Funciones para desbloquear la escritura del registro OSCCON
__builtin_write_OSCCONH(0x01); // Nuevo reloj: FRC w/ PLL
__builtin_write_OSCCONL(OSCCON | 0x01); // Iniciar cambio de reloj

while (OSCCONbits.COSC!=1); // Esperar que se produzca cambio reloj

while (OSCCONbits.LOCK!=1); // Esperar a que se sincronice el PLL
}

/* Nombre: RemapeaPerifericos
 * Descripción: Sitúa en los pines adecuados los periféricos del micro
 * Argumentos: Ninguno
 * Valor devuelto: Ninguno
 */

void RemapeaPerifericos(void)
{
    // Funciones para desbloquear la escritura del registro OSCCON
    __builtin_write_OSCCONL(OSCCON & 0xBF); // Desbloquea el PPS

    RPINR18bits.U1RXR = 5; // Asigna U1RX al pin 14 que es RP5
    RPOR2bits.RP4R = 3; // Asignar U1TX al pin 11 que es RP4

    __builtin_write_OSCCONL(OSCCON | 0x040); // Bloquea los PPS
}

```

EMISOR_DELANTE.H

```

/*
 * emisor_delante.h
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#ifndef EMISOR_DELANTE_H
#define EMISOR_DELANTE_H
#include "p33FJ32MC202.h"

```

```
void init_T2();  
void enviar_senal_delante();  
void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void);  
  
#endif /* EMISOR_DELANTE_H */
```

EMISOR_DELANTE.C

```
/*  
 * emisor_delante.c  
 * María del Carmen Rubiales Mena  
 * Última revisión: 8 de abril de 2020  
 */  
  
#include "emisor_delante.h"  
#include "p33FJ32MC202.h"  
  
static unsigned int contador = 0;  
static unsigned int contador2 = 0;  
static unsigned int DC = 12;  
static unsigned int periodo = 25;  
  
void init_T2(){  
  
    TMR2 = 0; // Inicio cuenta  
    PR2 = 40; // 1 us, final de cuenta  
    IFS0bits.T2IF = 0; // Pone a cero final de cuenta  
    IEC0bits.T2IE = 1; // Habilita interrupción  
    T2CON=0x8000; // Sin preescalado, on  
  
}  
  
void enviar_senal_delante(){  
  
    TRISB &= 0xEF;  
    PORTB &= 0xEF;  
    init_T2();  
  
}  
  
void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void){  
  
    if(contador2 < 50){//El periodo de 40 kHz es 25 us, 50 son dos ciclos  
        IFS0bits.T2IF = 0;  
        if(contador < DC){  
            PORTB |= 0x10;  
        }else{  
            PORTB &= 0xEF;  
            if(contador > (periodo - 1)){
```

```
        contador = 0;
    }
}

contador++;
contador2++;
}

if(contador2 > 50){
    contador2 = 0;
    contador = 0;
}
}
```

RECEPTOR_DELANTE.H

```
/*
 * receptor_delante.h
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#ifndef RECEPTOR_DELANTE_H
#define RECEPTOR_DELANTE_H
#include "p33FJ32MC202.h"

void init_receptor_delante(int canales);
int get_ad_delante(int canal);

#endif
```

RECEPTOR_DELANTE.C

```
/*
 * receptor_delante.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#include "p33FJ32MC202.h"
#include "receptor_delante.h"

void init_receptor_delante(int canales){
    TRISB |= canales & 0x3F >> 2 ;
    TRISA |= canales & 3 ;
    AD1PCFGL = ~ canales;
    AD1CON3 = 0x105;
    AD1CON2 = 0;
```

```
        AD1CON1 = 0x80E0;
        IPC3bits.AD1IP = 6;
    }

    int get_ad_delante(int canal){
        AD1CHS0 = canal;
        AD1CON1bits.SAMP = 1;
        IFS0bits.AD1IF = 0;

        while (!IFS0bits.AD1IF);
        return ADC1BUF0;
    }
```

TIEMPO.H

```
/*
 * tiempo.h
 * Autora: Mª del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#ifndef TIEMPO_H
#define TIEMPO_H
#include "p33FJ32MC202.h"

int calcular_distancia(int tiempo);

#endif /* TIEMPO_H */
```

TIEMPO.C

```
/*
 * tiempo.c
 * Autora: Mª del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#include "tiempo.h"
#include "p33FJ32MC202.h"

int calcular_distancia(int tiempo){

    int distancia_cm = 0;
    distancia_cm = 340*tiempo/100000/2; //Distancia en centímetros

    return distancia_cm;
}
```

VIBRADOR.H

```
/*
```



```
* vibrador.h
* Autora: Ma del Carmen Rubiales Mena
* Última revisión: 8 de abril de 2020
*/

#ifndef VIBRADOR_H
#define VIBRADOR_H
#include "p33FJ32MC202.h"

void init_vibra(unsigned int frec);
void set_dc_pwm(unsigned int dc);
void vib(unsigned int d1,unsigned int d2,unsigned int d3);

#endif
```

VIBRADOR.C

```
/*
* vibrador.c
* Autora: María del Carmen Rubiales Mena
* Última revisión: 8 de abril de 2020
*/

#include "p33FJ32MC202.h"
#include "vibrador.h"

/*
// -----
// ----- FUNCIONES -----
// -----
* Nombre: init_vibra
* Descripción: Inicializa el modulo PWM en los pines PWM2L1
* Argumentos: int Frecuencia en Hz
* Valor devuelto: Ninguno
*/

#define FCY 39613750

void init_vibra(unsigned int frec){
    TRISBbits.TRISB9 = 0;
    int preesc = FCY/frec/0x7FFF;
    if(preesc < 1) preesc = 0;//Configura el preescalado del timer
    else if (preesc < 4) preesc = 1;
    else if(preesc < 16) preesc = 2;
    else preesc = 3;
    P2TPER = FCY/frec/(1 << (preesc*2));
    PWM2CON1bits.PMOD1 = 1; //Independencia de PWM2L1 PWM2H1
    PWM2CON1bits.PEN1L = 1; //Hace que PWM2L1 sea controlado por PWM
    P2TCON = 0x8000 | (preesc << 2);//Habilita modulo
                                //Configura preescalado
}
```

```
/* Nombre: set_dc_pwm
 * Descripción: Configura valor del ciclo de trabajo del módulo PWM para
 * cada señal
 * int dc -- ciclo de trabajo en % * Valor devuelto: Ninguno
 */

void set_dc_pwm(unsigned int dc){
    //dc =(int)(dc * 25.0 / 100.0);
    if(dc > 25)//Tensión máx. de los vibradores 3V -> 12V*25%
        dc = 25;
    P2DC1 = dc*(2*P2TPER/100);
}

void vib(unsigned int d1,unsigned int d2,unsigned int d3){

return; }
```

MAIN.C

```
/*
 * main.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 8 de abril de 2020
 */

#include "p33FJ32MC202.h"
#include "emisor_delante.h"
#include "receptor_delante.h"
#include "tiempo.h"
#include "vibrador.h"
#include "config.h"

// -----
// ----- BITS DE CONFIGURACIÓN -----
// -----

// 1. Eliminar el segmento de arranque flash
// 2. Permitir la escritura del segmento de arranque flash
_FBS(BSS_NO_BOOT_CODE & BWRP_WRPTECT_OFF);

// 1. No proteger la memoria de programa contra escritura
// 2. No proteger el código
_FGS(GWRP_OFF & GCP_OFF);

// 1. Utilizar el oscilador interno (FRC) en el arranque
// 2. Arrancar directamente con el oscilador seleccionado
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// 1. Permitir la conmutación del reloj y deshabilitar la monitorización
// de fallos
// 2. Desactivar el oscilador primario
// 3. Utilizar el oscilador secundario como entrada y salida digital
// 4. Permitir múltiples remapeos de los pines
_FOSC(FCKSM_CSECMD & POSCMD_NONE & OSCIOFNC_ON & IOL1WAY_OFF);
```

```
// 1. Permitir la deshabilitación del watchdog timer
// poniendo a 0 el bit SWDTEN del registro RCON
_FWDT(FWDTEN_OFF);

// 1. Esperar 128 ms y resetear el microcontrolador al enchufar la
// alimentación
// 2. Controlar los pines de PWM desde el registro PORT al arrancar
// 3. Los pines PWM high están activos a nivel alto
// 4. Los pines PWM low están activos a nivel alto
// 5. Utilizar los pines estándar (SDA1 y SCL1) para el I2C
_FPOR(FPWRT_PWR128 & PWMPIN_ON & HPOL_ON & LPOL_ON & ALTI2C_OFF);

// 1. Programar y depurar a través de los pines PGC1 y PGD1
// 2. Desactivar el interfaz para JTAG
_FICD(ICS_PGD1 & JTAGEN_OFF);

//RB0 receptor delantero
//RB1 receptor trasero
//RB4 emisor delantero
//RB5 emisor trasero
//RB9 vibrador

static int tiempo = 0;
static int tiempo_seguro = 0;

void init_T3(){
    TMR3 = 0; // Inicio cuenta
    PR3 = 40; // 1 us, final de cuenta
    IFS0bits.T3IF = 0; // Pone a cero final de cuenta
    IEC0bits.T3IE = 1; // Habilita interrupción
    T2CON=0x8000; // Sin preescalado, on
}

void __attribute__((interrupt, no_auto_psv)) _T3Interrupt(void) {
    tiempo ++;
    tiempo_seguro ++;
    IFS0bits.T3IF = 0;
}

void espera(int milisegundos){
    int i;
    TMR1 = 0;
    PR1 = 40000;
    IFS0bits.T1IF = 0;
    T1CON = 0x8000;
    for(i=0;i<milisegundos;i++){
        while(IFS0bits.T1IF==0);
        IFS0bits.T1IF =0;
    }
    T1CON &=0x7FFF;
```

```
}

int main(void) {

    InicializarReloj();
    RemapeaPerifericos();
    AD1PCFGL=0xFFFF;

    PORTB = 0x0000;
    TRISB = 0xFFFF;

    init_receptor_delante(2);
    init_vibra(13000);
    init_T3();

    int distancia = 0;
    while(1) {

        set_dc_pwm(0);
        if(get_ad_delante(2)>200) {
            distancia = calcular_distancia(tiempo);
            if(distancia>17) {
                set_dc_pwm(5);
                espera(2000);
            }
            if(distancia<13) {
                set_dc_pwm(25);
                espera(2000);
            }
            tiempo = 0;
            tiempo_seguro = 0;
            enviar_senal_delante();
        }
        if(tiempo_seguro>10000) {
            enviar_senal_delante();
            tiempo = 0;
            tiempo_seguro = 0;
        }
    }

    return 0;
}
```

MICROPROCESADOR DEL CABALLO PRECEDENTE

CONFIG.H

```
// -----  
// ----- INTERFAZ DEL MODULO DE CONFIGURACIÓN -----  
// -----  
  
#ifndef _CONFIG_H  
#define _CONFIG_H  
  
// -----  
  
#include "p33FJ32MC202.h"  
  
// Frecuencia de operación del microprocesador (Hz)  
#define FCY 39613750  
  
// -----  
// ----- PROTOTIPOS DE LAS FUNCIONES PÚBLICAS -----  
// -----  
  
void InicializarReloj(void);  
void RemapeaPerifericos(void);  
  
#endif
```

CONFIG.C

```
#include "p33FJ32MC202.h"  
#include "config.h"  
  
// -----  
// ----- FUNCIONES -----  
// -----  
  
/* Nombre: InicializarReloj  
 * Descripción: Inicializa reloj interno FRC para que funcione con PLL.  
 * Argumentos: Ninguno  
 * Valor devuelto: Ninguno  
 */  
void InicializarReloj(void)  
{  
    // Configurar la frecuencia del oscilador FRC(FOSC), frecuencia nominal  
    // (Fin) son 7.37 MHz, para que el microprocesador opere a 40 MIPS (FCY)  
    // FOSC = Fin * M/(N1*N2)      FCY = FOSC/2  
    // FOSC = 79.2275 MHz      FCY = 39.61375 MHz  
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1=2  
    PLLFBD = 41; // Multiplicador del PLL: M=PLLFBD+2=43  
    CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2
```

```
// Si hiciese falta llegar hasta las 40 MIPS de forma exacta, se podría
// hacer un ajuste fino de la frecuencia del reloj usando OSCTUN para
// incrementar o decrementar la frecuencia a razón de OSCTUN * 30 kHz.
// IMPORTANTE: Esta característica implementada para corregir derivas
// producidas a causa de la temperatura y no está garantizado que los
// incrementos sean de 30 kHz
/*
    OSCTUN = 21; // Fin = 7.37 MHz + OSCTUN * 30 kHz = 8 MHz
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1 = 2
    PLLFBD = 40; // Multiplicador del PLL: M = PLLFBD + 2 = 43
    CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2
*/

// Funciones para desbloquear la escritura del registro OSCCON
__builtin_write_OSCCONH(0x01); //Nuevo reloj: FRC w/ PLL
__builtin_write_OSCCONL(OSCCON | 0x01); //Iniciar cambio de reloj

while (OSCCONbits.COSC != 1); //Esperar cambio reloj
while (OSCCONbits.LOCK != 1); //Esperar que se sincronice el PLL
}

/* Nombre: RemapeaPerifericos
 * Descripción: Sitúa en los pines adecuados los periféricos del micro
 * Argumentos: Ninguno
 * Valor devuelto: Ninguno
 */
void RemapeaPerifericos(void)
{

// Funciones para desbloquear la escritura del registro OSCCON
__builtin_write_OSCCONL(OSCCON & 0xBF); // Desbloquea el PPS

RPINR18bits.U1RXR = 5; // Asigna U1RX al pin 14 que es RP5
RPOR2bits.RP4R = 3; // Asignar U1TX al pin 11 que es RP4

__builtin_write_OSCCONL(OSCCON | 0x040); // Bloquea los PPS
}
}
```

RECEPTOR_TRASERO.H

```
/*
 * receptor_trasero.h
 * Autora: Mª del Carmen Rubiales Mena
 * Última revisión: 3 de abril de 2020
 */

#ifndef RECEPTOR_TRASERO_H
#define RECEPTOR_TRASERO_H
#include "p33FJ32MC202.h"

int get_ad_detras(int canal);
void init_sensor_detras(int canales);
```

```
#endif /* RECEPTOR_TRASERO_H */
```

RECEPTOR_TRASERO.C

```
/*
 * receptor_trasero.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 3 de abril de 2020
 */

#include "p33FJ32MC202.h"
#include "receptor_trasero.h"

void init_sensor_detras(int canales){
    TRISB |= canales & 0x3F >> 2 ;
    TRISA |= canales & 3 ;
    AD1PCFGL = ~ canales;
    AD1CON3 = 0x105;
    AD1CON2 = 0;
    AD1CON1 = 0x80E0;
    IPC3bits.AD1IP = 6;
}

int get_ad_detras(int canal){
    AD1CHS0 = canal;
    AD1CON1bits.SAMP = 1;
    IFS0bits.AD1IF = 0;

    while (!IFS0bits.AD1IF);
    return ADC1BUF0;
}
```

EMISOR_TRASERO.H

```
/*
 * emisor_trasero.h
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 3 de abril de 2020
 */

#ifndef EMISOR_TRASERO_H
#define EMISOR_TRASERO_H
#include "p33FJ32MC202.h"

void init_T1();
void init_emisor_trasero();
void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void);

#endif /* EMISOR_TRASERO_H */
```

EMISOR_TRASERO.C

```
/*
 * emisor_trasero.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 3 de abril de 2020
 */

#include "p33FJ32MC202.h"

static unsigned int contador = 0;
static unsigned int contador2 = 0;
static unsigned int DC = 12;
static unsigned int periodo = 25;

void init_T1() {
    TMR1 = 0; // Inicio cuenta
    PR1 = 40; // 1 us, final de cuenta
    IFS0bits.T1IF = 0; // Pone a cero final de cuenta
    IEC0bits.T1IE = 1; // Habilita interrupcion
    T1CON=0x8000; // Sin preescalado, on
}

void init_emisor_trasero() {
    PORTB &= 0xDF;
    TRISB &= 0xDF;
    init_T1();
}

void __attribute__((interrupt, no_auto_psv)) _T1Interrupt(void) {
    if(contador2 < 50) { //El periodo de 40 kHz es 25 us, 50 son dos ciclos
        IFS0bits.T1IF = 0;
        if(contador < DC) {
            PORTB |= 0x20;
        } else {
            PORTB &= 0xDF;
            if(contador > (periodo - 1)) {
                contador = 0;
            }
        }
        contador++;
        contador2++;
    }
}
```



```
if(contador2 > 50){  
    contador2 = 0;  
    contador = 0;  
}  
}
```

MAIN.C

```
/*  
 * main.c Programa principal del sistema  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 3 de abril de 2020  
 */  
  
#include "p33FJ32MC202.h"  
// -----  
// ----- BITS DE CONFIGURACIÓN -----  
// -----  
  
// 1. Eliminar el segmento de arranque flash  
// 2. Permitir la escritura del segmento de arranque flash  
_FBS(BSS_NO_BOOT_CODE & BWRP_WRPTECT_OFF);  
// 1. No proteger la memoria de programa contra escritura  
// 2. No proteger el código  
_FGS(GWRP_OFF & GCP_OFF);  
// 1. Utilizar el oscilador interno (FRC) en el arranque  
// 2. Arrancar directamente con el oscilador seleccionado  
_FOSSEL(FNOSC_FRC & IESO_OFF);  
// 1. Permitir la conmutación del reloj y deshabilitar la monitorización  
// de fallos  
// 2. Desactivar el oscilador primario  
// 3. Utilizar el oscilador secundario como entrada y salida digital  
// 4. Permitir múltiples remapeos de los pines  
_FOSC(FCKSM_CSECMD & POSCMD_NONE & OSCIOFNC_ON & IOL1WAY_OFF);  
// 1. Permitir la deshabilitación del watchdog timer  
// poniendo a 0 el bit SWDTEN del registro RCON  
_FWDI(FWDIEN_OFF);  
// 1. Esperar 128 ms y resetear el microcontrolador al enchufar la  
// alimentación  
// 2. Controlar los pines de PWM desde el registro PORT al arrancar  
// 3. Los pines PWM high están activos a nivel alto  
// 4. Los pines PWM low están activos a nivel alto  
// 5. Utilizar los pines estándar (SDA1 y SCL1) para el I2C  
_FPOR(FPWRT_PWR128 & PWMPIN_ON & HPOL_ON & LPOL_ON & ALTI2C_OFF);  
// 1. Programar y depurar a través de los pines PG1 y PG1  
// 2. Desactivar el interfaz para JTAG  
_FICD(ICS_PG1 & JTAGEN_OFF);  
  
#include "config.h"  
#include "emisor_trasero.h"  
#include "receptor_trasero.h"
```

```
int main(void) {  
  
    InicializarReloj();  
    RemapeaPeriféricos();  
    AD1PCFGL=0xFFFF;  
  
    PORTB = 0x0000;  
    TRISB = 0x0001;  
  
    init_sensor(3);  
  
    while(1) {  
  
        if(get_ad_detras(3) >= 200) {  
            init_emisor_trasero();  
        }  
  
    }  
  
    return 0;  
}
```

MICROPROCESADOR PARA TODOS LOS SENSORES

CONFIG.H

```
// -----  
// ----- INTERFAZ DEL MODULO DE CONFIGURACIÓN -----  
// -----  
  
#ifndef _CONFIG_H  
#define _CONFIG_H  
  
// -----  
  
#include "p33FJ32MC202.h"  
  
// Frecuencia de operación del microprocesador (Hz)  
#define FCY 39613750  
  
// -----  
// ----- PROTOTIPOS DE LAS FUNCIONES PÚBLICAS -----  
// -----  
  
void InicializarReloj(void);  
void RemapeaPerifericos(void);  
  
#endif
```

CONFIG.C

```
#include "p33FJ32MC202.h"  
#include "config.h"  
  
// -----  
// ----- FUNCIONES -----  
// -----  
  
/* Nombre: InicializarReloj  
 * Descripción: Inicializa reloj interno FRC para que funcione con PLL.  
 * Argumentos: Ninguno  
 * Valor devuelto: Ninguno  
 */  
void InicializarReloj(void)  
{  
    // Configurar la frecuencia del oscilador FRC(FOSC), frecuencia nominal  
    // (Fin) son 7.37 MHz, para que el microprocesador opere a 40 MIPS (FCY)  
    // FOSC = Fin * M/(N1*N2)      FCY = FOSC/2  
    // FOSC = 79.2275 MHz        FCY = 39.61375 MHz  
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1=2  
    PLLFBD            = 41; // Multiplicador del PLL: M=PLLFBD+2=43
```

```

        CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2

// Si hiciese falta llegar hasta las 40 MIPS de forma exacta, se podría
// hacer un ajuste fino de la frecuencia del reloj usando OSCTUN para
// incrementar o decrementar la frecuencia a razón de OSCTUN * 30 kHz.
// IMPORTANTE: Esta característica implementada para corregir derivas
// producidas a causa de la temperatura y no está garantizado que los
// incrementos sean de 30 kHz
/*
    OSCTUN = 21; // Fin = 7.37 MHz + OSCTUN * 30 kHz = 8 MHz
    CLKDIVbits.PLLPRE = 0; // Preescalado del PLL: N1 = 2
    PLLFBD = 40; // Multiplicador del PLL: M = PLLFBD + 2 = 43
    CLKDIVbits.PLLPOST = 0; // Postescalado del PLL: N2 = 2
*/

// Funciones para desbloquear la escritura del registro OSCCON
__builtin_write_OSCCONH(0x01); //Nuevo reloj: FRC w/ PLL
__builtin_write_OSCCONL(OSCCON | 0x01); //Iniciar cambio de reloj

while (OSCCONbits.COSC != 1); //Esperar cambio reloj
while (OSCCONbits.LOCK != 1); //Esperar que se sincronice el PLL
}

/* Nombre: RemapeaPerifericos
 * Descripción: Sitúa en los pines adecuados los periféricos del micro
 * Argumentos: Ninguno
 * Valor devuelto: Ninguno
 */
void RemapeaPerifericos(void)
{

// Funciones para desbloquear la escritura del registro OSCCON
__builtin_write_OSCCONL(OSCCON & 0xBF); // Desbloquea el PPS

RPNR18bits.U1RXR = 5; // Asigna U1RX al pin 14 que es RP5
RPOR2bits.RP4R = 3; // Asignar U1TX al pin 11 que es RP4

__builtin_write_OSCCONL(OSCCON | 0x040); // Bloquea los PPS
}

```

RECEPTOR_DELANTE.H

```

/*
 * receptor_delante.h
 * Autores: Mª del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#ifndef RECEPTOR_H
#define RECEPTOR_H
#include "p33FJ32MC202.h"

```

```
void init_receptor_delante(int canales);  
int get_ad_delante(int canal);  
  
#endif
```

RECEPTOR_DELANTE.C

```
/*  
 * receptor_delante.c  
 * Autora: Ma del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#include "p33FJ32MC202.h"  
#include "receptor_delante.h"  
  
void init_receptor_delante(int canales){  
    TRISB |= canales & 0x3F >> 2 ;  
    TRISA |= canales & 3 ;  
    AD1PCFGL = ~ canales;  
    AD1CON3 = 0x105;  
    AD1CON2 = 0;  
    AD1CON1 = 0x80E0;  
    IPC3bits.AD1IP = 6;  
}  
  
int get_ad_delante(int canal){  
    AD1CHS0 = canal;  
    AD1CON1bits.SAMP = 1;  
    IFS0bits.AD1IF = 0;  
  
    while (!IFS0bits.AD1IF);  
    return ADC1BUF0;  
}
```

RECEPTOR_TRASERO.H

```
/*  
 * receptor_trasero.h  
 * Autora: Ma del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#ifndef RECEPTOR_TRASERO_H  
#define RECEPTOR_TRASERO_H  
#include "p33FJ32MC202.h"  
  
int get_ad_trasero(int canal);  
void init_sensor_trasero(int canales);  
  
#endif /* RECEPTOR_TRASERO_H */
```

RECEPTOR_TRASERO.C

```
/*
 * receptor_trasero.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#include "p33FJ32MC202.h"
#include "receptor_trasero.h"

void init_sensor_trasero(int canales){
    TRISB |= canales & 0x3F >> 2 ;
    TRISA |= canales & 3 ;
    AD1PCFGL = ~ canales;
    AD1CON3 = 0x105;
    AD1CON2 = 0;
    AD1CON1 = 0x80E0;
    IPC3bits.AD1IP = 6;
}

int get_ad_trasero(int canal){
    AD1CHS0 = canal;
    AD1CON1bits.SAMP = 1;
    IFS0bits.AD1IF = 0;

    while (!IFS0bits.AD1IF);
    return ADC1BUF0;
}
```

EMISOR_DELANTE.H

```
/*
 * emisor_delante.h
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#ifndef EMISOR_DELANTE_H
#define EMISOR_DELANTE_H
#include "p33FJ32MC202.h"

void init_T2();
void enviar_senal_delante();
void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void);

#endif /* EMISOR_DELANTE_H */
```

EMISOR_DELANTE.C

```
/*
 * emisor_delante.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#include "emisor_delante.h"
#include "p33FJ32MC202.h"

static unsigned int contador = 0;
static unsigned int contador2 = 0;
static unsigned int DC = 12;
static unsigned int periodo = 25;

void init_T2() {

    TMR2 = 0; // Inicio cuenta
    PR2 = 40; // 1 us, final de cuenta
    IFS0bits.T2IF = 0; // Pone a cero final de cuenta
    IEC0bits.T2IE = 1; // Habilita interrupcion
    T2CON=0x8000; // Sin preescalado, on
}

void enviar_senal_delante() {

    TRISB &= 0xEF;
    PORTB &= 0xEF;
    init_T2();
}

void __attribute__((interrupt, no_auto_psv)) _T2Interrupt(void) {

    if(contador2 < 50){ //El periodo de 40 kHz es 25 us,50 son dos ciclos
        IFS0bits.T2IF = 0;
        if(contador < DC){
            PORTB |= 0x10;
        }else{
            PORTB &= 0xEF;
            if(contador > (periodo - 1)){
                contador = 0;
            }
        }
        contador++;
        contador2++;
    }
}
```

```
if(contador2 > 50){  
    contador2 = 0;  
    contador = 0;  
}  
}
```

EMISOR_TRASERO.H

```
/*  
 * emisor_trasero.h  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#ifndef EMISOR_TRASERO_H  
#define EMISOR_TRASERO_H  
#include "p33FJ32MC202.h"  
  
void init_T3();  
void init_emisor_trasero();  
void __attribute__((interrupt, no_auto_psv)) _T3Interrupt(void);  
  
#endif /* EMISOR_TRASERO_H */
```

EMISOR_TRASERO.C

```
/*  
 * emisor_trasero.c  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#include "p33FJ32MC202.h"  
#include "emisor_trasero.h"  
  
static unsigned int contador = 0;  
static unsigned int contador2 = 0;  
static unsigned int DC = 12;  
static unsigned int periodo = 25;  
  
void init_T3(){  
  
    TMR3 = 0; // Inicio cuenta  
    PR3 = 40; // 1 us, final de cuenta  
    IFS0bits.T3IF = 0; // Pone a cero final de cuenta  
    IEC0bits.T3IE = 1; // Habilita interrupción  
    T3CON=0x8000; // Sin preescalado, on  
  
}
```



```
void init_emisor_trasero() {

    PORTB &= 0xDF;
    TRISB &= 0xDF;
    init_T3();
}

void __attribute__((interrupt, no_auto_psv)) _T3Interrupt(void) {

    if(contador2 < 50){ //El periodo de 40 kHz es 25 us, 50 son dos ciclos
        IFS0bits.T3IF = 0;
        if(contador < DC){
            PORTB |= 0x20;
        }else{
            PORTB &= 0xDF;
            if(contador > (periodo - 1)){
                contador = 0;
            }
        }
        contador++;
        contador2++;
    }

    if(contador2 > 50){
        contador2 = 0;
        contador = 0;
    }
}
```

VIBRADOR.H

```
/*
 * vibrador.h
 * Autora: Mª del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#ifndef VIBRADOR_H
#define VIBRADOR_H
#include "p33FJ32MC202.h"

void init_vibra(unsigned int frec);
void set_dc_vibrador_trasero(unsigned int dc);
void vib(unsigned int d1, unsigned int d2, unsigned int d3);

#endif
```

VIBRADOR.C

```
/*
 * vibrador.c
 * Autora: Ma del Carmen Rubiales Mena
 * Última revisión: 12 de abril de 2020
 */

#include "p33FJ32MC202.h"
#include "vibrador.h"

/*
// -----
// ----- FUNCIONES -----
// -----
 * Nombre: init_vibra
 * Descripción: Inicializa el modulo PWM en los pines PWM2L1
 * Argumentos: int Frecuencia en Hz
 * Valor devuelto: Ninguno
 */

#define FCY 39613750

void init_vibra(unsigned int frec){
    TRISBbits.TRISB9 = 0;
    int preesc = FCY/frec/0x7FFF;
    if(preesc < 1) preesc = 0; //configura preescalado del timer
    else if (preesc < 4) preesc = 1;
    else if(preesc < 16) preesc = 2;
    else preesc = 3;
    P2TPER = FCY/frec/(1 << (preesc*2));
    PWM2CON1bits.PMOD1 = 1; // independencia de PWM2L1 PWM2H1
    PWM2CON1bits.PEN1L = 1; // hace que PWM2L1 sea controlado por PWM
    P2TCON = 0x8000 | (preesc << 2);
    //habilita módulo y configura preescalado
}

/* Nombre: set_dc_pwm
 * Descripción: Configura el valor del ciclo de trabajo de un módulo PWM
 * para cada señal int dc -- ciclo de trabajo en %
 * Valor devuelto: Ninguno
 */

void set_dc_vibrador_trasero(unsigned int dc){
    //dc =(int) (dc * 25.0 / 100.0);

    if(dc > 25) //Tension max de los vibradores 3V -> 12V*25%
        dc = 25;
    P2DC1 = dc*(2*P2TPER/100);
}
```

```
void vib(unsigned int d1,unsigned int d2,unsigned int d3){  
  
return; }
```

TIEMPO.H

```
/*  
 * tiempo.h  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#ifndef TIEMPO_H  
#define TIEMPO_H  
#include "p33FJ32MC202.h"  
  
int calcular_distancia(int tiempo);  
  
#endif /* TIEMPO_H */
```

TIEMPO.C

```
/*  
 * tiempo.c  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */  
  
#include "tiempo.h"  
#include "p33FJ32MC202.h"  
  
int calcular_distancia(int tiempo){  
  
    int distancia_cm = 0;  
    distancia_cm = 340*tiempo/100000/2; //Distancia en centímetros  
  
    return distancia_cm;  
}
```

MAIN.C

```
/*  
 * main.c  
 * Autora: Mª del Carmen Rubiales Mena  
 * Última revisión: 12 de abril de 2020  
 */
```

```
#include "p33FJ32MC202.h"
#include "emisor_delante.h"
#include "receptor_delante.h"
#include "tiempo.h"
#include "vibrador.h"
#include "config.h"
#include "tmr.h"
#include "pwm.h"
#include "adc.h"
#include "sensors.h"
#include "values.h"

// -----
// ----- BITS DE CONFIGURACIÓN -----
// -----

// 1. Eliminar el segmento de arranque flash
// 2. Permitir la escritura del segmento de arranque flash
_FBS(BSS_NO_BOOT_CODE & BWRP_WRPTECT_OFF);

// 1. No proteger la memoria de programa contra escritura
// 2. No proteger el código
_FGS(GWRP_OFF & GCP_OFF);

// 1. Utilizar el oscilador interno (FRC) en el arranque
// 2. Arrancar directamente con el oscilador seleccionado
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// 1. Permitir conmutación reloj y deshabilitar monitorización de fallos
// 2. Desactivar el oscilador primario
// 3. Utilizar el oscilador secundario como entrada y salida digital
// 4. Permitir múltiples remapeos de los pines
_FOSC(FCKSM_CSECMD & POSCMD_NONE & OSCIOFNC_ON & IOL1WAY_OFF);

// 1. Permitir la deshabilitación del watchdog timer
// poniendo a 0 el bit SWDTEN del registro RCON
_FWDT(FWDTEN_OFF);

// 1. Esperar 128 ms, resetear microcontrolador al enchufar alimentación
// 2. Controlar los pines de PWM desde el registro PORT al arrancar
// 3. Los pines PWM high están activos a nivel alto
// 4. Los pines PWM low están activos a nivel alto
// 5. Utilizar los pines estándar (SDA1 y SCL1) para el I2C
_FPOR(FPWRT_PWR128 & PWMPIN_ON & HPOL_ON & LPOL_ON & ALTI2C_OFF);

// 1. Programar y depurar a través de los pines PGC1 y PGD1
// 2. Desactivar el interfaz para JTAG
_FICD(ICS_PGD1 & JTAGEN_OFF);

//Pines usados para los sensores de distancia delantero y trasero:
```

```
//RB0 receptor delantero
//RB1 receptor trasero
//RB4 emisor delantero
//RB5 emisor trasero
//RB9 vibrador

//Pines usados en los sensores laterales:

//RB11 - A - RB3
//RB14 - B - RA0
//RB12 - C - RA1
//RB2 - POTENCIOMETRO
//RB13 - vib1
//RB15 - vib2

//static unsigned int miliseg = 0;//cuenta de los milisegundos
static unsigned int microseg = 0;

// -----
// ----- FUNCIONES -----
// -----

/* Nombre: init_tmr
 * Descripción: Inicializa el timer a lms
 * Argumentos: Ninguno
 * Valor devuelto: Ninguno
 */

void init_tmr(void){
    TMR1 = 0; //cuenta a 0
    PR1 = 40;// contar hasta 40, interrupción salta cada microsegundo
    IEC0bits.T1IE = 1; //activar interrupciones
    IFS0bits.T1IF = 0; //bit de interrupción de fin de cuenta
    T1CON = 0X8000; // preescalado 0 y activado
}

/* Nombre: _T1Interrupt
 * Descripción: Interrupción del timer
 * Argumentos: Ninguno
 * Valor devuelto: Ninguno
 */

void __attribute__((interrupt,no_auto_psv)) _T1Interrupt(void){
    IFS0bits.T1IF = 0;
    microseg++;
    tiempo ++;
    tiempo_seguro ++;
    if(microseg >= 10000)
        microseg = 0;
}
```

```
/* Nombre: espera
 * Descripción: pausa el programa el tiempo indicado
 * Argumentos: tiempo en microsegundos max 10000microseg
 * Valor devuelto: Ninguno
 */

void wait(unsigned int tiempo){ unsigned int cuenta;
    cuenta = microseg + tiempo;
    if(cuenta >= 10000)
        cuenta -= 10000;
    while(microseg < cuenta);
}

/* Nombre: millis
 * Descripción: devuelve el valor de microseg
 * Argumentos: Ninguno
 * Valor devuelto: microseg
 */

unsigned int micros(void){
    return microseg;
}

static int tiempo = 0;
static int tiempo_seguro = 0;

int map(int input){
    double slope, output;
    int input_start, input_end, output_start, output_end;
    input_start = MIN_VALUE;
    input_end = MAX_VALUE;
    output_start = 25;
    output_end = 0;
    slope = 1.0 * (output_end-output_start)/(input_end-input_start);
    output = output_start + slope * (input - input_start);
    return (int)output;
}

int main(void){

    InicializarReloj();
    RemapeaPerifericos();
    AD1PCFGL=0xFFFF;

    PORTB = 0x0000;
    TRISB = 0xFFFF;

    int distancia = 0;
    unsigned int potenciometro;
    int dist[2]; //distancias en microsegundos int pwm_dc[2];
    int i;
```

```
init_receptor_delante(2);
init_vibra(13000);
init_emisor_trasero();
enviar_senal_delante();
init_sensors();
init_tmr();
init_pwm(PWM_FREQ);
init_ad();
init_sensor_trasero(3);

while(1){

    //Sensor delantero
    set_dc_vibrador_trasero(0);
    if(get_ad_delante(2)>200){
        distancia = calcular_distancia(tiempo);
        if(distancia>17){
            set_dc_vibrador_trasero(5);
            wait(2000);
        }

        if(distancia<13){
            set_dc_vibrador_trasero(25);
            wait(2000);
        }

        tiempo = 0;
        tiempo_seguro = 0;
        enviar_senal_delante();
    }

    if(tiempo_seguro>10000){
        enviar_senal_delante();
        tiempo = 0;
        tiempo_seguro = 0;
    }

    //Sensor trasero
    if(get_ad_trasero(3) >= 200){
        init_emisor_trasero();
    }

    //Sensores laterales
    potenciómetro = get_ad();
    set_dc_pwm(1,0); //parar conmutaciones del pwm para los sensores
    set_dc_pwm(2,0);

    for(i = 0; i < 2; i++){

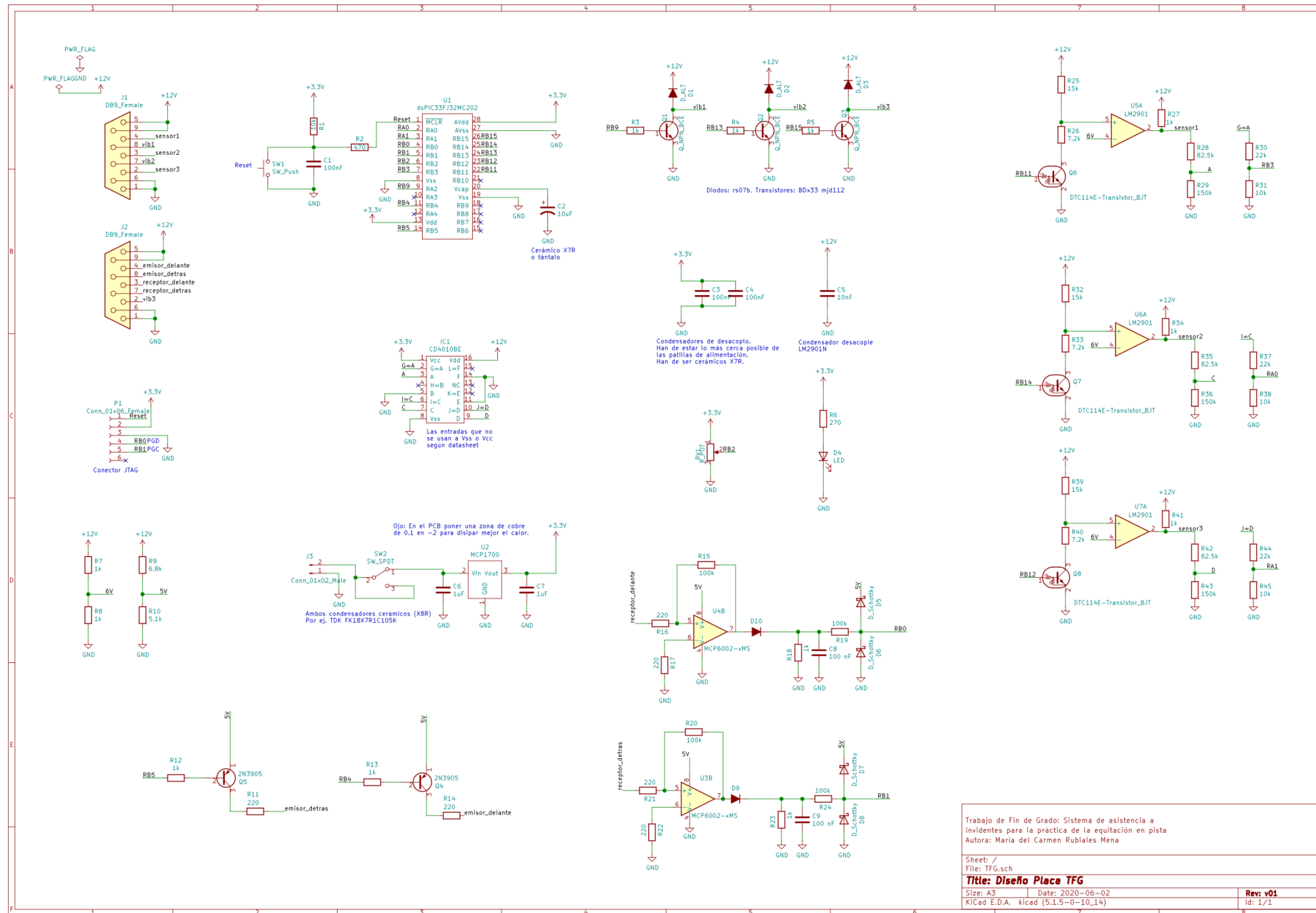
        dist[i] = rollingAverage(get_sensor(i+1),i+1);
        pwm_dc[i] = map(dist[i]);
        if(pwm_dc[i] < 0)
            pwm_dc[i] = 0;
        wait(2);
    }
}
```

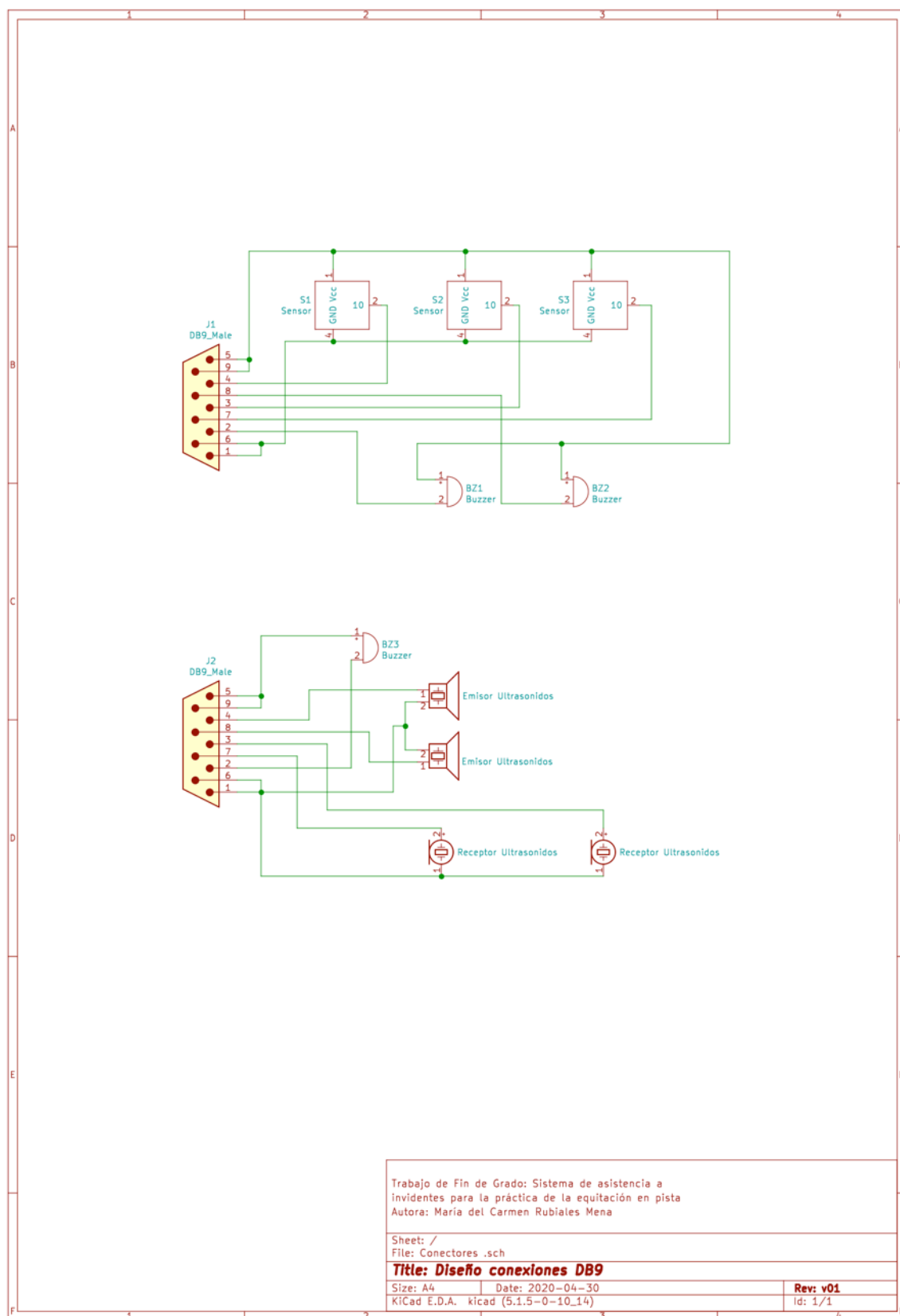
```
set_dc_pwm(1, (int)pwm_dc[0]*potenciometro/0x3FF);  
set_dc_pwm(2, (int)pwm_dc[1]*potenciometro/0x3FF);  
  
for(i = 0; i<1000;i++){  
    wait(1000);}  
  
return 0;}
```

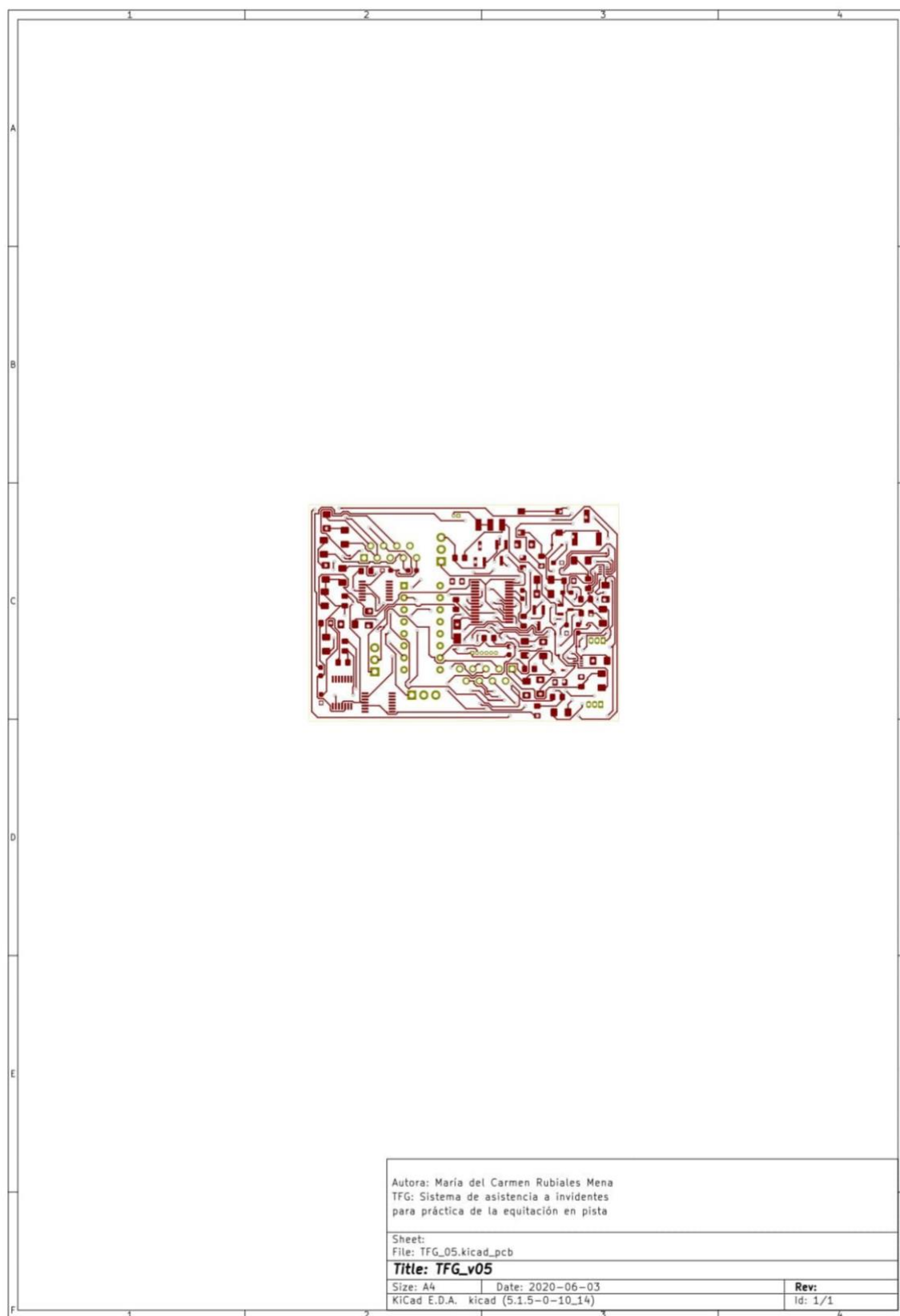

ANEXO IV: DISEÑO DE LA PLACA

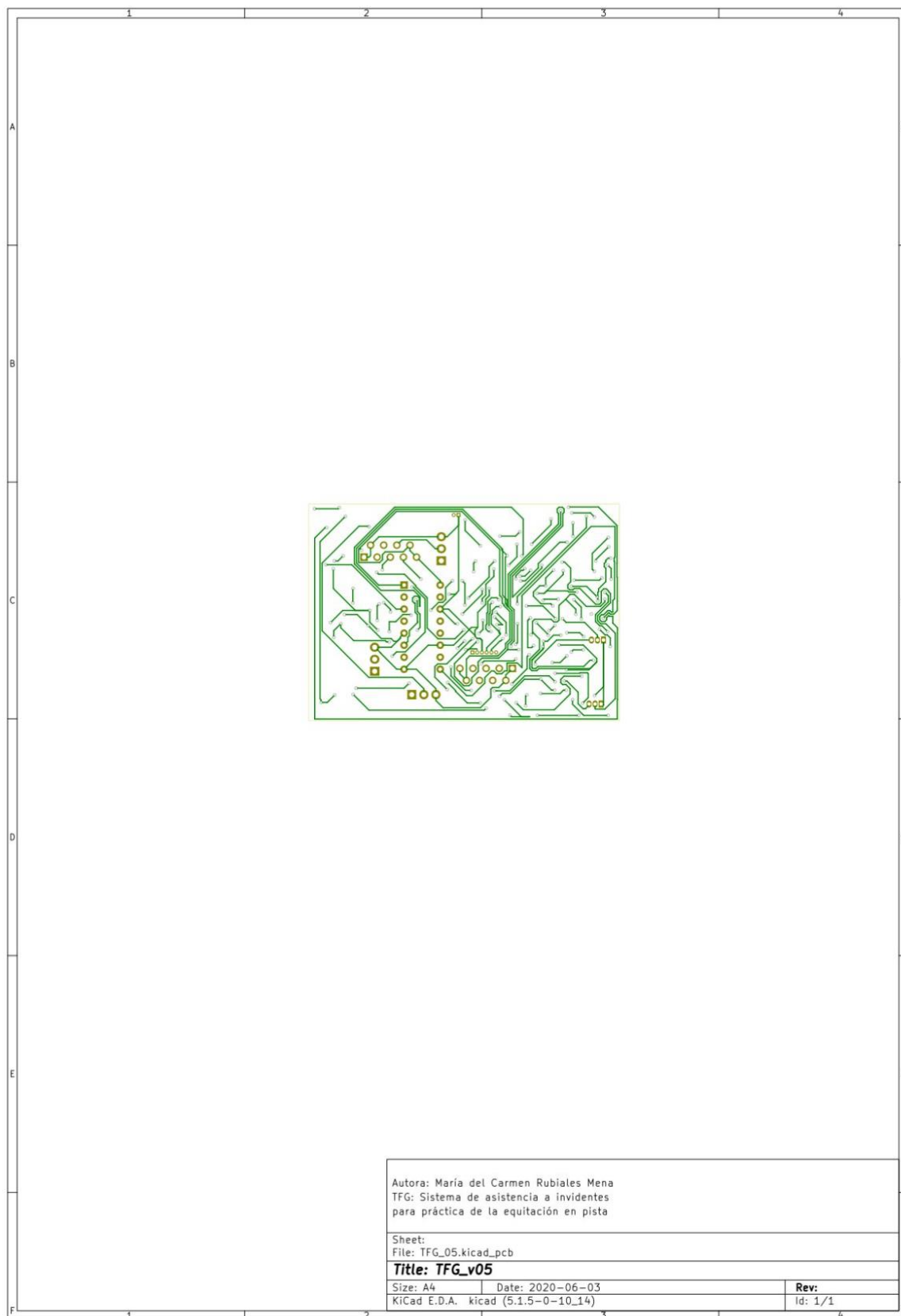
En este anexo se adjuntan los planos referentes al diseño de la placa de circuito impreso (PCB). En esta placa se fusiona el proyecto anterior con el actual, por lo que cuenta con los circuitos correspondientes a los sensores laterales, el sensor trasero y el sensor delantero. Los planos que se encuentran en este anexo, por orden de aparición, son:

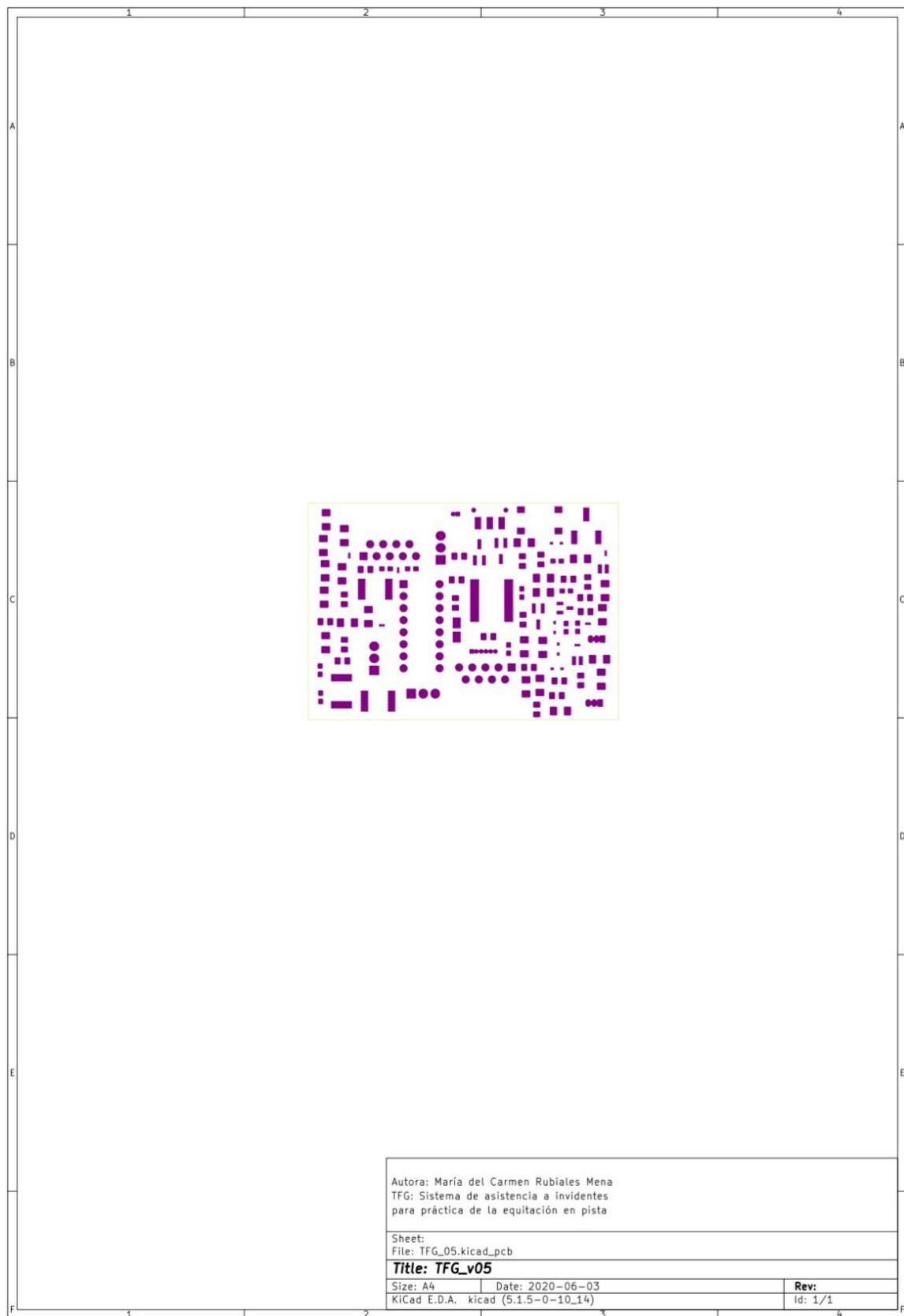
1. Plano con todos los circuitos que se encuentran en la PCB, incluyendo los desarrollados en este trabajo y en el anterior.
2. Plano de las conexiones de los sensores y vibradores a dos conectores de tipo DB9 (macho). Estos se unen a su vez con otros dos conectores DB9 (hembra) que se encuentran en la PCB.
3. Planos pertinentes al diseño final de la placa.

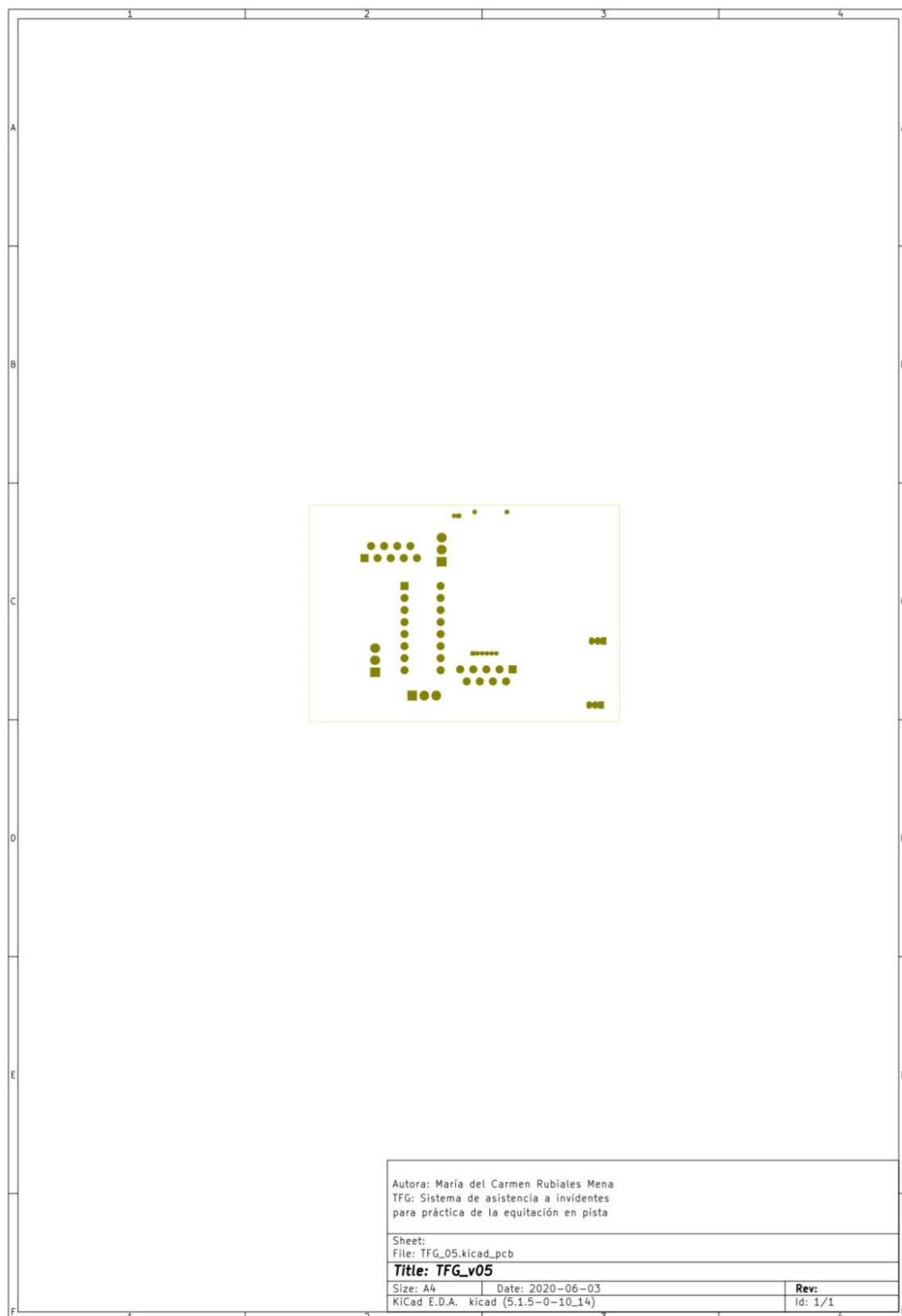


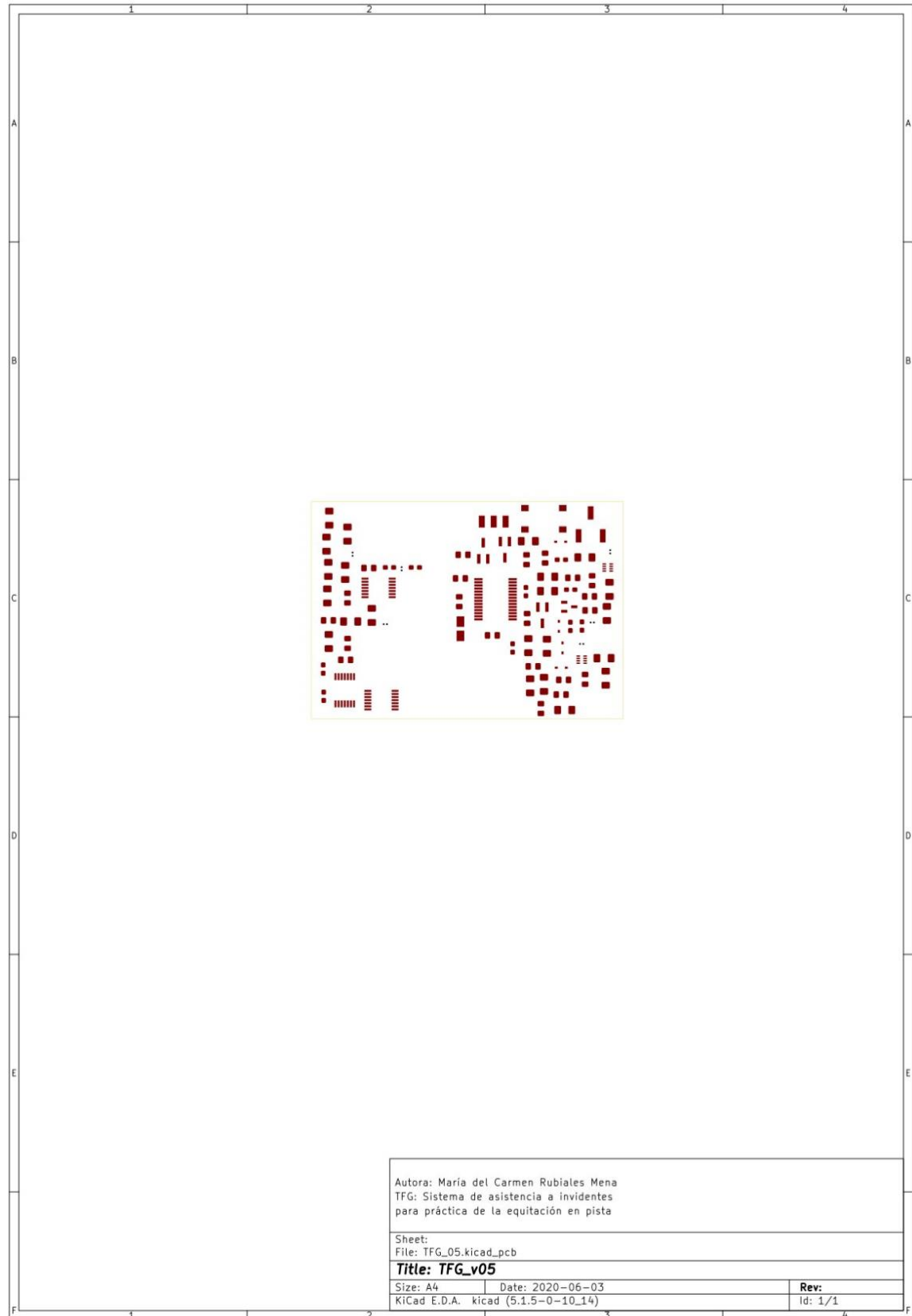


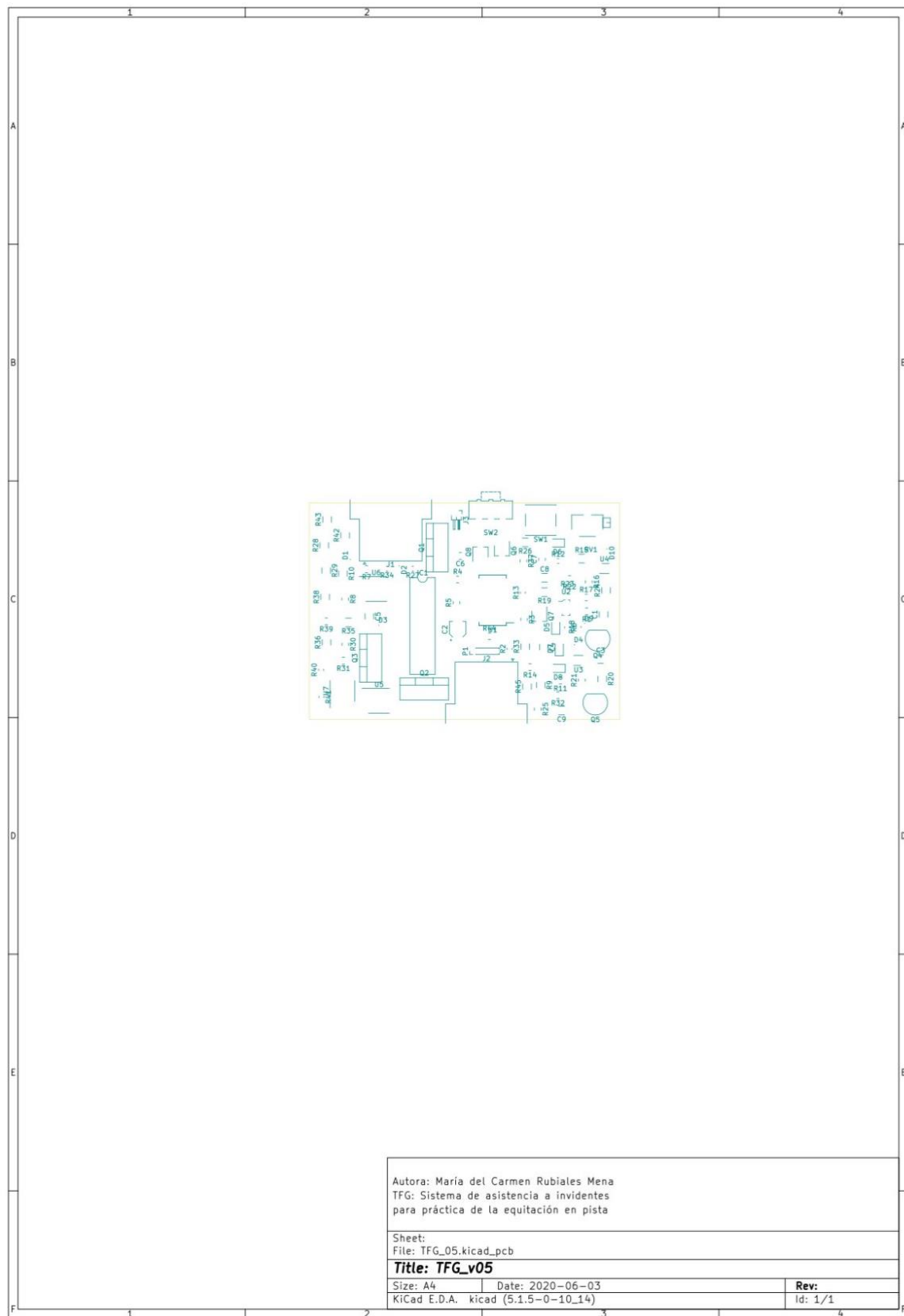












ANEXO V: DATASHEETS DE LOS COMPONENTES

En este anexo se adjuntan los datasheets de los componentes más destacables del proyecto, es decir, del emisor y del receptor de ultrasonidos, del actuador utilizado y del microprocesador dsPIC33FJ32MC202. De este último se añaden solo las primeras páginas relativas a sus características generales y a su esquema con los pines y sus nombres. El texto completo y detallado puede encontrarse en el siguiente enlace <https://www.microchip.com/wwwproducts/en/dsPIC33FJ32MC202> (último acceso 31 de marzo de 2020).

El resto de los componentes empleados son básicos y de uso común por lo que no supone ningún problema encontrar la información pertinente a sus funcionalidades y posibles usos.

G-7210
FADISEL S.L.

©© Los nombres registrados y marcas que se citan son propiedad de sus respectivos titulares.

EMISOR Y RECEPTOR de ULTRASONIDOS.

Esta pareja de transductores se complementan para su aplicación en cualquier tipo de aplicaciones de ultrasonidos, tales como: telemandos, alarmas, detectores de movimiento, cuenta-plezas, detectores de nivel, sensores de proximidad, medidores de distancias, sensores de líquidos, puertas automáticas, etc.

CARACTERÍSTICAS Y DIMENSIONES.

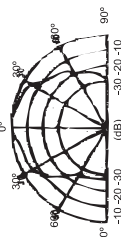
El elemento transmisor es de cerámica y se halla ubicado en una cápsula abierta. Convierte la energía eléctrica en energía acústica (mecánica).

El elemento receptor también es de cerámica y está montado en una cápsula de estructura abierta. Convierte la energía acústica en eléctrica. Ambos están diseñados para trabajar apareados.

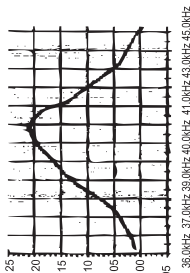
Como complemento ideal, recomendamos la utilización de **nuestro módulo híbrido miniaturizado C-0508**. Se trata de un potente módulo híbrido Thick-Film, que incluye el generador de 40kHz para el emisor de ultrasonidos y el circuito de reconocimiento de amplitud de modulación para el receptor. En la hoja de instrucciones de este componente se incluye un ejemplo de esquema de conexión.

	EMISOR	RECEPTOR
Construcción	estructura abierta	estructura abierta
Frecuencia nominal	40 ± 1kHz	40 ± 1kHz
Sensibilidad		-64dBmin.(0db = 1V/ μ bar)
SPL (Sound Pressure Level)	115dBmin.(0db = 0.02mPa)	
Dirección	80°	80°
Capacidad		2500 ± 25% pF
Temperatura de uso		-40°C a 85°C
Mínimo detectable		0.2m
Máximo detectable		4m
Resolución		9mm
Máxima tensión de entrada	20Vpp señal continua.	

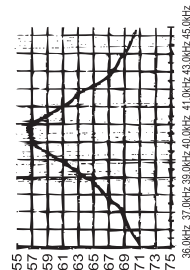
Direccionalidad emisor:



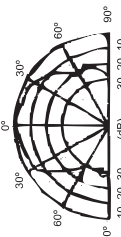
SPL (Sound Pressure Level) (dB)



Sensibilidad (dB)



Direccionalidad receptor:



CONSIDERACIONES.

Este componente está destinado para su uso por parte de profesionales, o usuarios con un nivel técnico o conocimientos suficientes, que les permita desarrollar por sí mismos los proyectos o aplicaciones deseados. Por este motivo no se facilitará asistencia técnica sobre problemas de implementación del citado componente en las aplicaciones en las que sea empleado.

Para cualquier problema relativo al funcionamiento del producto (excluidos los problemas de aplicación), póngase en contacto con nuestro departamento técnico. Fax 93 432 29 95; Correo electrónico: sat@fadisel.com.

La documentación técnica de este producto responde a una transcripción de la proporcionada por el fabricante.

Los productos de la familia 'Componentes' de Cebek disponen de 1 año de garantía a partir de la fecha de compra. Quedan excluidos el trato o manipulación incorrectos.

Disponemos de más productos que pueden interesarle, visítenos en: www.fadisel.com ó solicite nuestro catálogo.

Vibration Motor, Vibrator Motor, Micro Motor, Coin & Pancake Motors , Pager Motors

2-4	Motor position	All positions	
2-5	Operating voltage	2.2 ~ 3.6V DC	
2-6	Operating conditions	-30 ~ 70°C, ordinary humidity Humidity :65±20%RH	No condensation of moisture.
2-7	Storage conditions	-40 ~ 80°C, ordinary humidity Humidity :65±20%RH	No condensation of moisture.

3. Measuring conditions

Items	Specifications	Condition & Remarks
3-1	Temperature	20±2°C
3-2	Humidity	(63 ~ 67%) RH
3-3	Motor position	Motor Shaft horizontal Lock the motor in a test fixture.

3-4 All data are based on the measurement under the temperature of 20 °C and humidity 65 %RH. However, the ranges of temperature 5~35 °C and humidity 45~85 %RH are to be applicable as long as no problems.

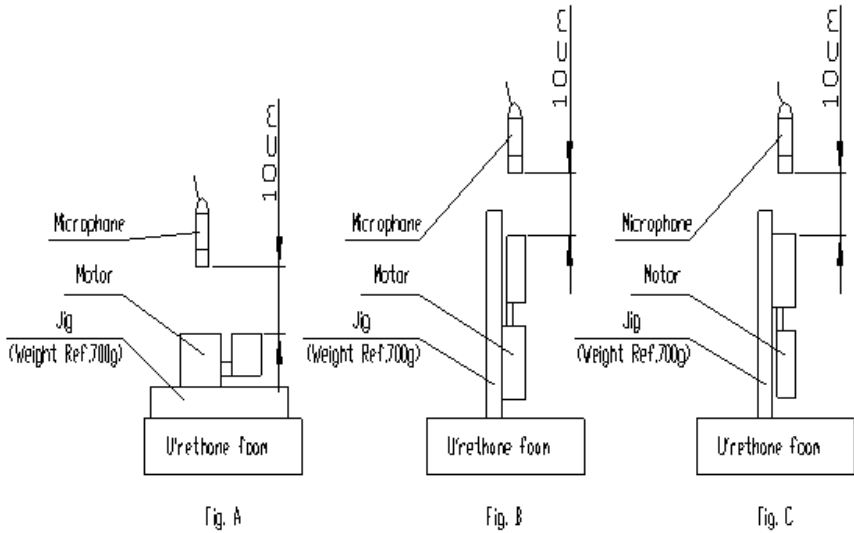
4. Mechanical specifications

Items	Specifications	Condition & Remarks
4-1	Configuration	As specified in 10. Outline drawing Outline drawing No: Z7AL2B1692082.
4-2	Appearance	There shall be no evidence of mechanical damage and shall not have inadequate corrosion and so on Visual examination (allowable extent is based on boundary sample)
4-3	Shaft end play	0.1 ~ 0.3mm
4-4	Holding strength of vibration weight	49N (5kgf) min.

5. Performance and characteristics

<http://www.vibratormotor.com/vib/Z7AL2B1692082.html> (2 of 6) [5/18/2009 4:59:21 PM]

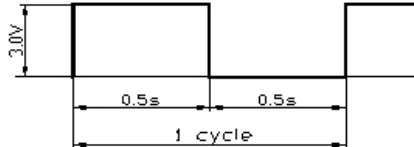
Vibration Motor, Vibrator Motor, Micro Motor, Coin & Pancake Motors , Pager Motors

Items		Specifications	Condition & Remarks
5-1	Rated speed	12,000±2,500rpm	At rated voltage and rated load (Counterweight).
5-2	Rated current	250mA max	
5-3	Stall current	680mA max	At rated voltage.
5-4	Starting voltage	1.7V DC max	(Counterweight) any position of rotor.
5-5	Insulation resistance	1M• min	At DC 100V between terminal and case.
5-6	Terminal resistance	5.5• approx. (±20%)	At 20°C.
5-7	Mechanical noise	50db (A) max	Measured at rated voltage and rated load (counterweight). Background noise: 28db (A) max. @ 10cm. Measuring instruments: B & K. The weight of jig: 700g.  Fig. A Fig. B Fig. C

Vibration Motor, Vibrator Motor, Micro Motor, Coin & Pancake Motors , Pager Motors

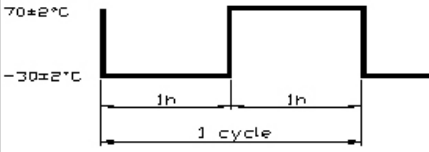
--	--

6. Reliability Test

Items		Standard test conditions	Condition & Remarks												
6-1	Life test	 <table><tr><th>Position</th><th>Voltage</th><th>Load</th><th>Temperature</th><th>Humidity</th><th>Life</th></tr><tr><td>Horizontal</td><td>Rated</td><td>weight</td><td>20 °C</td><td>65 %</td><td>200,000 cycles</td></tr></table>	Position	Voltage	Load	Temperature	Humidity	Life	Horizontal	Rated	weight	20 °C	65 %	200,000 cycles	After 2 hours exposure in ordinary temperature and humidity, Motors shall be approved as specified in item 7-1.
Position	Voltage	Load	Temperature	Humidity	Life										
Horizontal	Rated	weight	20 °C	65 %	200,000 cycles										
6-2	Low Temp exposure	Temperature: -40±2°C Time: 96hrs	After 2 hours exposure in ordinary temperature and humidity, Motors shall be approved as specified in item 7-2.												
6-3	High Temp exposure	Temperature: 60±2°C Time: 96hrs													

<http://www.vibratormotor.com/vib/Z7AL2B1692082.html> (4 of 6) [5/18/2009 4:59:21 PM]

Vibration Motor, Vibrator Motor, Micro Motor, Coin & Pancake Motors, Pager Motors

6-4	Humidity exposure	Temperature: $40 \pm 2^\circ\text{C}$ Humidity: $90 \sim 95\% \text{ RH}$ Exposure time: 96hrs No condensation of moisture	
6-5	Vibration	Displacement: 1.5mm (p-p) Frequency: $10 \sim 55\text{Hz}$ Acceleration: 22m/s^2 Period: 20 Mins log sweep ($10 \sim 55 \sim 10\text{Hz}$) Direction: x, y, z Time: Every 2 hours	After the test motors shall be approved as specified in item 7-2.
6-6	Free fall	Test state: Set the motor to the approximately 75 g (include the motor) weight of block drop the motor on the concrete floor. Height: 1.5 m Direction: 6/6faces Number of times: Twice each	After the test motors shall be approved as specified in item 7-2.
6-7	Heat shock	Test cycle: 20 cycles 	After 2 hours exposure in ordinary temperature and humidity, Motors shall be approved as specified in item 7-2.

7. Requirements

Items		Requirements	
7-1	Table A	1) Rated speed: 2) Rated current: 3) Starting voltage:	Initial data -30% Min, Initial data +60% Max $\pm 30\% / \text{Initial data} \pm 30\% \text{ max}$ 2.0 V DC max
7-2	Table B	1) Rated speed: 2) Rated current: 3) Starting voltage:	$\pm 30\% / \text{Initial data} \pm 30\% \text{ max}$ $\pm 30\% / \text{Initial data} \pm 30\% \text{ max}$ 2.0V DC max

8. Matters to be paid attention to when using motor

8-1 Please lay the motors carefully in transportation to avoid any serious damage to the motor body or its electric function because of collision.

8-2 Please use and storage motors according to N0.2 item (Operating Conditions) in specification, or else motor characteristics would be affected.

Vibration Motor, Vibrator Motor, Micro Motor, Coin & Pancake Motors , Pager Motors

8-3 Make arrangement to limit the storage period to 6 months or less. Condensation of atmosphere must be avoided in motor usage or opening the packaging of the motor.

8-4 For proper operation. storage and operating environment must not contain corrosive gases. For example H₂S. SO₂. NO₂. CL₂. etc. In addition storage environment must not have materials that emit corrosive gases especially from silicon. cyanic. formalin and phenol group. In the mechanism or the set. existence of corrosive gases may cause no rotation in motor.

8-5 Please don't stall the shaft for a long time after powering, and not to touch the weight when motor is rotating.

8-6 There should be no sundries (such as grain, fibre, hair, small tape, glue etc.) in the shaft end play.

[Home](#) | [Product Info](#) | [About us](#) | [Factory tour](#) | [ISO Certifications](#) | [RoHS Info](#) | [Sample Request](#) | [Contact us](#) | [Email us](#)

Copyright 2009 © VibratorMotor.com



dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304

16-bit Digital Signal Controllers (up to 32 KB Flash and 2 KB SRAM) with Motor Control and Advanced Analog

Operating Conditions

- 3.0V to 3.6V, -40°C to +150°C, DC to 20 MIPS
- 3.0V to 3.6V, -40°C to +125°C, DC to 40 MIPS

Core: 16-bit dsPIC33F CPU

- Code-efficient (C and Assembly) architecture
- Two 40-bit wide accumulators
- Single-cycle (MAC/MPY) with dual data fetch
- Single-cycle mixed-sign MUL plus hardware divide

Clock Management

- 2% internal oscillator
- Programmable PLLs and oscillator clock sources
- Fail-Safe Clock Monitor (FSCM)
- Independent Watchdog Timer (WDT)
- Fast wake-up and start-up

Power Management

- Low-power management modes (Sleep, Idle, Doze)
- Integrated Power-on Reset and Brown-out Reset
- 1.35 mA/MHz dynamic current (typical)
- 55 µA IPD current (typical)

High-Speed PWM

- Up to four PWM pairs with independent timing
- Dead time for rising and falling edges
- 12.5 ns PWM resolution
- PWM support for:
 - DC/DC, AC/DC, Inverters, PFC, Lighting
 - BLDC, PMSM, ACIM, SRM
- Programmable Fault inputs
- Flexible trigger configurations for ADC conversions

Advanced Analog Features

- ADC module:
 - Configurable as 10-bit, 1.1 Msps with four S&H or 12-bit, 500 ksps with one S&H

Packages

Type	SPDIP	SOIC	SSOP	QFN-S	QFN	TQFP
Pin Count	28	28	28	28	44	44
Contact Lead/Pitch	.100"	1.27	0.65	0.65	0.65	0.80
I/O Pins	21	21	21	21	35	35
Dimensions	1.365x.285x.135"	17.9x7.50x2.05	10.2x5.3x1.75	6x6x0.9	8x8x0.9	10x10x1

Note: All dimensions are in millimeters (mm) unless specified.

- Six analog inputs on 28-pin devices and up to nine analog inputs on 44-pin devices
- Flexible and independent ADC trigger sources

Timers/Output Compare/Input Capture

- Three 16-bit timers/counters. Can pair up two to make one 32-bit.
- Two Output Capture modules configurable as timers/counters
- Four Input Capture modules
- Peripheral Pin Select (PPS) to allow function remap

Communication Interfaces

- One UART module (10 Mbps)
- With support for LIN 2.0 protocols and IrDA®
- One 4-wire SPI module (15 Mbps)
- One I²C™ module (up to 1 Mbaud) with SMBus support
- PPS to allow function remap

Input/Output

- Sink/Source up to 10 mA (pin specific) for standard VOH/VOL, up to 16 mA (pin specific) for non-standard VOH1
- 5V-tolerant pins
- Selectable open drain, pull-ups, and pull-downs
- Up to 5 mA overvoltage clamp current
- External interrupts on all I/O pins

Qualification and Class B Support

- AEC-Q100 REV G (Grade 0 -40°C to +150°C)
- Class B Safety Library, IEC 60730

Debugger Development Support

- In-circuit and in-application programming
- Two program and two complex data breakpoints
- IEEE 1149.2-compatible (JTAG) boundary scan
- Trace and run-time watch

dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304

dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 Product Families

The device names, pin counts, memory sizes and peripheral availability of each device are listed below. The following pages show their pinout diagrams.

TABLE 1: dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 CONTROLLER FAMILIES

Device	Pins	Program Flash Memory (Kbyte)	RAM (Kbyte)	Remappable Peripherals								10-Bit/12-Bit ADC	I ² C™	I/O Pins	Packages
				Remappable Pins	16-bit Timer	Input Capture	Output Compare Standard PWM	Motor Control PWM	Quadrature Encoder Interface	UART	External Interrupts ⁽³⁾	SPI			
dsPIC33FJ32MC202	28	32	2	16	3 ⁽¹⁾	4	2	6ch ⁽²⁾ 2ch ⁽²⁾	1	1	3	1	1ADC, 6 ch	1	21 SPDIP SOIC SSOP QFN-S
dsPIC33FJ32MC204	44	32	2	26	3 ⁽¹⁾	4	2	6ch ⁽²⁾ 2ch ⁽²⁾	1	1	3	1	1ADC, 9 ch	1	35 QFN TQFP
dsPIC33FJ16MC304	44	16	2	26	3 ⁽¹⁾	4	2	6ch ⁽²⁾ 2ch ⁽²⁾	1	1	3	1	1ADC, 9 ch	1	35 QFN TQFP

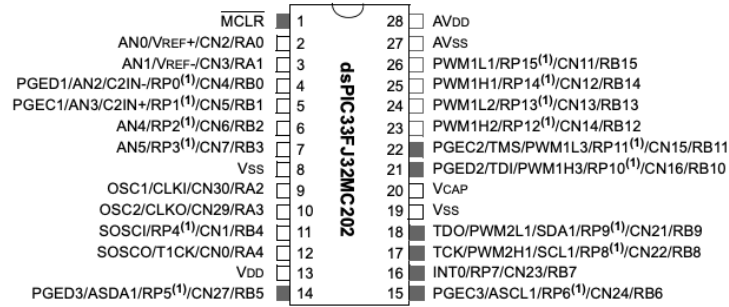
Note 1: Only two out of three timers are remappable.
Note 2: Only PWM fault inputs are remappable.
Note 3: Only two out of three interrupts are remappable.

dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304

Pin Diagrams

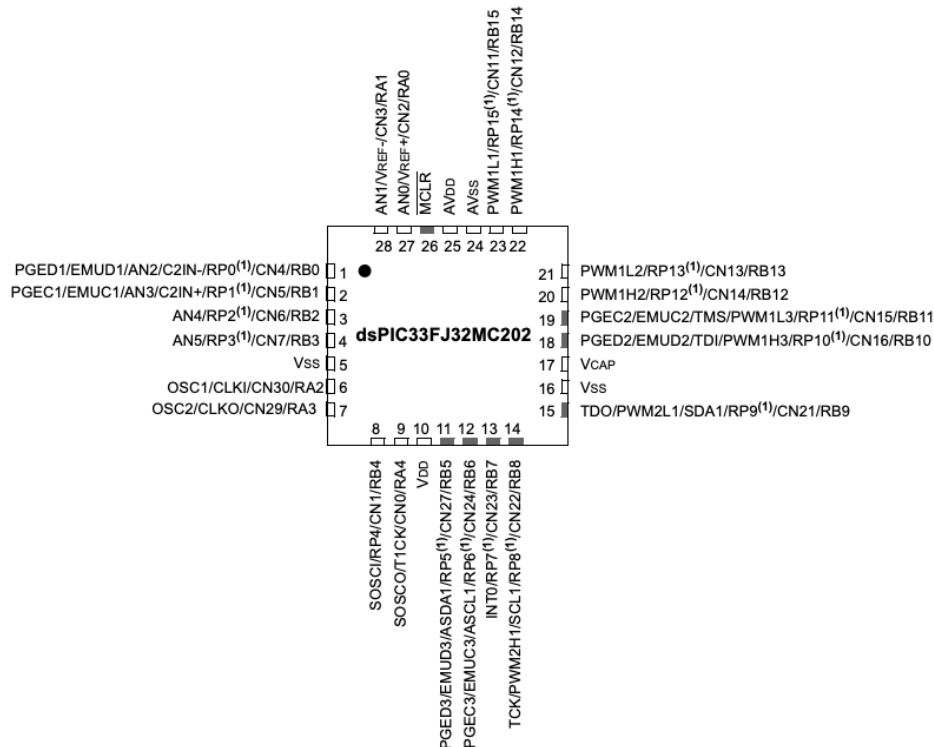
28-PIN SPDIP, SOIC, SSOP

■ = Pins are up to 5V tolerant



28-Pin QFN-S⁽²⁾

■ = Pins are up to 5V tolerant



Note 1: The RPN pins can be used by any remappable peripheral. See [Table 1](#) for the list of available peripherals.

2: The metal plane at the bottom of the device is not connected to any pins and is recommended to be connected to VSS externally.