



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Desarrollo de un sistema embebido de bajo coste para
sensor de herradura

Trabajo Fin de Grado

Grado en Ingeniería Eléctrica

AUTOR/A: Morales Zarco, Cristian

Tutor/a: Rodríguez Ballester, Francisco

CURSO ACADÉMICO: 2021/2022



Escuela Técnica Superior de Ingeniería del Diseño



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DESARROLLO DE UN SISTEMA EMBEBIDO DE BAJO COSTE PARA SENSOR DE HERRADURA

Autor: Cristian Morales Zarco
Tutor: Francisco Rodríguez Ballester

Trabajo Fin de Grado de la carrera Ingeniería Eléctrica 2022

Resumen en castellano

En una instalación de una empresa de almacenaje y distribución se emplean un gran número de transfers (plataformas de transporte) de palés que son controlados mediante un sistema que los localiza y mueve automáticamente.

Estos transfers viajan sobre unos railes; para determinar en cada momento la posición en la que se encuentra el transfer se utilizan una serie de fotocélulas de herradura montadas en el transfer y por las que, durante el movimiento, entran y salen placas fijadas a las diferentes posiciones de trabajo.

Este tipo de sensor con fotocélulas dan muchos problemas en dicha instalación debidos, entre otros motivos, a la suciedad y al gran margen de error que presenta a la hora de determinar las distintas posiciones a lo largo de la zona de trabajo.

Como tema de este TFG se va a desarrollar una maqueta de un nuevo dispositivo de proximidad capaz de sustituir al actualmente instalado, basado en un microcontrolador programado en lenguaje C junto con el uso de tecnología de ultrasonidos, manteniendo el formato de herradura del sensor actual.

Este desarrollo se apoya en y amplía los conocimientos obtenidos en la asignatura de la mención Informática Industrial. Este desarrollo también se realizará de forma que las conexiones y comunicaciones con el PLC que gestiona el movimiento del transfer sean las mismas que con los sensores a sustituir, evitando tener que recurrir a una reprogramación del sistema de control.

Tras el desarrollo se pondrá a prueba el nuevo dispositivo y se comentarán los resultados obtenidos.

Resum en valencià

En una instal·lació d'una empresa de magatzematge i distribució s'empren un gran nombre de transfers (plataformes de transport) de palés que són controlats mitjançant un sistema que els localitza i mou automàticament.

Aquests transfers viatgen sobre uns rails; per a determinar en cada moment la posició en la qual es troba el transfer s'utilitza una sèrie de fotocèl·lules de ferradura muntades en el transfer i per les quals, durant el moviment, entren i ixen plaques fixades a les diferents posicions de treball.

Aquest tipus de sensor amb fotocèl·lules donen molts problemes en aquesta instal·lació deguts, entre altres motius, a la brutícia i al gran marge d'error que presenta a l'hora de determinar les diferents posicions al llarg de la zona de treball.

Com a tema d'aquest TFG es desenvoluparà una maqueta d'un nou dispositiu de proximitat capaç de substituir a l'actualment instal·lat, basat en un microcontrolador programat en llenguatge C juntament amb l'ús de tecnologia d'ultrasons, mantenint el format de ferradura del sensor actual.

Aquest desenvolupament es recolza en i amplia el coneixements obtinguts en l'assignatura de l'esment Informàtic Industrial. Aquest desenvolupament també es realitzarà de manera que les connexions i comunicacions amb el PLC que gestiona el moviment del transfer sigan les mateixes que amb els sensors a substituir, evitant haver que recórrer a una reprogramació del sistema de control.

Després del desenvolupament es posarà a prova el nou dispositiu i es comentaran els resultats obtinguts.

Summary in English

In a facility of a storage and distribution company, many pallet transfers (transport platforms) are used and controlled by a system that automatically locates and moves them.

These transfers travel on rails; to determine the position of the transfer at any given moment, a series of horseshoe photocells mounted on the transfer are used, through which, during the movement, plates attached to the different working positions enter and exit.

These type of sensor with photocells causes many problems in this installation due, among other reasons, to the dirt and the large margin of error it has when determining the different positions along the working area.

The theme of this TFG is to develop a model of a new proximity device able to replace the currently installed one, based on a microcontroller programmed in C language along with the use of ultrasound technology, maintaining the horseshoe format of the current sensor.

This development builds on and expands the knowledge gained in the Industrial Informatics subject. This development will also be carried out in such a way that the connections and communications with the PLC that manages the movement of the transfer are the same as with the sensors to be replaced, avoiding the need for a reprogramming of the control system.

After development, the new device will be tested, and the results will be discussed.

Índice

| | | |
|--------|---|----|
| 1. | Introducción..... | 6 |
| 2. | Sistema embebido..... | 7 |
| 3. | Fotocélula de herradura | 8 |
| 4. | Alternativa de sensor: Ultrasonidos..... | 10 |
| 4.1. | Propuesta de solución | 12 |
| 4.1.1. | Objetivo..... | 13 |
| 4.1.2. | Introducción al programa..... | 15 |
| 5. | Alternativa de sensor: Parte hardware | 23 |
| 5.1. | Tarjeta microcontrolador MSP432 | 23 |
| 5.2. | Sensor US100 | 24 |
| 6. | Alternativa de sensor: Parte software..... | 24 |
| 6.1. | Fichero.c..... | 26 |
| 6.1.1. | Macros privadas..... | 26 |
| 6.1.2. | Tipos de datos privados | 26 |
| 6.1.3. | Definición de variables privadas..... | 26 |
| 6.1.4. | Declaración de funciones privadas | 27 |
| 6.1.5. | Implementación de funciones privadas | 27 |
| 6.1.6. | Implementación de funciones públicas | 27 |
| 6.2. | Fichero.h | 27 |
| 6.3. | Esquema de las funciones..... | 28 |
| 6.4. | Módulo Timer_A..... | 31 |
| 6.4.1. | Macros privadas..... | 31 |
| 6.4.2. | Tipos de datos privados | 31 |
| 6.4.3. | Definición de variables privadas..... | 32 |
| 6.4.4. | Declaración de funciones privadas | 33 |
| 6.4.5. | Implementación de funciones privadas | 33 |
| 6.5. | Módulo Ultrasonidos | 39 |
| 6.5.1. | Macros privadas..... | 39 |
| 6.5.2. | Tipos de datos privados | 39 |
| 6.5.3. | Definición de variables privadas..... | 40 |
| 6.5.4. | Implementación de funciones públicas | 40 |
| 6.5.5. | Declaración de funciones públicas..... | 41 |
| 6.6. | Módulo LEDs | 41 |

| | | |
|--------|--|----|
| 6.6.1. | Macros privadas..... | 42 |
| 6.6.2. | Tipos de datos privados | 42 |
| 6.6.3. | Definición de variables privadas..... | 42 |
| 6.6.4. | Declaración de funciones privadas | 42 |
| 6.6.5. | Implementación de funciones privadas [x] | 42 |
| 6.6.6. | Implementación de funciones públicas | 43 |
| 6.6.7. | Declaración de funciones públicas..... | 43 |
| 6.7. | Fichero principal “main.c” | 43 |
| 6.7.1. | Código del programa..... | 44 |
| 7. | Tests y resultados..... | 44 |
| 8. | Conclusiones | 47 |
| 9. | Presupuesto | 48 |
| 9.1. | Costes y precios | 48 |
| 10. | Pliego de condiciones | 51 |
| 10.1. | Producto..... | 51 |
| 10.2. | Materiales | 51 |
| 10.3. | Procesos | 55 |
| 10.4. | Acabados..... | 56 |
| 11. | Planos y esquemas | 57 |
| 12. | Bibliografía | 65 |

1. Introducción

La idea para mi elección del Trabajo de Fin de Grado ha sido a través de mi paso en las prácticas de empresa. Las he llevado a cabo en uno de los almacenes de la cooperativa de Consum, donde acompañaba a un grupo de electromecánicos encargados del mantenimiento de toda la maquinaria encargada del transporte de palets por la instalación.

Tras un par de semanas allí viendo y entendiendo el trabajo de la maquinaria, busqué posibilidades de TFG, a poder ser que tuviera relación con la rama de automática en la que estoy matriculado. Con ayuda a sugerencias del tutor, escogí como tema del trabajo alternativas a las fotocélulas de herradura que indican la posición de los transportadores (o transfers) de palets (en la Imagen 1 se muestra una fotografía de uno de los transfers de la empresa), ya que estos tenían varios problemas. Además, busco que el cambio del sensor no aumente en exceso el coste referido a la instalación de estos.

Para ello se instalará el sensor de ultrasonidos a una herradura, similar a la instalada, con la intención de que tenga las mismas conexiones de entrada y salida, como si se tratara de una fotocélula.

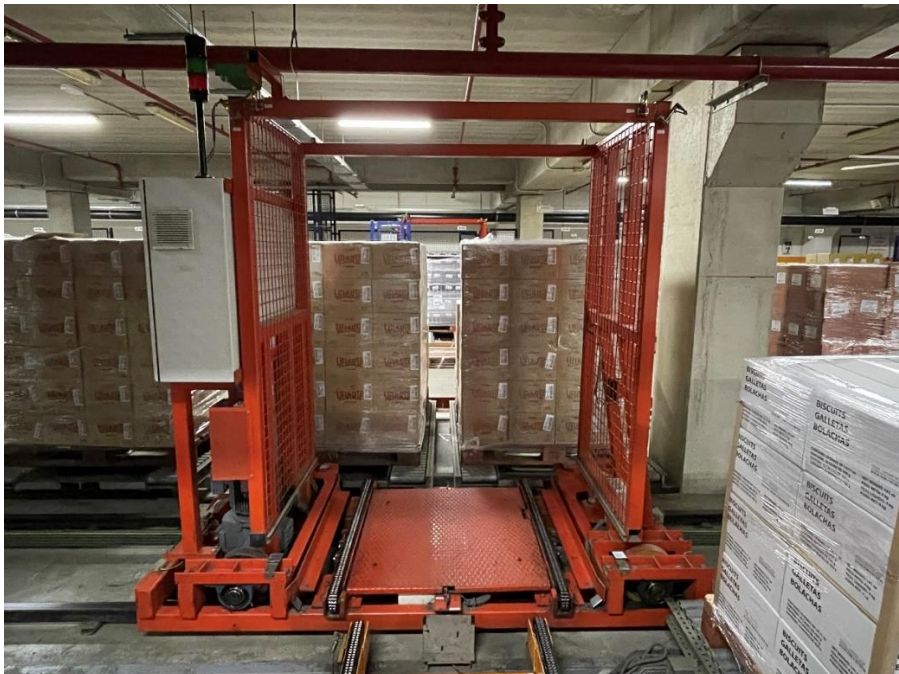


Imagen 1. Transfer para palets (producción propia)

2. Sistema embebido

El sistema embebido es un sistema electrónico preparado para realizar una tarea concreta. La mayoría de los componentes que lo componen están instalados en la placa base. Además, permite realizar la programación con distintos lenguajes, dependiendo de la aplicación, tales como el lenguaje ensamblador del microcontrolador incorporado al sistema, lenguaje C o C++, o Java. ([1] Pág. 3-4)

Para este trabajo, el sistema embebido consiste en un microcontrolador que se encargará de comunicarse con el sensor escogido para sustituir a la fotocélula. Este microcontrolador tendrá como objetivo hacer que el cableado con el que se conecta una fotocélula de herradura al sistema (tensión, neutro y comunicación), sea el mismo que para un sensor de ultrasonidos de herradura.

En este caso, el microcontrolador que se utilizará es el MSP432P401R (Imagen 2), del fabricante Texas Instruments, para realizar la programación y los ensayos con la alternativa de sensor. La tarjeta donde está instalado el microcontrolador contiene: el propio microcontrolador MSP432P401R, 40 pines de conexión para trabajar, dos botones y dos leds con los que interactuar, un depurador de errores XDS110-ET, y un UART controlador de comunicación serie. ([2] Pág. 3)



Imagen 2. Tarjeta MSP-EXP432P401R (tomada de <https://export.rsdelivers.com/es/product/texas-instruments/msp-exp432p401r/msp-exp432p401r-micro-amp-up-dev-kits/1977387>).

Debido a que esta tarjeta de la que disponemos es de un tamaño que dificultaría el objetivo de que la herradura final sea similar a la anterior, se utilizará una tarjeta más reducida (Imagen 3), pero con el mismo microcontrolador. La tarjeta con la que trabajaremos se utilizará para escribir el código del programa y realizar las pruebas con el sensor.

Este programa permitirá que sea fácil adaptar sus contenidos referidos al control de: los leds de la tarjeta y los temporizadores y pines del microcontrolador. Dependiendo de la tarjeta que se utilice, la distribución de los pines puede variar de una a la otra tal y como se puede observar en qué pines son accesibles de cada una, además de los leds que son alimentados desde pines distintos.



Imagen 3. Tarjeta Mini MSP432. (tomada de <https://www.mouser.es/ProductDetail/Mikroe/MIKROE-2682?qs=HXFqYaX1Q2y9y4y0walAMw%3D%3D>)

3. Fococélula de herradura

El sensor fotoeléctrico de tipo de haz de barrera (en la Imagen 4 se muestra el sensor de fotocélula que hay instalado en la empresa) se utiliza para medir el cambio de la cantidad de luz captada por el dispositivo receptor causado por un objeto obstaculizando la trayectoria del haz de luz. Esta obstrucción ocurre cuando el objeto se interpone entre el emisor y el receptor del haz. En este caso, ambos se encuentran en los dos extremos de una herradura, de manera que cuando un objeto pasa entre medias de estos, activa el sensor. [3]



Imagen 4. Fotocélula de herradura (producción propia).

La aplicación que se le da en la instalación donde he hecho las prácticas de empresa es el siguiente. Hay un total de cinco fotocélulas de herradura posicionadas paralelamente entre ellas tal que, mediante una combinación de activación y desactivación de ciertas fotocélulas utilizando unas placas de metal (véase bien en la Imagen 5), el transfer se sitúa en el pasillo para realizar las cargas y descargas. Como aclaración del funcionamiento del sistema haré una breve explicación para cada herradura:

- Herradura 1: cada puesto en el que se va a descargar un palet, tiene una placa de metal que indica al transfer que se encuentra en una posición de descarga.
- Herradura 2: solamente lee una placa que se encuentra en el extremo inicial del recorrido del transfer por los raíles.
- Herradura 3: solamente lee una placa que se encuentra en el recorrido final del recorrido del transfer por los raíles.
- Herradura 4: es una sola placa de más longitud que las que detectan las primeras tres herraduras. Indica al transfer que se encuentra próximo al extremo inicial.
- Herradura 5: es una sola placa de más longitud que las utilizadas en el resto de las herraduras. Indica al transfer que se encuentra próximo al extremo final.

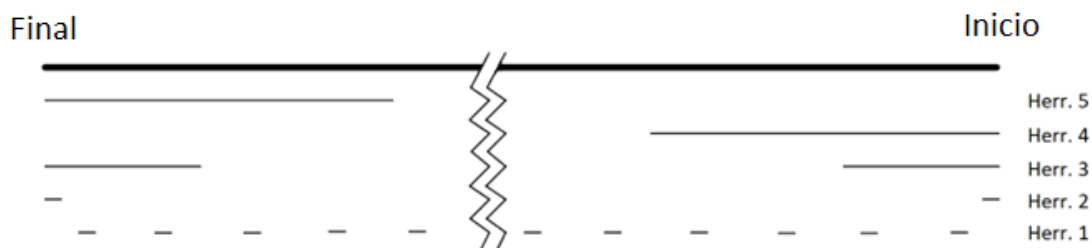


Imagen 5. Placas de lectura de las fotocélulas de herradura (producción propia).

Estas herraduras están conectadas a uno de los módulos de entrada que tiene cada transfer. Este módulo en concreto tiene la función de recopilar los datos de todos los sensores de proximidad que tiene la máquina en cuestión, no solo los de posición. Finalmente, la señal que recibe el módulo es transmitida al PLC donde se hace el control del transfer. [4]

Cada herradura se acopla con tres cables (alimentación, neutro y comunicación) y tienen un concreto modo de funcionamiento para la detección de obstáculos. Por ello, hay que programar los microcontroladores que se conectarán al sensor de ultrasonidos, que se explicará más adelante, para que mantengan el mismo funcionamiento que las fotocélulas de herradura.

Para terminar, se comentarán las características principales del sensor de barrera fotoeléctrica. Por un lado, el haz de luz viaja en línea recta, siendo así fácil de reconocer su área de detección. La gran precisión que dispone este sensor, además del reducido tamaño del haz de luz, permite una distancia de detección más elevada en comparación a otros sensores (aunque en este caso, donde se encuentra instalada en una herradura, no es una característica significativa). [3]

Pese a estas características respecto al sensor fotoeléctrico, en la instalación se trabaja en un ambiente cargado de polvo y restos sólidos provocado por la rotura de los palets de madera utilizados en el transporte del producto. Esto provoca repetidas interrupciones en el funcionamiento de la maquinaria.

4. Alternativa de sensor: Ultrasonidos

El funcionamiento de este tipo de sensores se basa en que el dispositivo emisor de una señal ultrasónica envía un impulso que rebota en el primer obstáculo que encuentra, provocando un eco de la señal que es recibido por el dispositivo receptor y se transforma en una señal eléctrica (véase bien en la Imagen 6). La detección del sensor se produce midiendo el tiempo transcurrido desde que el emisor envía la señal hasta que el receptor capta el eco de la señal, calculando la distancia entre el dispositivo y el obstáculo, siempre que la distancia se encuentre en el rango de detección del sensor. Este rango puede ir desde unos pocos centímetros a unas pocas decenas de metros, dependiendo de la potencia y sensibilidad del sensor, y siempre que el entorno sea más limpio de ruido a medida que aumenta la distancia. [5]

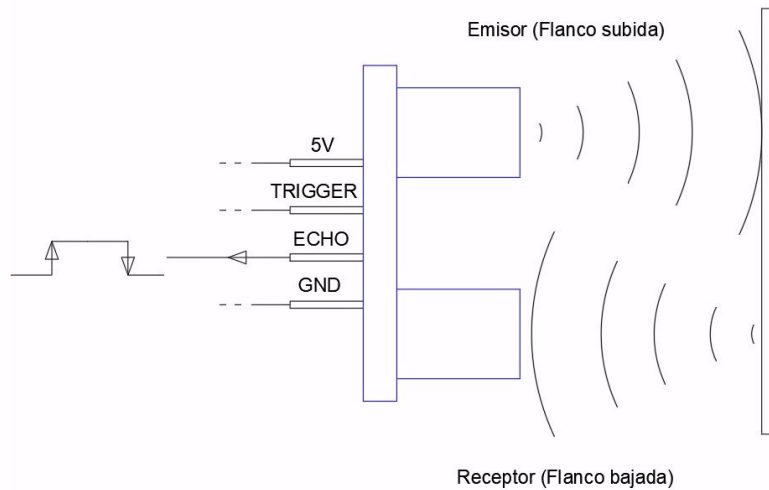


Imagen 6. Movimiento del sonido (producción propia).

En el momento de medir las distancias en base de la velocidad a la que viaja el sonido. Este comportamiento también varía en función de la temperatura a la que se encuentre el aire por el que se transmite, y de la humedad que haya en el ambiente. Debido a estas consideraciones, los valores medidos tendrán mayor grado de irregularidad, resultando en una imprecisión a la hora de medir la velocidad del sonido.

Dado a que el valor exacto de la distancia a la que se encuentre el objeto no nos es relevante, no tomamos consideración de las condiciones atmosféricas a la que se encuentre el aire. Para este trabajo, solamente queremos saber cuándo atraviesa un objeto la herradura, no su distancia.

En cualquier caso, si en algún momento se fuera a utilizar el sensor para calcular la distancia de proximidad lo más preciso posible, para reducir las variaciones producidas por los cambios de temperatura, el sensor de ultrasonidos tiene integrado un sensor de temperatura. Este sensor consiste en un diodo que, según la temperatura ambiente, hará que se regule el valor que devuelve de detección.

Que la detección se realice a través de ultrasonido en lugar de con fotocélula puede tener varias ventajas: [5]

- No habrá problemas en si el objeto a detectar es translúcido, no sea de un color concreto o que sea metálico, ya que el ultrasonido solamente necesita un objeto en el que producir un eco.
- También tienen muy buena resistencia a los entornos industriales.
- No le suponen problemas ambientes sucios o con polvo.

4.1. Propuesta de solución

En este apartado se va a explicar los elementos que se han empleado para el desarrollo tanto del código de programación, como de las conexiones que tendrán los dispositivos que operan en este trabajo. El sensor que se va a utilizar es el US100 (Imagen 7), un sensor de bajo coste debido a su diseño sencillo, baja potencia y que trabajará en un entorno poco desfavorable.

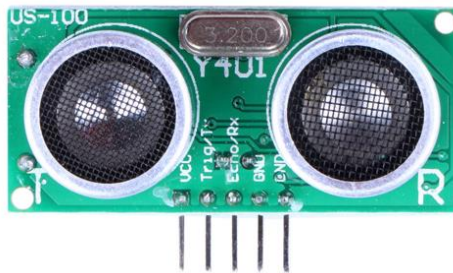


Imagen 7. Sensor US100 (tomada de <https://www.microjpm.com/products/ad38318/>).

Se va a diseñar una herradura que permita la instalación de hasta dos sensores, dando la opción de elegir entre que contenga uno o dos sensores. Gracias a esto, las opciones de compra del producto serán más variado, lo que hace posible elegir la que más convenga para acoplarlas a la instalación.

Un ejemplo de distribución de herraduras sería reducir el número de herraduras a tres (Imagen 9), lo que abarataría el coste del lote de herraduras, y se conseguiría más espacio para instalarlas. Para este nuevo diseño, se ha realizado un ejemplo de la carcasa que se utilizaría y que contendrá tanto los sensores como el microcontrolador (Imagen 8):

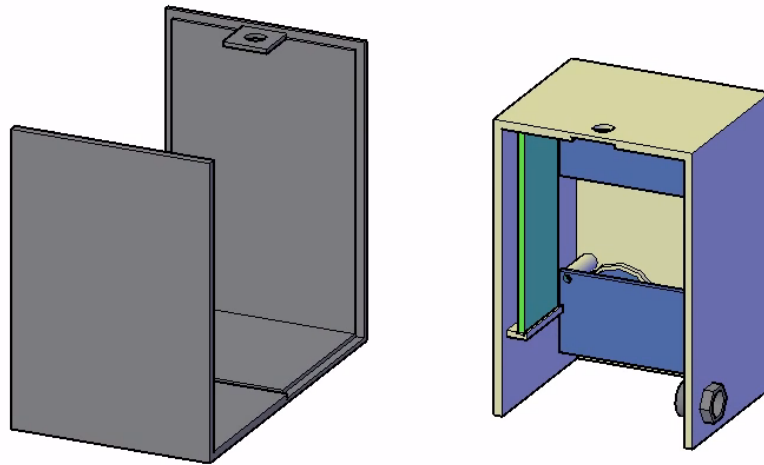


Imagen 8. Diseño AutoCAD 3D (producción propia).

Para que esto sea posible, van a ser necesarias nuevas placas de detección que dispongan de una doble anchura (en la Imagen 9 se observan las placas de metal a punto de atravesar las herraduras instaladas en paralelo), de forma que cada anchura de la misma placa referencie a uno de los carriles de detección de la instalación original.

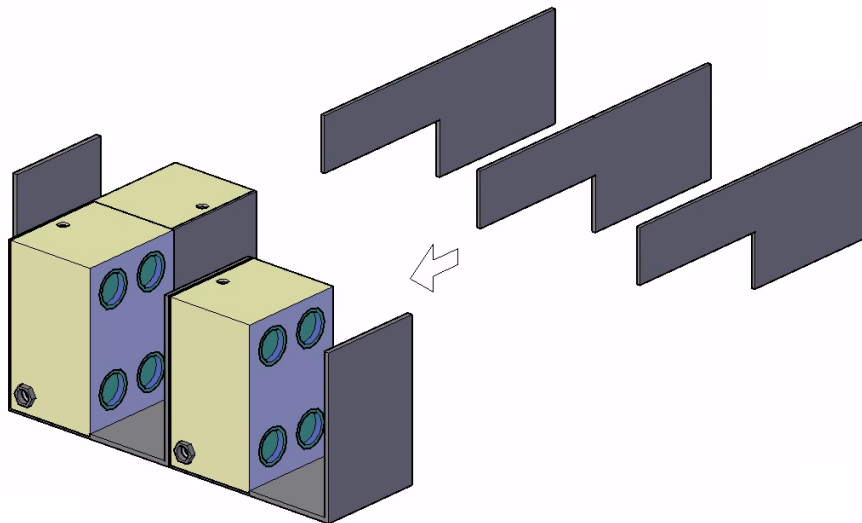


Imagen 9. Ejemplo de distribución de las herraduras (producción propia).

4.1.1. Objetivo

Como ya he comentado anteriormente, el objetivo del programa está en conectar el sensor de ultrasonidos con el módulo del PLC de forma que a sus ojos no haya diferencia a si estuviera conectada con una fotocélula de herradura.

Por una parte, para que el sensor de ultrasonidos haga una lectura, se le enviará una señal trigger para activarlo. Para hacer que la lectura se realice siempre y solo se detenga cuando aparezca un obstáculo, se le enviarán pulsos de disparo al sensor de, al menos, 10 microsegundos cada cierto tiempo, recogiendo el pulso en la señal de eco y realizándose la medida. ([6] Pág. 18)

La frecuencia con la que enviar este pulso dependerá de la anchura de las herraduras. Todas estas son atravesadas por una sola placa de detección, por ello la anchura de cada una son unos 40 milímetros. Diez milímetros más de anchura respecto a la fotocélula ya instalada, y se debe a que este sensor tiene un rango mínimo de detección de 20 milímetros. ([6] Pág. 9)

Esta distancia indica el tiempo que tardará en viajar la señal de alta frecuencia desde que se emite hasta que se recibe la señal. Por lo tanto, hay dos tiempos de viaje de la señal: uno en caso de detectar obstáculo, y otro para el caso de no detectar nada. Estos tiempos se calculan utilizando la fórmula ([6] Pág. 8):

$$Distancia\ obstáculo\ [cm] = \frac{Ancho\ de\ pulso\ del\ ECHO\ [\mu s]}{58}$$

Como ya vimos, el comportamiento del sonido varía con la temperatura y humedad. En nuestro caso, consideraremos 21.6°C, y una humedad del 40% del aire. Para estos valores, y según la ficha técnica del sensor ([6] Pág. 8), se utiliza el divisor de “58”. Entonces, si se lo dividimos al tiempo que tarda la señal ultrasonido en golpear contra el objeto y volver, medido en microsegundos, obtendremos la distancia entre el emisor y receptor, y el objeto contra el que colisiona, medido en “cm”.

- Caso con obstáculo, 2 centímetros entre el sensor y el objeto:

$$Ancho\ de\ pulso\ del\ ECHO = 2 * 58 = 116\ \mu s$$

Cuando el tiempo resulta alrededor de este valor, se considera que hay un obstáculo y el microcontrolador enviará una señal de detección al PLC.

- Caso sin obstáculo, 4 centímetros entre el sensor y el objeto:

$$Ancho\ de\ pulso\ del\ ECHO = 4 * 58 = 232\ \mu s$$

En el caso de no haber detección, se debe volver a enviar el pulso que pone en marcha de nuevo el sensor cada 62 milisegundos. Para ello se necesita utilizar un timer_A en modo UP, utilizando un canal para marcar el periodo para que sea de este valor, y otro canal en modo comparación que generará el pulso de 20 microsegundos como una señal PWM.

Por otra parte, hay que medir el ancho del pulso producido por el canal ECHO (Imagen 10) usando el modo captura del timer_A. Estas capturas consisten en obtener los valores de tiempo del flanco de subida y de bajada del pulso. Por último, tras la captura del flanco de bajada se hará una resta entre la captura realizada en el flanco de bajada menos el de subida, resultando en el ancho de pulso del eco.

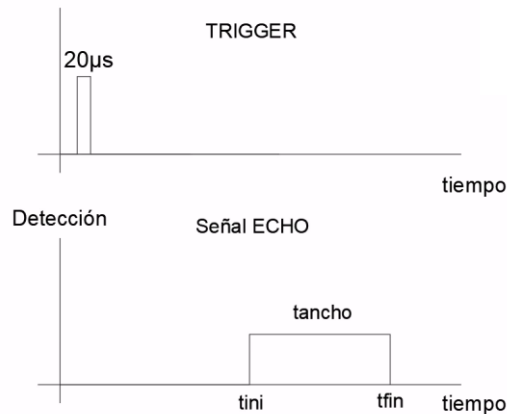


Imagen 10. Señal TRIGGER y ECHO (producción propia).

4.1.2. Introducción al programa

En cada herradura irá instalado un solo microcontrolador que controle los sensores. Por ello, los datos que necesita el programa serán distintos dependiendo de si en la herradura se instalarán uno o dos sensores.

Se va a evitar hacer uso de un solo fichero en donde se escriba toda la programación al completo ya que esto provoca que el programa sea: muy difícil de aplicar fuera del trabajo, complicado de redactar, y confuso a la hora de hacer las pruebas. Además, el programa seguirá requiriendo que se le adapte dependiendo de la tarjeta que se utilice, como por ejemplo en el caso de este trabajo, donde el desarrollo se hará utilizando tanto la tarjeta MSP432 (Imagen 2) como la mini MSP432 (Imagen 3).

4.1.2.1. Módulos

Para reducir al máximo posible los cambios que haya que hacer de un programa a otro, y así poder reutilizar el mayor número posible de líneas de código, se va a desarrollar un programa

que pueda usarse en otros proyectos. La forma correcta de hacer el programa sería entonces empleando módulos.

Los módulos consisten en la agrupación de los procesos que cubren la misma área de trabajo. Esto hace que el programa resulte en ser más sencillo de escribir y fácil de seguir. Cada módulo escrito en lenguaje C estará compuesto por dos ficheros, uno “.c” y otro “.h”. En el “fichero.h” se escribirán las declaraciones de las funciones que ponen en marcha el módulo (funciones públicas). Mientras que en el “fichero.c” se desarrollarán todas las funciones que lo hacen funcionar.

En este caso, se va a distinguir entre el módulo “Timer_A”, “Ultrasonidos” y “LEDs”, de forma que con simplemente cambiar los datos de la configuración de cada uno sea posible utilizarlos en posibles trabajos futuros (Imagen 11).

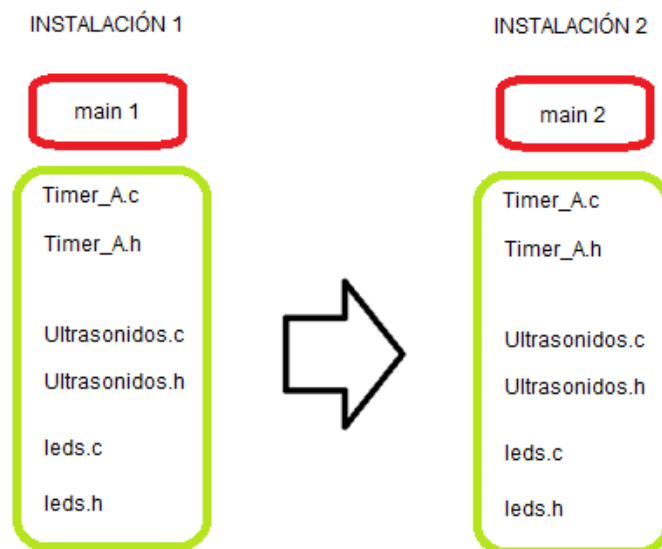


Imagen 11. Estructura modular (producción propia).

Por lo tanto, los ficheros que compondrán la aplicación son: los utilizados para controlar los temporizadores timer_A que hay integrados en el microcontrolador, “Timer_A.c” y “Timer_A.h”; los referidos al sensor de ultrasonidos que se va a utilizar, el sensor US100, “Ultrasonidos.c” y “Ultrasonidos.h”; los utilizados para controlar los leds de la tarjeta, “leds.c” y “leds.h”; y, por último, tendremos también el archivo principal “main”, que será el que envíe la información de las señales captadas por el sensor al módulo que está conectado al PLC.

4.1.2.2. Datos

Dentro del programa, este va a ser el único apartado que se va a necesitar actualizar de cara a cuando se vaya a cambiar de tarjeta o de instalación.

Aquí se incluirá la información referente a los componentes que se vayan a utilizar y por tanto necesiten una configuración en el trabajo. Los componentes en cuestión serán: los temporizadores timerA del microcontrolador, los pines por donde circularán las señales y se harán las conexiones, y los sensores ultrasonidos.

La forma en la que se rellenan estos datos es simple. Primero, creas una estructura donde se guionizan los parámetros que vas a asignar a cada tipo de componente según necesites para el proyecto. Ya entonces, solo queda definir los valores para cada uno, donde se indicarán sus características y cuántos de ellos (temporizadores, pines o sensores) se van a utilizar. Estas características se explicarán en el apartado de “Software”.

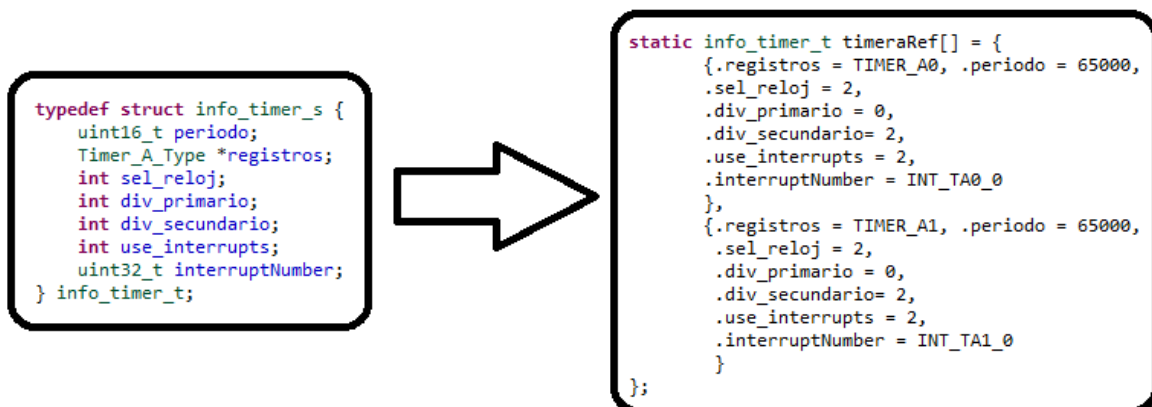


Imagen 12. Datos temporizadores Timer_A (producción propia).

Como ejemplo, según podemos observar en la Imagen 12, en el recuadro de la izquierda se indican qué características van a tener los temporizadores, en este caso. Mientras que en el recuadro de la derecha se rellena la información como interese. En este segundo podemos observar que está dividido en dos llaves, donde la primera contiene la información referente al primer temporizador “TIMER_A0”, y la siguiente la del segundo “TIMER_A1”.

Utilizando este método de relleno de datos, es muy fácil e intuitivo cambiar la información, o incluso quitar o añadir temporizadores según convenga. Además, esta idea es la misma utilizada para el resto de los componentes.

4.1.2.3. Pines y conexiones

El microcontrolador sobre el que haremos el desarrollo y la depuración de errores es el MSP432P401R. Este micro tiene hasta un total de 4 temporizadores timerA, donde cada uno tiene 4 canales de comparación y captura, y un quinto canal para el control de modo del temporizador. Cada canal de cada temporizador tiene un pin con disponible acceso.

De estos temporizadores solamente utilizaremos uno, el “TA2” accesible desde los pines “P5.6, P5.7, P6.6, P6.7” (Imagen 13):

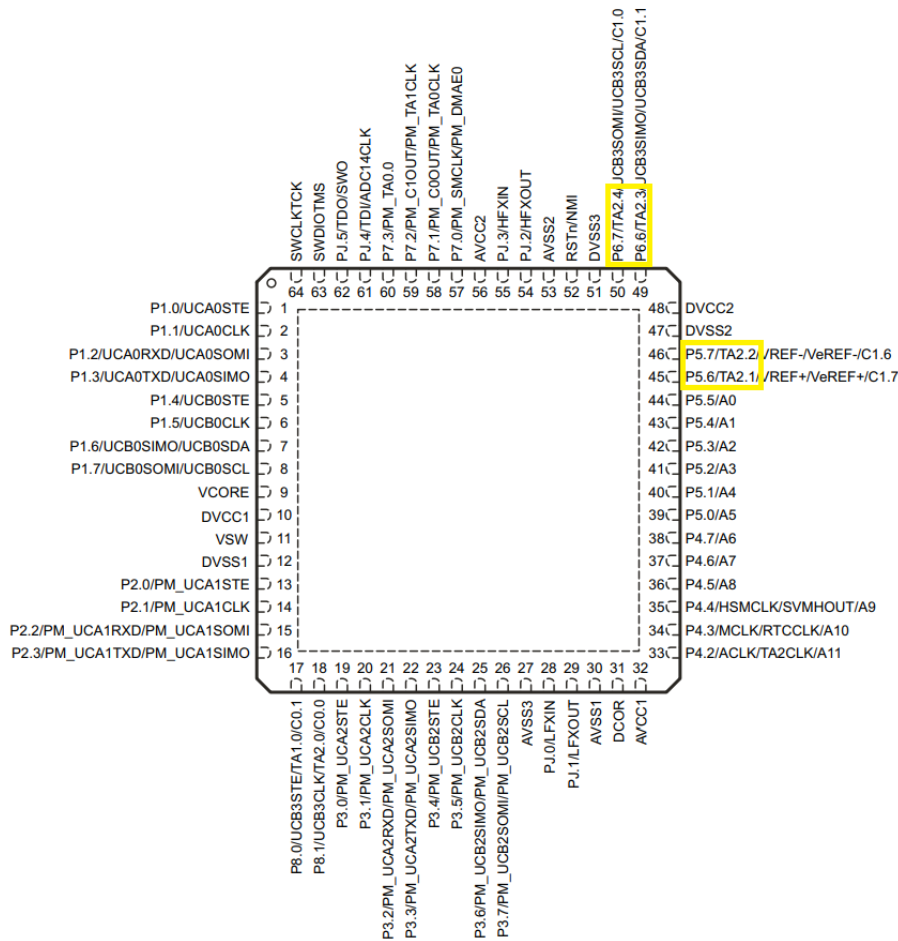


Imagen 13. Pines microcontrolador MSP432 (tomada de <https://datasheet.octopart.com/MSP-EXP432P401R-Texas-Instruments-datasheet-66100013.pdf>, Pág. 9).

Para hacer uso de los temporizadores que tiene el micro, se debe acudir a los pines que tengan una conexión con estos. Esta información se puede obtener de su ficha técnica adjunta observando qué pines son accesibles ([2] Pág. 36, Figura 28). El funcionamiento sobre esto se puede resumir en que cada pin puede tener hasta 4 nombres, el que actúa como GPIO, y después el primario, secundario y terciario. Estos tres últimos son los que indican sus conexiones con los distintos componentes del microcontrolador (temporizadores, relojes).

Microcontrolador MSP432

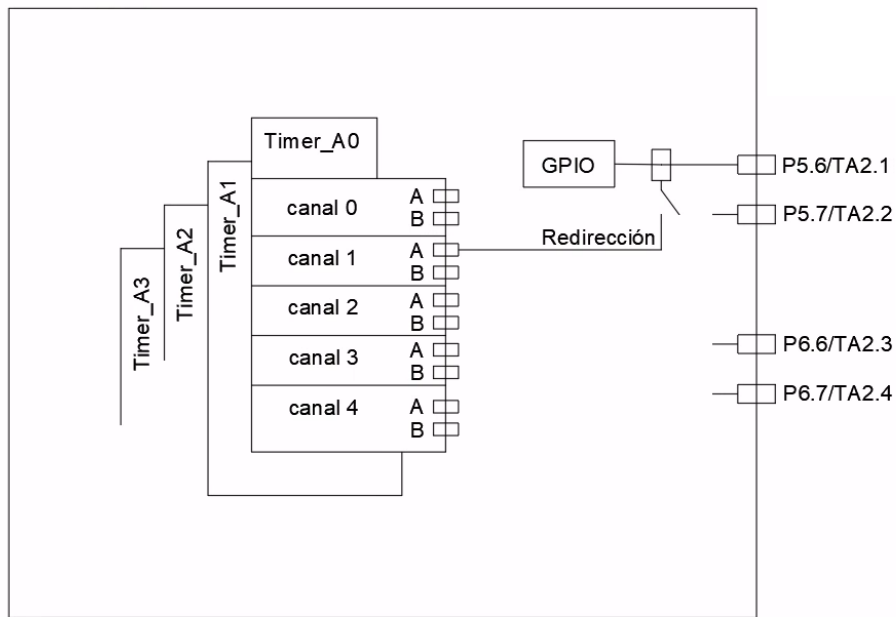


Imagen 14. Esquema pines del microcontrolador (producción propia).

Tal y como se observa en la Imagen 14, cada pin tiene su modo de funcionamiento GPIO. Algunos de estos también pueden estar conectados a otros componentes del micro, y que vienen indicados en si disponen de nombre primario, secundario y terciario. En ellos se pueden leer de qué otra función puede hacer uso el pin.

Por ejemplo, con el ejemplo del esquema, podemos observar el nombre “P5.6/TA2.1”. Este pin se compone del nombre con función GPIO “P5.6”, y su nombre primario “TA2.1”, el que indica una conexión al canal 1 del temporizador 2.

Para intercambiar el modo entre GPIO y los otros, se hace uso de los registros de selección de modo (“SEL0” y “SEL1”) ([7] Pág. 679).

Los canales de comparación y captura los utilizaremos para:

- En caso de usar el modo comparación, este consiste en emitir una señal PWM con un pulso de periodo establecido en los datos de los canales y pines. Esta señal se conecta al TRIGGER del sensor para iniciar una medida.
- En el otro caso, al usar el modo captura, este se conecta a la señal ECHO de respuesta del sensor y se encarga de percibir los flancos de subida o de bajada, guardando el momento de tiempo en el que ocurre cada uno.

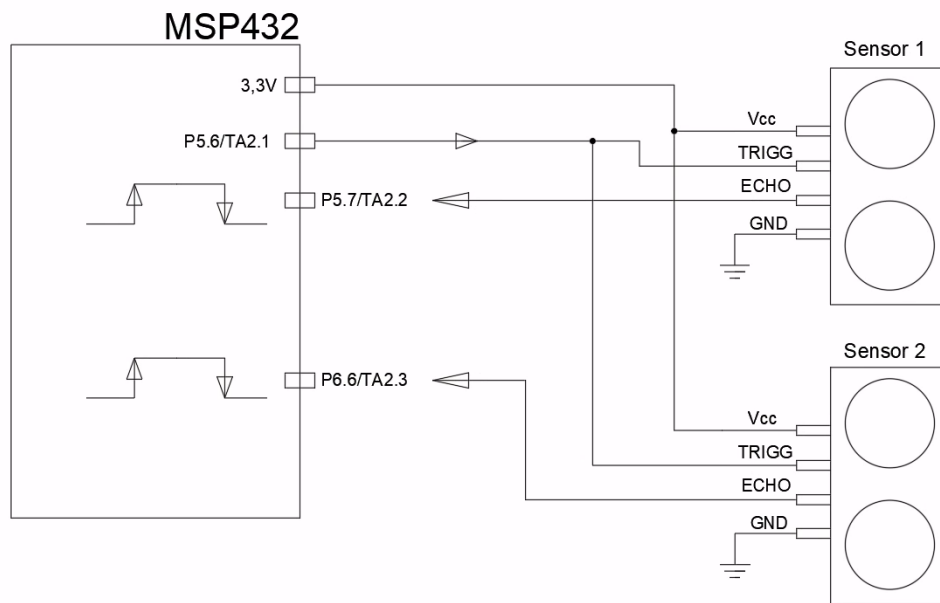


Imagen 15. Esquema conexión sensor y micro (producción propia).

En la Imagen 15 se observan las conexiones que se deberán hacer entre los sensores y el microcontrolador para que el sistema funcione correctamente.

Aparte de que ambos sensores se alimentan a “3,3V”, y que necesitan su conexión a tierra para protegerlos, observamos el uso de los canales en modo de comparación y captura.

El canal que se estará utilizando en modo de comparación será el que produce el pin “P5.6”. Este emitirá un pulso PWM de, al menos, 10 microsegundos que pondrá en marcha los sensores, siendo la señal de trigger de los detectores de proximidad. También vemos que es posible utilizar una misma señal PWM para poner en marcha varios sensores.

En cuanto a las señales ECHO que devuelven los dispositivos, cada una es recibida como una entrada para el microcontrolador. Se utiliza un canal para cada señal ECHO. Este será capaz de distinguir, si así lo hemos especificado en los parámetros del pin, entre flanco de subida y de bajada del pulso ECHO.

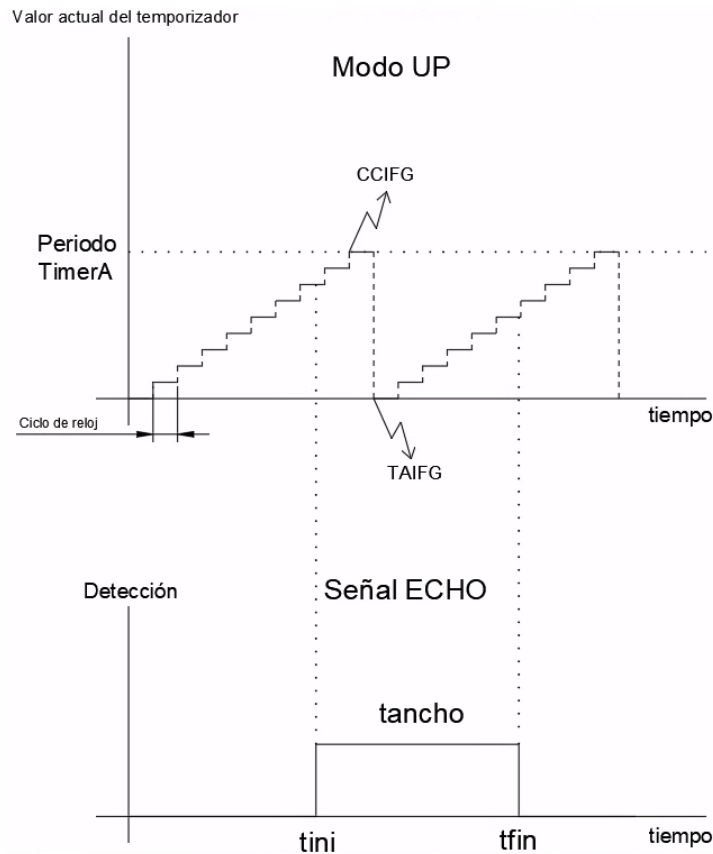


Imagen 16. Señal ECHO (producción propia).

Con la ayuda de la representación de la señal ECHO junto al conteo del temporizador (Imagen 16), haré una explicación de su funcionamiento:

- El periodo con el que el temporizador lleva el conteo del tiempo transcurrido del programa se mide en ciclos de reloj.
- A mayor frecuencia tenga el reloj, más precisas serán las mediciones. En este caso, la frecuencia con la que va a trabajar será de 1MHz, para ello se divide la frecuencia del reloj, 3 MHz, entre un divisor de valor 3.
- Se observa el funcionamiento en modo UP del temporizador y los flags de interrupción que actúan en cada periodo de tiempo:

Por un lado, el modo UP consiste en un conteo entre el valor cero, y el almacenado en la variable de entrada “periodo”.

Por otro lado, para cada vez que pasa un periodo del temporizador, se activan dos flags de interrupción: el flag de comparación y captura (CCIFG), que es la que utilizamos, y el flag del timerA (TAIFG). De esta forma haremos el conteo de cuántos periodos han transcurrido desde el inicio del programa, mediante el conteo de las rampas.

- El ancho del pulso de la señal ECHO que recibe el microcontrolador se calcula restando el momento de detección del receptor (“tfin”) menos el momento de emisión de la señal ultrasonido. Este tiempo resultante abarca el doble de la distancia que hay entre los elementos emisor y receptor, y el obstáculo contra el que rebota (viaje de ida y de vuelta).

4.1.2.4. Interrupciones

Si se activa el flag de interrupción (IFG), se activa la solicitud de interrupción (IRQ) y se pausa el programa principal mientras se ejecuta el manejador de interrupción asociado a la interrupción (nótese en la Imagen 17. Esquema uso de interrupciones (producción propia).). Del mismo modo, cuando se borra el flag, se desactiva esta solicitud y la aplicación vuelve a ponerse en marcha.

Como acabamos de señalar, en el momento que se activa la solicitud de interrupciones y se pausa el programa principal, comienza el manejador de interrupciones (ISR) que pone en marcha la función de respuesta a una interrupción. Dependiendo de qué flag de interrupciones se active, podemos ejecutar las funciones que deseemos utilizar.

El uso de interrupciones las utilizaremos en este trabajo para las mediciones de los sensores. Una forma de representarlo de forma simplificada sería la siguiente:

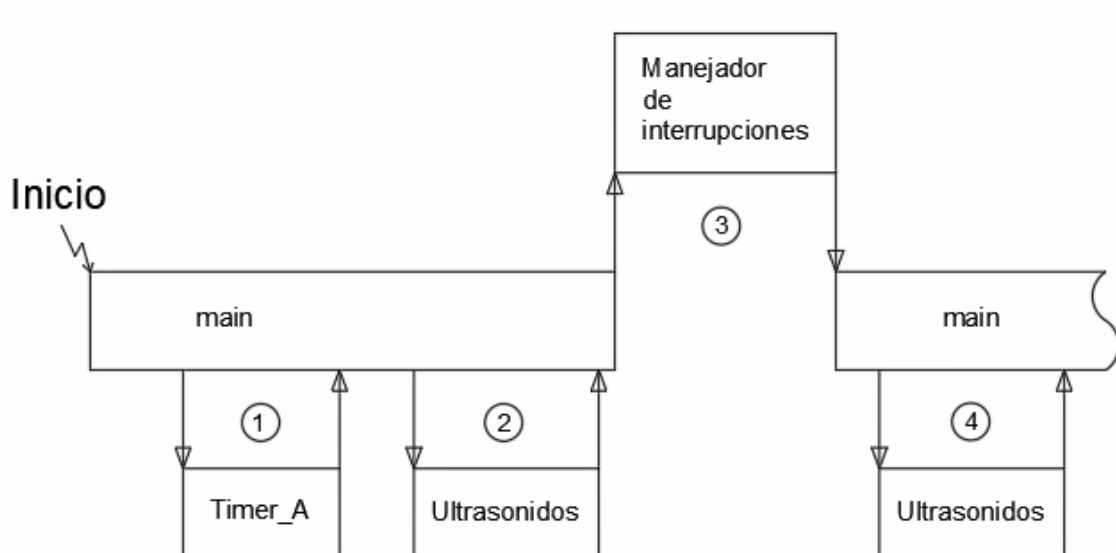


Imagen 17. Esquema uso de interrupciones (producción propia).

En este esquema representado en la Imagen 17, se observa el uso de los módulos que componen el trabajo junto con el manejador de interrupciones:

- Se inicia el fichero principal (main).

- Desde este se pasa a los módulos del TimerA y del ultrasonido para ponerlos en marcha. (1) y (2)
- Es entonces cuando actúan las interrupciones. El sensor envía las señales y se leen los flancos de subida y bajada de la señal ECHO emitida por este.
- Cada flanco activa el flag de interrupción necesario para almacenar el momento que ha ocurrido cada uno. Calculando con estos datos la distancia a la que se encuentre el obstáculo. Además, mientras el manejador de interrupciones está activado, el programa principal se mantiene en pausa. (3)
- Tras el cálculo, se elimina el flag de interrupción correspondiente, el programa vuelve a ponerse en marcha, recibe el valor de la distancia calculado, y envía la información al PLC. Dependiendo de este valor, se considerará que el sensor estará detectando un obstáculo en el interior de la herradura o no. (4)
- A partir de este momento, se repite el uso del manejador con el ultrasonido para detectar los flancos, y de hacer los cálculos recopilando información necesaria del temporizador.

5. Alternativa de sensor: Parte hardware

La parte hardware del producto final quedaría compuesto por: el microcontrolador Mini MSP432, y uno o dos sensores ultrasonidos US100, según desee el cliente.

5.1. Tarjeta microcontrolador MSP432

Se alimenta a una tensión de 3.3V, aunque, a excepción de la tarjeta mini MSP432, a través de uno de sus pines proporciona una tensión de 5V, por si fuera necesario alimentar un periférico a ese valor.

Dependiendo del modelo de la tarjeta que tiene instalado el microcontrolador, esta dará acceso a un número determinado de pines, no a todos. Excepto el puerto que ofrece un suministro de tensión de 5V, el resto de las señales se transportan a 3,3V.

Tanto la tarjeta de desarrollo como la mini MSP432 disponen de leds que se utilizarán para comprobar su correcto funcionamiento.

5.2. Sensor US100

La placa que tiene el sensor de ultrasonidos necesita de una tensión de alimentación de 3,3V.

Los dos canales por los que intercambia información con el microcontrolador son el trigger y el de la señal ECHO. Ambos estarán conectados a pines que tengan acceso a los temporizadores del microcontrolador.

Dispone de un diodo que calcula la temperatura ambiente (Imagen 18), permitiendo obtener resultados de medida más precisos. Este elemento sólo se usa si en lugar de pulsos para TRIGGER y ECHO se utilizan comunicaciones serie UART.

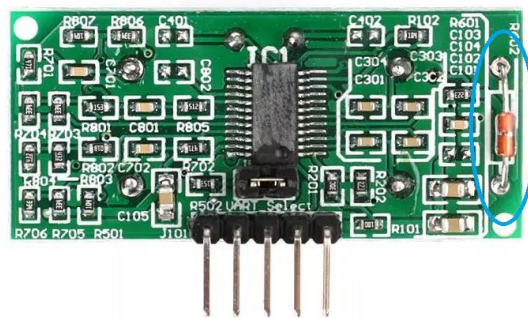


Imagen 18. Diodo sensor de temperatura (tomada de <https://uelectronics.com/producto/sensor-ultrasonico-us-100-compensacion-de-temperatura/>).

6. Alternativa de sensor: Parte software

Aquí se explicarán los módulos que conforman el programa: Timer_A, Ultrasonidos y LEDs.

Los contenidos que forman cada módulo serán, normalmente, el mismo para todos y seguirá el siguiente esquema (Imagen 19):

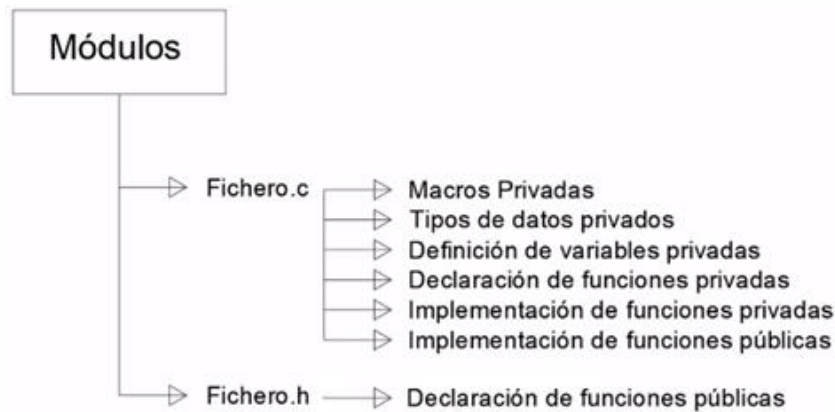


Imagen 19. Esquema general de contenidos de los módulos (producción propia).

Como manera de introducción al apartado de software se hará una breve explicación de los contenidos (Imagen 19) que componen los módulos del programa.

Por otro lado, cabe explicar a qué nos vamos a referir en estos ficheros cuando nos referimos a función “pública” o “privada” y una “declaración” o “definición”:

- Declaración (en la Imagen 20 se observa un ejemplo de declaración de una función): es una notificación al programa para indicarle que vas a crear una función nueva. Esta declaración deberá contener un nombre, una indicación del tipo de función que va a ser y los datos con los que va a trabajar. Si no hay declaración de una función, el programa no la ejecutará.

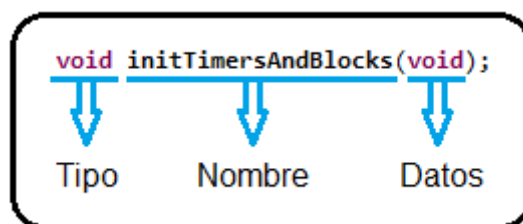


Imagen 20. Estructura declaración (producción propia).

- Definición: es la construcción de una función declarada anteriormente.
- Función pública: es una función a la que se puede acceder y hacer uso desde otros módulos aparte de en el que está definido.
- Función privada: es la función que solo se puede utilizar desde el fichero donde se ha creado.

6.1. Fichero.c

Este fichero del módulo contiene toda la información sobre los datos y funciones privadas, además del desarrollo de las funciones públicas.

6.1.1. Macros privadas

En este primer apartado, el programa realizará un cálculo cada vez que se ponga en marcha la aplicación. Este cálculo obtendrá el dato referido al número de elementos que se va a emplear en cada módulo (número de temporizadores, canales, sensores ultrasonidos y leds).

La fórmula utilizada para obtener este valor se basa en dividir el espacio en memoria del microcontrolador que ocupan todos los datos introducidos de cada elemento del sistema, entre el espacio que ocuparía uno solo. Se puede observar la estructura del código de los dos componentes de la división en la “Imagen 12. Datos temporizadores Timer_A (producción propia).”, de la que se obtendría que se están empleando 2 temporizadores.

6.1.2. Tipos de datos privados

Este segundo apartado, contiene los parámetros con los que se caracterizarán los distintos elementos del sistema.

Se puede observar un ejemplo, para el caso de los temporizadores, en el lado izquierdo de la imagen adjunta “Imagen 12. Datos temporizadores Timer_A (producción propia).”.

6.1.3. Definición de variables privadas

En este apartado, se rellenarán los parámetros indicados en el apartado anterior.

Se puede observar un ejemplo, para el caso de los temporizadores, en el lado derecho de la imagen adjunta “Imagen 12. Datos temporizadores Timer_A (producción propia).”.

6.1.4. Declaración de funciones privadas

El apartado contendrá las declaraciones de las funciones que solo se vayan a utilizar en el “Fichero.c” del módulo.

En el trabajo, con ayuda del esquema de la Imagen 17, se observa que estas funciones corresponden a las que se encargan de configurar y poner en marcha los distintos elementos, como los interruptores y canales y los sensores de ultrasonidos, además de las que inician las interrupciones.

6.1.5. Implementación de funciones privadas

Contiene el desarrollo de las funciones escritas en el apartado anterior.

6.1.6. Implementación de funciones públicas

Este último apartado del “Fichero.c” tiene las funciones declaradas en el “Fichero.h”, o también otras funciones como las declaradas dentro del microcontrolador que permiten el uso de algún elemento interno, en este caso del manejador de interrupciones.

Estas funciones suelen ser: solicitud de información que almacena el módulo (características del periodo de un temporizador, el valor de distancia medido por el sensor...), iniciar una función que pone en marcha los distintos elementos, o también utilizar estos elementos directamente (encender o apagar los LEDs).

6.2. Fichero.h

Finalmente, este segundo fichero del módulo se basará en la declaración de las funciones públicas que explicadas anteriormente.

6.3. Esquema de las funciones

A continuación, antes de pasar a explicar en detalle qué hace cada función, repasaremos las funciones principales que componen el programa, encargadas de interactuar con los distintos elementos del sistema (Imagen 21), y cómo son empleadas:

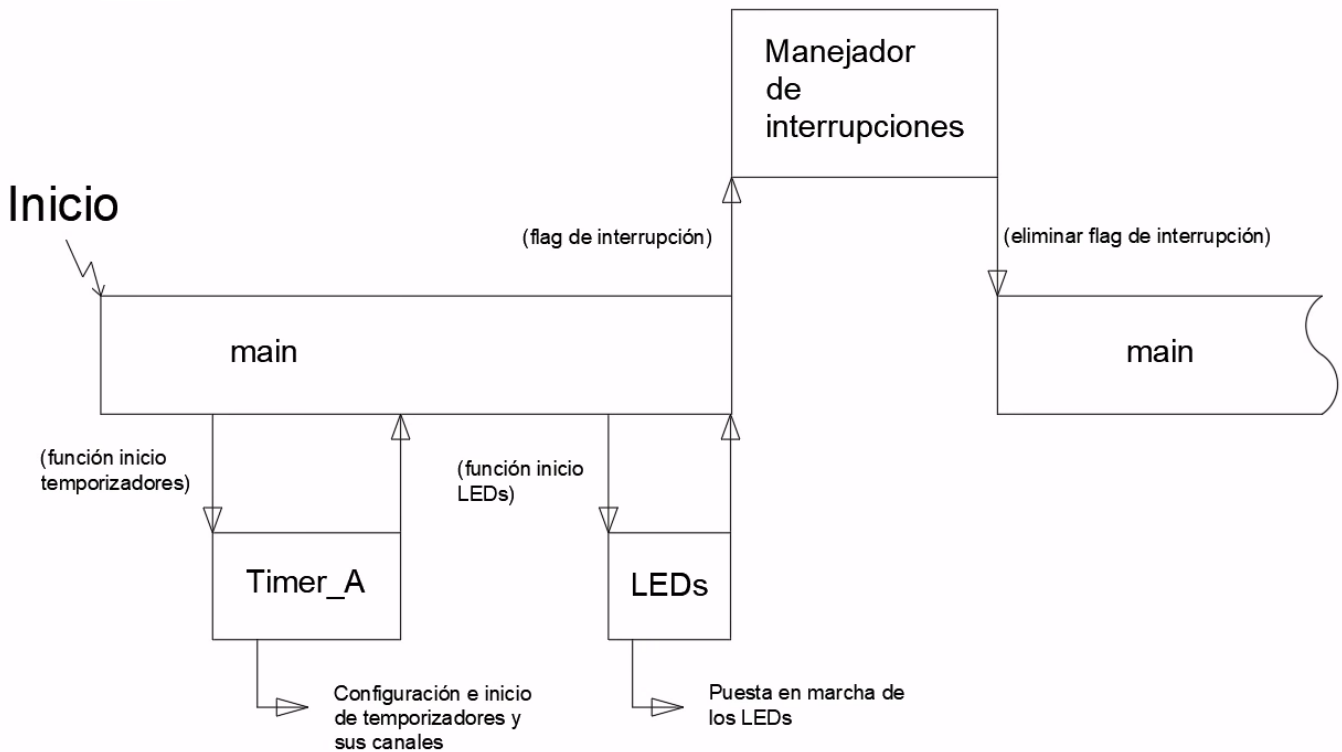


Imagen 21. Pasos seguidos del programa (producción propia).

Desde que se inicia el programa, hasta que las medidas del sensor se ponen en funcionamiento, en el programa sucede:

- Las funciones que ponen en marcha los temporizadores y sus canales:

Por un lado, los temporizadores, antes de ponerse en marcha, se tienen que parametrizar e indicar cómo se va a trabajar con ellos.

Por otro lado, todos los temporizadores tienen un total de cuatro canales, o bloques, (Imagen 14) que nos permite utilizar cada uno para, o bien comparación para emitir señales PWM, o bien hacer capturas de tiempo del temporizador. La selección de qué canal se encargará de utilizar un modo o el otro lo hará la función correspondiente en base a los datos introducidos anteriormente en el programa.

Por último, se utilizará una tercera función que configura los pines de manera que cada uno se adapte al canal del temporizador elegido.

- Luego pasamos a la función de inicio de los LEDs se encarga de apagar todos los LEDs que puedan estar encendidos.
- Por último, a medida que el temporizador avanza en su cuenta, este activa unos flags de interrupción que reinician su cuenta, junto con los flags vinculados a cada canal de temporizador. Aquí intervienen las funciones que se encargan de manejar las interrupciones.

Además, dependiendo de qué interrupciones se activen, iniciaremos las funciones que nos interese ejecutar. Como, por ejemplo, una que almacene el conteo de cuánto tiempo lleva en marcha del temporizador, u otra que envíe la captura realizada por un canal a otro sitio donde se requiera.

Las funciones con las que se configuran los pines utilizados en los LEDs y los canales de los temporizadores usan los registros GPIO. Estos registros se pueden consultar de la ficha técnica del microcontrolador MSP432 ([7] Pág. 678-679), donde con estos, por ejemplo, se indica si un pin recibe o emite información (“DIR”), si es de solo lectura (“IN”), si utiliza la potencia habitual o más elevada (“DS”), entre otros.

Una vez está el sensor en pleno funcionamiento, las funciones que intervienen vendrán basadas del siguiente esquema (Imagen 22):

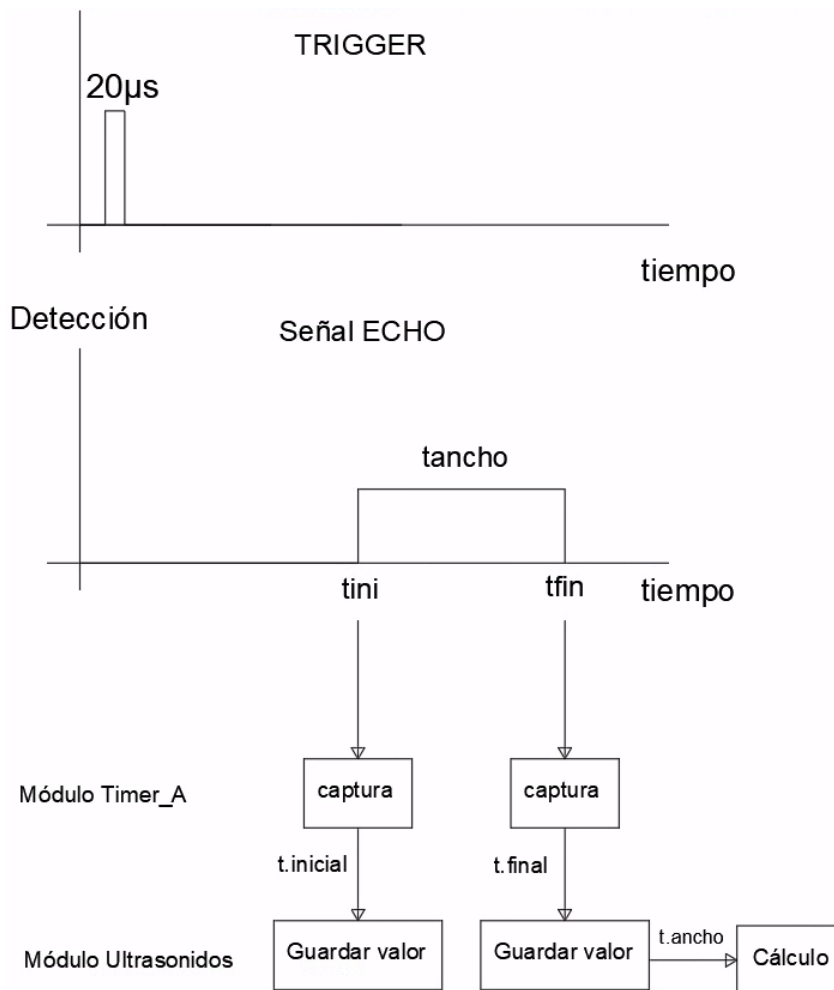


Imagen 22. Proceso de detección del sensor (producción propia).

En esta imagen observamos cómo un pulso de señal transmitida desde el temporizador al sensor por el canal TRIGGER de 20 microsegundos. Una vez recibida por el sensor, pasado un margen concreto de tiempo, este inicia una medida representada como un pulso que significa en el tiempo que ha transcurrido desde que el sensor ha emitido una señal ultrasónica hasta que este mismo la recibe.

En el transcurso de este proceso, se han producido dos interrupciones en el momento que se ha realizado cada una de las capturas:

- Tras activarse cada interrupción e intervenir las funciones del manejador de interrupciones, debido a que estas interrupciones se han activado a causa de las capturas hechas por el temporizador, provocamos que se inicie la función que transmite la medida obtenida de tiempo de la captura al módulo de ultrasonidos.

- Una vez en el módulo de interrupciones, se almacena este valor de tiempo en las variables que contienen la información de lectura de los sensores, sobre escribiendo su valor anterior.
- Por último, cada vez que se produce la captura de la recepción de la señal ultrasónica, se activa una función que hace el cálculo del ancho del pulso, almacenando el valor resultante dentro de la variable de los datos de los sensores.

6.4. Módulo Timer_A

A continuación, procedemos a desarrollar qué parámetros van a contener los temporizadores y canales, junto con las funciones que van a hacer uso de todos ellos.

6.4.1. Macros privadas

Aquí se obtienen los datos referidos al número de temporizadores y bloques que se van a utilizar. Consiste en una división entre el espacio en memoria que ocupan los temporizadores, o canales, que hayamos definido, dividido entre lo que ocupa uno solo.

6.4.2. Tipos de datos privados

En este apartado se escriben los parámetros de los siguientes datos:

- Los temporizadores:
 - Indicar su periodo.
 - Los registros del temporizador donde se anotará su forma de trabajo.
 - El reloj que se va a utilizar con sus divisores de frecuencia.
 - Indicar si se van a utilizar interrupciones.
- Los pines:
 - Cuales se van a utilizar.
 - Modo de funcionamiento de cada uno.
 - Si son pares o impares.
 - Se necesita saber si están para uso libre o si ya están conectados. Para ello, se va a hacer uso de la ficha técnica de la tarjeta de Texas Instruments. ([2] Para el caso de la tarjeta MSP432, y [8] para la tarjeta mini MSP432)

- Los bloques de los temporizadores:
 - Es donde se parametrizan los pines a los pines, ya que cada bloque utilizará el pin que vayamos a utilizar.
 - A qué timer pertenece el bloque.
 - Cuál de los 4 bloques del timer se utiliza.
 - Se pueden utilizar tanto para generar una señal PWM como para capturar flancos.
 - Se indica el ancho de pulso para el caso de señales PWM.
 - Se indica el tipo de terminal del pin. ([2] Para el caso de la tarjeta MSP432, y [8] para la tarjeta mini MSP432)
 - En caso de estar en modo captura de flancos, indicar si se refiere a detección del de subida, de bajada, o ambos.
 - Indicar si se va a hacer uso de interrupciones.

6.4.3. Definición de variables privadas

A continuación, se procede a asignar los valores a los parámetros indicados en el apartado anterior.

Por una parte, la frecuencia del reloj con la que se va a trabajar va a ser de 1 Hz, resultado de la división entre la frecuencia del reloj del temporizador, 3 MHz, por su divisor secundario.

Por otra parte, se van a rellenar los datos para tener dos pines en funcionamiento: uno en modo PWM, que se encargará de producir la señal de 20 microsegundos que asegura el funcionamiento constante de los sensores; y el segundo en modo captura para obtener los tiempos de los flancos del sensor.

Además, los pines encargados de la captura de flancos deberán hacer uso de las interrupciones ya que:

- Emplean una función en ese momento para guardar una captura del tiempo en el que ocurre cada uno.
- En el flanco de bajada tras la captura del flanco, se hará el cálculo entre la captura del flanco de bajada menos el de subida, obteniendo el valor del ancho del pulso y, dependiendo de si es menos o más ancho, determinar si hay un objeto más próximo de lo normal o no.

6.4.4. Declaración de funciones privadas

Las declaraciones de funciones y variables se hacen para que el programa pueda acceder a ellas, si no se declara, la variable/función no se podrá usar.

Por lo tanto, en este apartado se incluirán las funciones que tengan que ver con la puesta en marcha de los temporizadores y la parametrización de los pines. También se incluirán las funciones encargadas de controlar las interrupciones.

6.4.5. Implementación de funciones privadas

El siguiente paso será escribir el código de las funciones privadas que se utilizarán en el módulo Timer_A:

6.4.5.1. *Iniciar temporizador en modo UP ([7] Pág. 786)*

Este es uno de los modos de trabajo que tiene el timer A. Consiste en que cuando el temporizador alcanza el valor del periodo indicado en los parámetros del temporizador (CCRO), se activa un flag de solicitud de interrupción (CCIFG), y tras un ciclo de reloj más, el temporizador comienza de nuevo la cuenta desde cero, activando otra solicitud de interrupción (TAIFG). Este proceso se repite indefinidamente mientras el temporizador esté activado (véase bien en la Imagen 23).

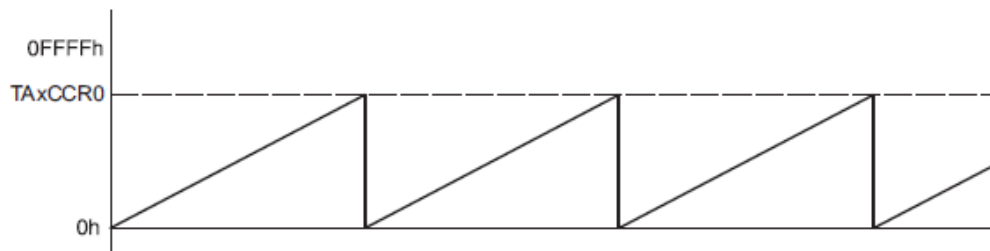


Figure 19-2. Up Mode

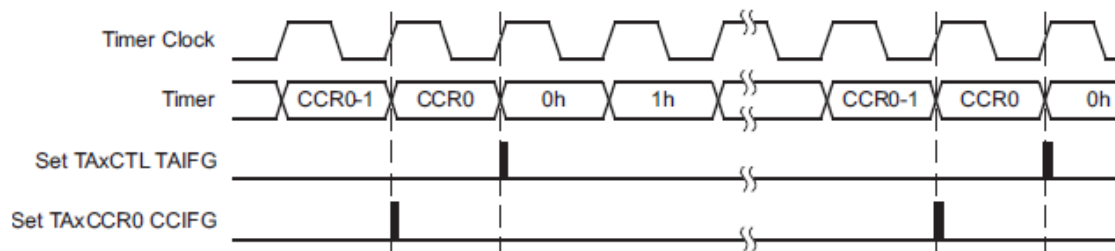


Figure 19-3. Up Mode Flag Setting

Imagen 23. Temporizador en modo UP (tomada de <https://schaumont.dyn.wpi.edu/ece4703b21/downloads/8bf98313124444641502c686bb90dbaa/msp432p401r-trm.pdf>, Pág. 787).

El uso de esta función es configurar uno de los bloques de un temporizador para que funcione en modo UP, esta configuración se hace siguiendo la lista de registros que hay en el manual técnico ([7] Pág. 796-802).

El primer paso es detener el temporizador (MC) y hacer una limpieza (CLR) de los valores que tiene actualmente el temporizador, entonces procedemos a indicar asignando valores en los campos de cada registro cómo se va a utilizar el temporizador.

Por un lado, en el bloque temporizador tenemos el “registro de control” (CTL), en el que indicamos cuál de los relojes que hay en el microcontrolador se va a utilizar (SSEL), además de qué divisor primario de frecuencia se le va a aplicar (ID).

También se desactivan tanto el flag de interrupción (IFG) como la habilitación de interrupciones (IE), añadiendo a continuación un condicional donde, si en los parámetros que hemos indicado del modo UP, el uso de interrupciones está activado, estas se habilitarán del registro de control del timerA.

Para terminar con este bloque, en el “registro de expansión” (EX0), se añadirá el divisor secundario (IDEX) que se aplicará a parte del primario, así se consiguen más combinaciones con las que encontrar la frecuencia del reloj deseada.

Por otro lado, en el bloque de comparación y captura, como el modo UP se va a utilizar para compararlo con la señal del canal ECHO del ultrasonido, configuramos este bloque para que trabaje en modo comparación. Para ello, en el “registro de control de comparación o captura”, se asigna valor cero al campo “modo captura” (CAP). Ya entonces, volvemos a desactivar el flag de interrupción (CCIFG) y la habilitación de interrupciones (CCIE) de este registro, de nuevo añadiendo otro condicional que indica que, si las interrupciones están habilitadas, se utilizarán del registro de control de comparación o captura. Por último, en el “registro de comparación o captura” (CCR), al trabajar en modo comparación, se almacena el valor del periodo del reloj y se le resta un ciclo de reloj.

Tras estas configuraciones se habilita el controlador de interrupciones del microcontrolador.

6.4.5.2. *Inicio del temporizador en modo UP ([7] Apartado 19.2.3)*

Para inicializar el temporizador, simplemente eliminamos el modo que haya en el momento en el campo del control principal, y se asigna el modo UP como nuevo modo de funcionamiento.

6.4.5.3. *Iniciar los bloques de los temporizadores*

Antes de comenzar a configurar los bloques se obtienen las direcciones de los registros del timer y el canal que se va a parametrizar.

Dependiendo de si el bloque se va a utilizar para generar una señal PWM o para capturar un valor del temporizador, los registros que se configurarán son:

- Generar señal PWM:
 - En el campo captura o comparación (CAP), elegir el modo comparación.
 - Configurar el modo de la unidad de salida para la generación de señales PWM.
 - Indicar qué terminal tiene (CCIS) el pin emisor de la señal extrayendo el dato de los parámetros del pin.
 - Indicar el ancho de pulso de la señal.
 - Se desactiva el flag de interrupción y se deshabilitan las interrupciones, ya que no se van a utilizar para este funcionamiento.
- Captura de un flanco:
 - Utilizar el modo captura.

- Indicar el terminal del pin, indicar qué tipo de flanco va a capturar.
- Sincronizar la señal ECHO del sensor con el reloj del temporizador.
- Se desactivan tanto el flag de interrupciones como la habilitación de interrupciones, añadiendo a continuación un condicional que las habilitará si así se indica en los parámetros del pin.

Al acabar la configuración de los bloques, se inicia la de los pines.

6.4.5.4. *Iniciar pines del temporizador*

Los pines suelen tener distintos nombres que dependen de la parte del microcontrolador al que están conectados ([9] Apartado 4.2). Para este trabajo se van a buscar pines conectados al temporizador timerA. Para ello, se buscarán en la ficha técnica del micro, como, por ejemplo, los pines “5.6/5.7/6.6/6.7” para el timerA2.

Tras tener las direcciones deseadas, se configura el resto del pin accediendo a sus registros ([7] Apto. 12.2). Dependiendo del uso que se le dé a cada uno: el pin se redireccionará (“SEL0” y “SEL1”) del GPIO al canal del temporizador que esté configurado en modo de comparación (*Imagen 14*), donde se genera la señal PWM y será un pin de salida (“DIR” y “REN”); y el resto de los pines que son de captura de flancos, se redireccionarán a los canales configurados en modo captura y serán de entrada.

6.4.5.5. *Función de reacción a una interrupción del registro de comparación o captura del timerA “0”*

Los cinco canales que hay en cada temporizador están organizados de forma que el canal 0 siempre se utiliza para indicar el modo de funcionamiento del temporizador, mientras que los demás se utilizan para comparación o captura. Es por esto por lo que se utilizan varias funciones.

Esta función pasa por cada uno de los temporizadores del microcontrolador, desactivando la interrupción que ha activado la función de comparación y captura (CCIFG), y llevando un conteo del tiempo transcurrido de la aplicación.

6.4.5.6. *Función de reacción a una interrupción del registro de comparación o captura del timerA "OTAIFG"*

Esta función hace lo mismo que la anterior, solo que esta se pone en marcha cuando se activa la interrupción del temporizador (TAIFG), y de la que no se hace ningún uso durante todo el trabajo.

6.4.5.7. *Función de reacción a una interrupción del registro de comparación o captura del timerA "N"*

Esta función se encarga de realizar las capturas de los flancos de subida y bajada acudiendo a la función "Callback" de captura. A esta le envía los datos del timer y bloque que ha activado el flag de interrupción, además de la captura del tiempo, y si el flanco que ocurre es de subida o de bajada.

6.4.5.8. *Implementación de funciones públicas*

A continuación, procederemos a explicar el desarrollo de las funciones públicas referentes a las declaradas en el fichero "Timer_A.h". Si es necesario, estas funciones se podrán utilizar en otros módulos. De esta forma, se transmite información desde este que tiene la información referida a los temporizadores a donde se pida.

6.4.5.9. *Inicio de temporizadores y bloques*

Esta función recorrerá todos los temporizadores y canales, llamando para cada uno a las funciones encargadas de configurar y poner en marcha todos estos. Estas funciones pertenecen a las funciones privadas desarrolladas en este mismo fichero.

6.4.5.10. *Obtener periodo del timerA "N"*

Esta función se encargará de dar el valor del periodo del temporizador que se pida.

6.4.5.11. *Obtener valor del timer*

Esta función se utilizará para, dependiendo de qué valor del canal recibe como valor de entrada, devolverá a qué temporizador pertenece este.

6.4.5.12. *Desarrollo de los vectores de interrupción*

Cada vez que se completa un periodo de un temporizador, se activa la interrupción "CCIFG" de cada uno. Estas funciones indican qué se hace cada vez que ocurre una interrupción.

Para esta parte se van a desarrollar todas las funciones que se distinguen por las referidas a cada temporizador del microcontrolador y, a partir de estas, se distinguirán de nuevo entre los canales cero ("0"), que indican el modo de funcionamiento, y el resto ("N"), que indican si se utilizan como comparación o captura.

Dependiendo del temporizador y canal desde donde ocurre la interrupción, se iniciará la función de reacción a interrupciones correspondiente.

6.4.5.13. *Definición de bajo nivel de las funciones "Callback"*

Este tipo de funciones "Callback" actúan de forma que, en este módulo se hace una definición de baja prioridad, de manera que, cuando se llame a la función para utilizarla, se redirija a la función definida prioritaria, que estará en otro módulo. En este caso, se encontraría en Ultrasonidos.

Estas se llaman desde las funciones que controlan las reacciones contra interrupciones.

6.4.5.14. *Declaración de funciones públicas*

Contendrá las declaraciones de las funciones públicas explicadas en el anterior fichero. Los archivos que tendrán acceso serán: el fichero "Timer_A.c" que escribe el código de todas ellas,

y también el fichero principal, que hará uso de estas para poner en marcha los temporizadores, canales y pines.

6.5. Módulo Ultrasonidos

Este módulo se va a encargar de almacenar la información referida a las medidas hechas por cada uno de los ultrasonidos que estén conectados. No va a contener funciones privadas ya que no hay configuración alguna sobre los ultrasonidos.

Al igual que en todos los módulos, el fichero de variables y funciones privadas debe tener acceso a las declaraciones públicas. Además, se añadirán las declaraciones del módulo del temporizador para, en caso de ser necesario, se pueda acceder a cualquier información necesaria que este contenga. Aunque, para el caso de este trabajo, las únicas funciones que se van a desarrollar son:

- La función que proporciona el dato de distancia calculado de cada uno de los sensores.
- La función que proporciona el dato de cuántos ultrasonidos se va a utilizar.
- Las funciones “Callback” encargadas del conteo de tiempo en funcionamiento de los sensores, y la que hace la recopilación de las capturas hechas en el módulo Timer_A, junto con el cálculo del ancho de pulso de la señal ECHO.

6.5.1. Macros privadas

El único dato que va a contener es el número de ultrasonidos declarados. El cálculo se hace de la misma forma que con el número de temporizadores del módulo anterior.

6.5.2. Tipos de datos privados

Este apartado contiene dos datos:

- Datos con los que parametrizar los ultrasonidos:
 - o Qué bloque de los definidos en el módulo “Timer_A” utiliza como trigger.
 - o Qué bloque utiliza para capturar los flancos de subida.

- Qué bloque utiliza para capturar los flancos de bajada.

Luego, para que cada sensor guarde los valores capturados en los flancos, con los que calcular las distancias de las mediciones, se almacenarán aquí también:

- El valor de captura de tiempo del flanco de subida.
- El valor de captura de tiempo del flanco de bajada.
- El ancho del pulso calculado de la resta de los dos anteriores.
- El cálculo de distancia.
- El número de periodos que ha pasado desde que ha comenzado el programa.

Estos datos tendrán una capacidad de memoria de 64 bits para que no produzca problemas.

- El dato del divisor que transforma el ancho de pulso medido en microsegundos a una medida de distancia en centímetros. Considerando una temperatura ambiente de 21,6°C y humedad en el aire del 40%, este valor es de “58”. [6]

6.5.3. Definición de variables privadas

En este apartado se parametrizan los sensores que se vayan a utilizar.

Por un lado, en base a qué temporizador se va a utilizar y cuantas capturas de flancos se vayan a tomar, se rellenarán los datos referentes a los tres primeros parámetros del apartado anterior.

Por el otro lado, los valores iniciales de tiempos y distancia se pondrán nulos. Una vez comience el programa se actualizarán y funcionarán correctamente.

6.5.4. Implementación de funciones públicas

Desde este apartado se redactarán las funciones que dan información sobre los sensores, y se definirán las funciones “Callback” que no se hizo en el módulo de los temporizadores.

6.5.4.1. *Valor de distancia*

Recibe el valor de un sensor y devuelve la distancia que contiene el parámetro de ese sensor.

6.5.4.2. *Número de sensores ultrasonidos*

Devuelve el número de los sensores ultrasonidos instalados.

6.5.4.3. *Función "Callback" de periodo*

Cada vez que se activa la interrupción de fin de periodo (CCIFG o TAIFG), aumentará en 1 el valor del número de periodos transcurridos para el canal trigger de los sensores.

6.5.4.4. *Función "Callback" de captura*

En esta función, dependiendo de qué bloque haya activado el flag de interrupción (CCIFG), se guardará el valor de la captura del flanco correspondiente en los parámetros del ultrasonido al que pertenezca ese bloque.

También se hará el cálculo de la distancia en el momento que se capture el flanco de bajada.

6.5.5. Declaración de funciones públicas

En este caso, como no se va a realizar ningún control respecto al funcionamiento del sensor de ultrasonidos, las funciones que se van a declarar son:

- La función que devuelve el número de sensores del programa.
- La función que devuelve la distancia calculada por un sensor.

6.6. Módulo LEDs

Como complemento de ayuda en el desarrollo del programa, y gracias a que en ambas tarjetas hay instalados LEDs a los que se puede acceder, se va a crear este módulo que configura y controla estos LEDs que hay conectados al microcontrolador.

Solo hay un inconveniente, y es que, al ser dos tarjetas diferentes, los LEDs estarán conectados a pines distintos, por lo que habrá que actualizar el programa dependiendo de qué tarjeta se vaya a utilizar.

6.6.1. Macros privadas

Al igual que en los otros dos módulos, se calcula el número de LEDs que se van a utilizar.

6.6.2. Tipos de datos privados

En este apartado se indican qué parámetros hay que introducir para cada pin conectado a un LED:

- Se indica el pin que se vaya a utilizar.
- Se indica si el número del pin es par o impar.

6.6.3. Definición de variables privadas

A continuación, se rellenan esos parámetros dependiendo de qué tarjeta se utilice.

6.6.4. Declaración de funciones privadas

Las funciones privadas declaradas en este módulo serán las encargadas de la configuración de los pines, distinguiendo entre:

- Configuración de pines pares.
- Configuración de pines impares.

6.6.5. Implementación de funciones privadas [x]

Ahora hay que desarrollar las funciones declaradas en el anterior apartado.

Las configuraciones de los pines que controlarán los LEDs de la tarjeta serán las mismas tanto para los pines pares como los impares:

- Los pines se utilizan en modo de funcionamiento GPIO (“SEL0” y “SEL1”).
- Al ser pines de alimentación a LEDs, se utilizarán únicamente como salida de información (“DIR”).
- Por último, se indica que los pines van a trabajar a potencia habitual (“DS”).

6.6.6. Implementación de funciones públicas

Al ser funciones públicas, van a ser accesibles desde otros módulos. Es por eso por lo que van a consistir en el encendido y apagado de los LEDs, además de la función de inicialización.

6.6.6.1. *Puesta en marcha de los LEDs*

Esta función pasa por todos los LEDs que se vayan a utilizar, los configura uno a uno utilizando las funciones anteriores explicadas de configuración, y los apaga para hacer el caso inicial.

6.6.6.2. *Encender LED*

Utilizando el registro "OUT", que indica el valor que tiene el pin, simplemente le ponemos valor "1" para encenderlo.

6.6.6.3. *Apagar LED*

Al igual que la función anterior, se utiliza el registro "OUT" para poner al pin un valor de "0".

6.6.7. Declaración de funciones públicas

Por último, y como siempre, las funciones que se vayan a utilizar en otros módulos deben declararse en el fichero correspondiente, "LEDs.h".

6.7. Fichero principal "main.c"

Este fichero se va a encargar de poner en marcha las funciones encargadas de iniciar los temporizadores y canales del módulo de temporizadores. También enviará, a través del canal de comunicación con el módulo de control, los resultados de detección de los sensores en todo momento.

Al ser este el fichero principal, va a necesitar el acceso a todos los módulos de los distintos elementos intervienen en el trabajo. Esto se realiza añadiendo los ficheros públicos de cada uno de los módulos.

6.7.1. Código del programa

Lo primero que debe hacer el programa es poner en marcha los temporizadores y canales. Esto provoca que, como ya se ha explicado, el ultrasonido reciba constantemente la señal que le indica que empiece a detectar.

Con los temporizadores en funcionamiento, los propios ultrasonidos estarán midiendo el tiempo que tarda en recibir la señal eco desde que se emite, traduciendo al mismo tiempo qué distancia corresponde. Entonces, en el propio programa, se llamará a todos los ultrasonidos que estén conectados, pidiéndoles a cada uno qué distancia han medido. Para terminar, dependiendo de la distancia que hayan marcado, el microcontrolador enviará las señales de detección o no detección de los ultrasonidos utilizando los LEDs de la tarjeta para indicarlo.

7. Tests y resultados

Una vez terminado el programa, solo queda iniciarlo y observar cómo se comportan los sensores. Para ello se va a utilizar un osciloscopio para ver las señales de entrada y salida de este sensor, donde el montaje sería (Imagen 24):

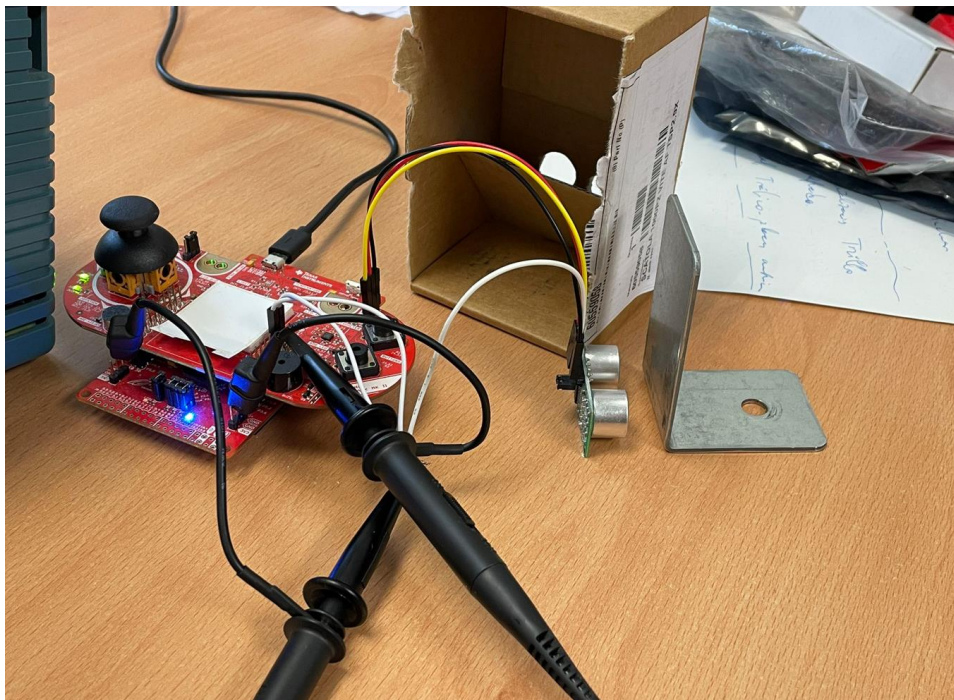


Imagen 24. Montaje testeo (producción propia).

Las pruebas para la comprobación de la aplicación se harán utilizando un solo sensor, y con la tarjeta grande MSP432.

Para la comprobación se colocará una pletina que actuará como pared de la herradura, y con una segunda se atravesará en el rango de detección del sensor para simular las placas de metal de la instalación (Imagen 25):

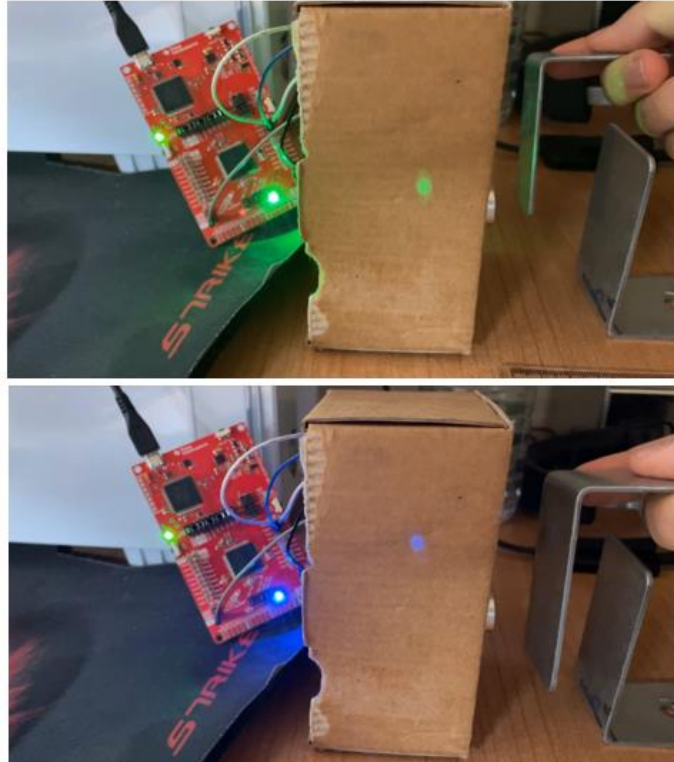


Imagen 25. Simulación de placa atravesando la herradura (producción propia).

En las imágenes se puede distinguir una luz led que cambia de color en función de si el sensor detecta (azul) o no detecta (verde).

Por otro lado, el osciloscopio se encargará de mostrar las señales de entrada y salida del sensor: la señal trigger y la señal ECHO. En un principio, con el pulso de 20 microsegundos del trigger, un tiempo después, el sensor generará la señal ultrasónica para la detección:

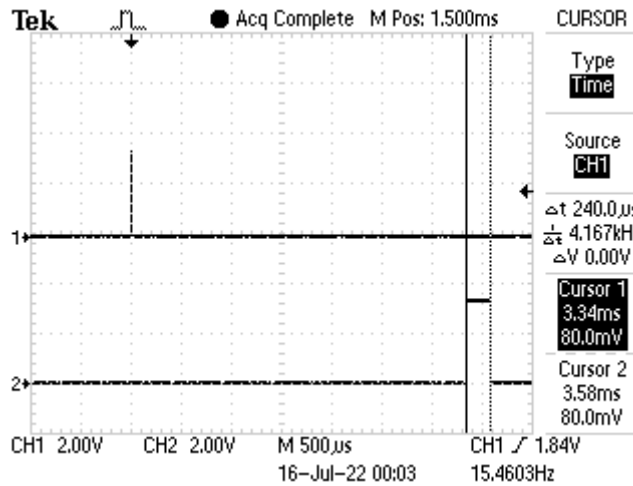


Imagen 26. Señal osciloscopio sin detectar objeto (producción propia).

Como se puede observar en la Imagen 26, la puesta en marcha del sensor es como se esperaba. Tras la señal trigger, que envía el pulso de 20 microsegundos, tras un margen de tiempo, el sensor envía el ultrasonido que genera un pulso a través de la señal ECHO. Este pulso es de aproximadamente 232 microsegundos, lo que nos indica que el ultrasonido ha atravesado toda la anchura de la herradura, los 4 centímetros, y que no se ha cruzado un obstáculo.

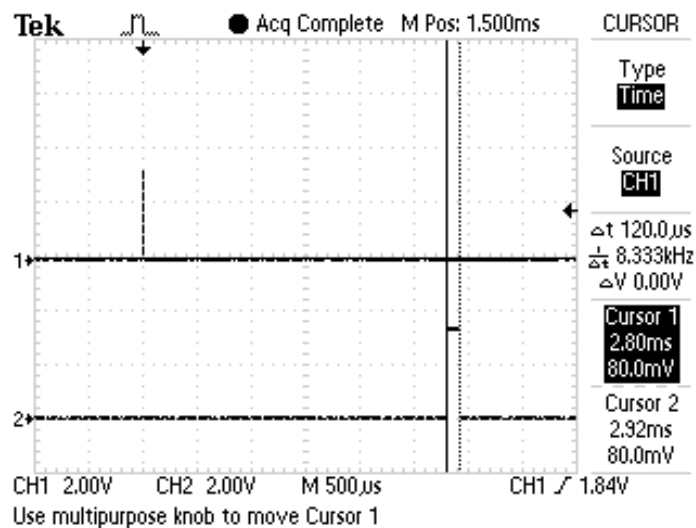


Imagen 27. Señal osciloscopio detectando objeto (producción propia).

En cuanto a lo que podemos observar en la Imagen 27, volvemos a ver el pulso del trigger que inicia el sensor al igual que antes. En este caso, el pulso de la señal ECHO tiene una anchura aproximada de 116 microsegundos con el que, gracias al cálculo para pasar a distancia, el eco ha rebotado a unos 2 centímetros desde la emisión de la señal, inferior a los 4 centímetros de anchura de la herradura, y significando en una obstrucción de la señal ultrasonido.

8. Conclusiones

Tras haber realizado el desarrollo del programa y las pruebas con un producto modelo, demostramos que el producto puede funcionar a falta de probarlo en el entorno laboral.

En este trabajo se buscaba crear y sustituir unos sensores en formato herradura, que emplean sensores fotoeléctricos. Se ha optado por utilizar el sensor de ultrasonidos y ha significado que la herradura distará, en tamaño, de la original. A causa de esto, se ha podido plantear la opción de juntar dos sensores por herradura y así recuperar parte de ese espacio perdido.

Además, con este intercambio al sensor de ultrasonidos conseguimos eliminar los problemas que causaba el uso del otro sensor. Principalmente, las interrupciones que se producían debido a la suciedad que hacía al sensor fotoeléctrico comunicar falsa detección. Gracias a esto se consigue mayor eficiencia tanto del sensor como de la máquina que la emplea.

9. Presupuesto

Este producto muestra una nueva forma de aplicación para los sensores ultrasónicos. Da la oportunidad de plantear otras opciones a la hora de seleccionar qué tipo de sensor necesita la instalación, y de cuál de ellas beneficiaría la eficiencia de la producción. Además, al utilizar este tipo de sensor, al igual que tiene ambientes de trabajo perjudiciales debido a su forma de funcionamiento (instalaciones con mucho ruido), también significa que puede trabajar donde otros causarían problemas.

Como ya hemos comentado anteriormente, el producto está compuesto por la carcasa de metal, uno o dos sensores US100, una tarjeta mini con microcontrolador MSP432P401R. Además, también añadir: los conectores de panel, macho y hembra; el regulador de tensión; los dos condensadores de tantalio; tornillos M1 y M2,5; y el cableado.

9.1. Costes y precios

Por un lado, el coste de abastecimiento consta de:

- Sensores US100.
- La tarjeta mini con el microcontrolador MSP432P401R.
- Los conectores de panel macho y hembra.
- Un regulador de tensión de 12V a 3,3V.
- Dos condensadores de tantalio de 10 microfaradios.
- 1 tornillo M2.5.
- 4 tornillos M1.
- Cableado interno de 0,5 milímetros.

Además, para la opción de compra de la pareja de pletinas para la instalación incluye:

- 2 pletinas.
- 4 tornillos M2,5.

Por otro lado, en cuanto a los costes de materia prima, necesaria para la fabricación de la herradura, se utilizará el aluminio 5052 debido a: su bajo coste, alta flexibilidad, resistencia a la oxidación y buena soldabilidad. [10]

El coste que tendría la compra de todos los componentes del prototipo para la montura sería (Tabla 1):

Tabla 1. Tabla de costes de los materiales (producción propia).

| Sensor de Ultrasonidos de herradura | Nº | Coste und. | Coste total [€] |
|---|----|------------|-----------------|
| Fabricación de carcasa completa | 1 | 63,70 | 63,7 |
| US100 (sensor) | 2 | 3,95 | 7,90 |
| MCP432 (microcontrolador) | 1 | 27,55 | 27,55 |
| Tornillo M1 | 4 | 0,01 | 0,04 |
| Tornillo M2,5 | 1 | 0,17 | 0,17 |
| Conector de panel macho | 1 | 7,23 | 7,23 |
| Conector de panel hembra | 1 | 5,13 | 5,13 |
| Cable 0,5mm [unidades en m] | 1 | 0,15 | 0,15 |
| LM11173V3 (Regulador de tensión 12V-3,3V) | 1 | 1,40 | 1,40 |
| T491C106M010AT (Condensador 10µF) | 2 | 1,04 | 2,08 |
| Coste total | | | 115,35 |

Mientras que por cada lote de pletinas se añadirían (Tabla 2):

Tabla 2. Tabla de costes del lote extra (producción propia).

| Extra | Nº | Coste u | Coste total [€] |
|--------------------------------|------|---------|-----------------|
| Aluminio 5052 [unidades en kg] | 0,08 | 2,74 | 0,22 |
| Tornillo M2,5 | 4 | 0,17 | 0,68 |
| Coste total | | | 0,90 |

Estos precios son para la fabricación de un solo producto, por lo que, si llegara a producirse a nivel industrial, significaría en una reducción del coste ya que, aparte de ser un comprador habitual, los costes fijos producidos de la maquinaria utilizada serían mucho menores.

Por ejemplo, he solicitado un presupuesto aproximado de cuánto sería el coste de fabricación de la carcasa que he diseñado, el precio alcanzaría un valor de 63,70 € para el caso de fabricación de una unidad. Pero, en el caso de que fuera un lote de 100 unidades, el precio por unidad de la herradura completa sería más o menos 100 veces inferior, por lo que el coste final de compra sería mucho más reducido que el mostrado en la tabla.

Por otro lado, aún quedaría obtener los costes producidos por el presupuesto que conllevaría contratar a un trabajador para que realizara este diseño y desarrollo. Este presupuesto contendría el salario de este empleado, además del coste de amortización del ordenador

utilizado junto con el programa en el que se han realizado las pruebas de programación del programa y el programa de diseño de la herradura.

Para empezar, el salario promedio de un ingeniero, sin contar pagas extras de la empresa, según el convenio colectivo [11], para 2022, sería de unos 1832,07 €/mes. Siendo a la hora aproximadamente 11,45 €/hora, el salario que tendría que recibir el trabajador, asumiendo unas 300 horas de desarrollo y estudio del TFG, serían unos 3435 €. Por lo tanto, como este estudio correspondería más bien a una inversión antes de comenzar con la producción y venta del producto, este dinero invertido se iría amortizando a medida que aumenten las ventas.

Por otra parte, el presupuesto que correspondería con la amortización del ordenador de trabajo utilizado con las licencias del software empleado resultaría:

- Amortización del ordenador: el ordenador utilizado ha sido el mío personal, cuyo precio de compra fueron unos 950 €.

Considerando que el tiempo de vida del ordenador es de 6 años, momento en el que su valor se anularía, el coste de amortización mensual del ordenador sería aproximadamente 13,19€/mes. Este valor resultaría de la fórmula:

$$\frac{950\text{€}}{12 \text{ meses} * 6 \text{ años}} = 13,1944 \text{ €/mes}$$

Por lo tanto, considerando que el TFG ha tomado unos 4 meses de desarrollo, el coste de amortización del ordenador serían 52,78 €.

- Licencias del software empleado: las aplicaciones software utilizadas para el desarrollo del TFG han sido: AutoCAD para el diseño de la herradura, servicio gratuito de la UPV; el compilador Code Composer Studios para la programación del programa, aplicación gratuita; y, por último, las aplicaciones de Office 365, Word, Excel y PowerPoint, servicio gratuito de la UPV.

Con esto finalizaría el coste de lo que supondría el presupuesto completo de este desarrollo realizado. Por lo tanto, para terminar, se va a mostrar una tabla con todos los costes considerados, además de lo que correspondería con los gastos generales y el beneficio industrial (Tabla 3):

Tabla 3. Presupuesto final (producción propia).

| Presupuesto completo | Coste individual [€] |
|--|----------------------|
| Costes de material (extra no incluido) | 115,35 |
| Mano de obra | 3435,00 |
| Amortización del ordenador | 52,78 |
| Licencias de software | 0,00 |
| Gastos Generales (13%) | 468,41 |
| Beneficio Industrial (6%) | 216,19 |
| Coste total | 4287,72 |

El presupuesto resultado del desarrollo de este trabajo asciende a 4287,72 € (Cuatro mil doscientos ochenta y siete euros con setenta y dos céntimos).

10. Pliego de condiciones

10.1. Producto

Sensor de ultrasonido junto con tarjeta mini MSP432 instalados en una herradura construida en aluminio (Imagen 28).

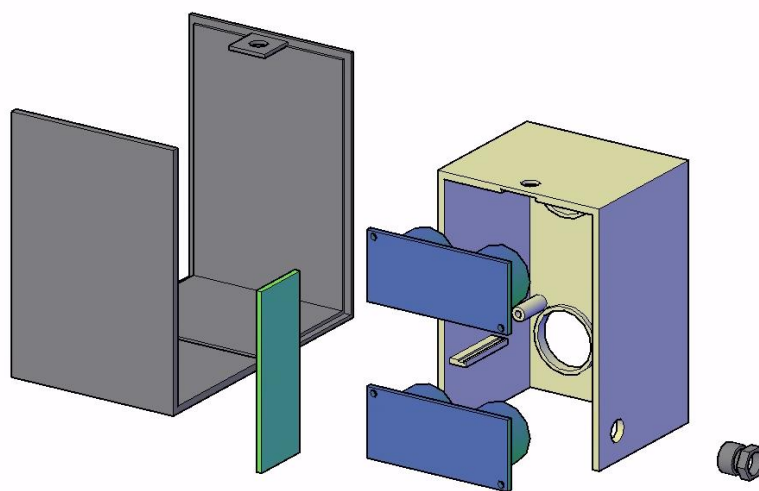


Imagen 28. Herradura despiezada (producción propia).

10.2. Materiales

Herradura (Imagen 29):

- Carcasa compuesta por dos piezas de aluminio que estructura todos los elementos del hardware que intervienen con el sensor.
- El montaje de la herradura es sencillo.
- La herradura puede tener instalada uno o dos sensores según se desee.
- Fácil acceso y manipulación de los componentes instalados en su interior.
- Contorno repasado para suavizar el esquinado.
- Opción de compra de dos tipos de pletina que permitirá acoplar la herradura.
- De color aluminio. Con opción a elegir otro de la paleta de colores RAL.
- Control de calidad: por cada lote de 100 unidades se seleccionarán 4 al azar, en donde se comprobará el estado de la pieza es bueno observando abolladuras, rasguños, defectos de montaje o mecanizado... De esta forma, llevando un seguimiento de qué tipo de desperfecto es y la frecuencia con la que ocurre, se puede llegar a saber qué está ocasionándolo y actuar en consecuencia.

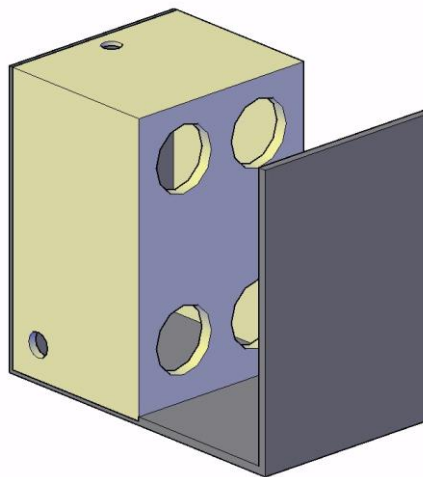


Imagen 29. Herradura montada (producción propia).

Sensor de ultrasonidos [12] (Imagen 30):

- Aparato electrónico que se encarga de la detección por proximidad de objetos y que es controlado a través de un microcontrolador.
- Señales de entrada: alimentación de 3V a 5V y señal de trigger (3.3V) de 10 microsegundos como mínimo enviado por el microcontrolador.
- Señales de salida: señal del ECHO (3,3V) de la señal TRIGGER y la conexión a tierra.
- Control de calidad: por cada lote de 100 unidades se seleccionarán 4 al azar, donde, además de localizar se tendrá que realizar unas pruebas a través de otro

microcontrolador. Este le enviará la señal de 20 microsegundos que inicia la detección del sensor y se comprobará que detecte correctamente las distancias.

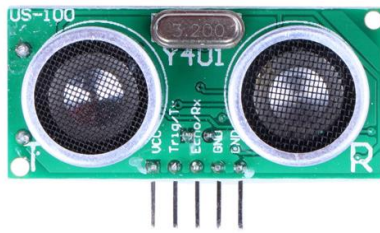


Imagen 30. Sensor US100 (tomada de <https://www.microjpm.com/products/ad38318/>).

Tarjeta mini MSP432P401R [13] (Imagen 31):

- Placa base con el microcontrolador MSP432 integrado con el que se programan los temporizadores que contiene para controlar un sensor.
- Señales de entrada: alimentación de 3,3V y las señales del eco de los sensores.
- Señales de salida: señal trigger que activa los sensores y las conexiones a tierra.
- Con 4 temporizadores timerA disponibles para trabajar.
- Control de calidad: se seleccionarán 4 unidades por cada lote de 100, donde se conectará a un ordenador que ejecutará el programa para comprobar el correcto funcionamiento de los temporizadores.



Imagen 31. Mini MSP432P401R (tomada de <https://www.mouser.es/ProductDetail/Mikroe/MIKROE-2682?qs=HXFqYaX1Q2y9y4y0waIAMw%3D%3D>).

Regulador de tensión [14] (Imagen 32):

- Dispositivo electrónico que transforma los 12V de suministro a 3,3V.
- Permite una tensión máxima de entrada de 15V.
- Tiene tres conexiones: entrada de energía, salida de energía reducida y tierra.
- Aporta tensión de salida fija en 3,3V.

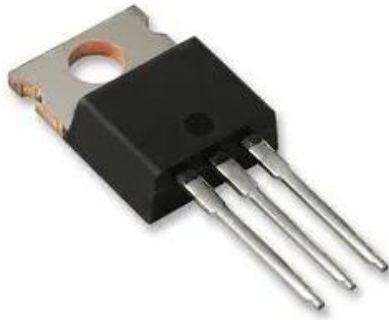


Imagen 32. LM1117T3V3 (tomada de https://es.farnell.com/stmicroelectronics/ld1117v33/req-tensi-ndo-3-3v-1117-to-220/dp/9755837?CMP=KNC-GES-GEN-KWL-PDP-Low-Aov-stmicroelectronics-15-July-22&mckv=bWzHZcnm_dc|pcrid|604239464371|&qclid=CjwKCAjw2rmWBhB4EiwAiJ0mtSJj0jYVOKztCM4mixS9-7Qrb7frouhIn6lZKnamGw-ojAKzCc9l0BoCiQIQAvD_BwE).

Condensador Tantalio [15] (Imagen 33):

- Condensador de tantalio que mantiene la línea de suministro eléctrico de corriente continua sin componente reactivo.
- Capacitancia de $10\mu\text{F}$.



Imagen 33. T491C106M010AT (tomada de https://es.farnell.com/kemet/t491c106k016at/conden-10-f-16v-10/dp/1457461?CMP=KNC-GES-GEN-KWL-PDP-High-Aov-kemet-15-July-22&mckv=dc|pcrid|603873203975|&qclid=CjwKCAjw2rmWBhB4EiwAiJ0mtXoJqck42WPong5jDHLX3hhWXQTmlZ2Q2pO8-PGmy401Djq4z2dtqxoCtXYQAvD_BwE).

Conector de panel macho [16] (Imagen 34):

- Elemento de conexión del cableado al dispositivo electrónico.
- Tiene 4 vías de transporte de cable: dos de alimentación y 2 de datos.



Imagen 34. Conector Binder macho (tomada de <https://es.rs-online.com/web/p/conectores-circulares-industriales/0464375>).

Conector de panel hembra [17] (Imagen 35):

- Elemento de conexión del cableado al dispositivo electrónico.
- Tiene 4 vías de transporte de cable: dos de alimentación y 2 de datos.



Imagen 35. Conector Binder hembra (tomada de <https://es.rs-online.com/web/p/conectores-circulares-industriales/0464460>).

Tornillo M2.5:

- Pieza metálica con una rosca de 2,5 milímetros de diámetro que unirá la herradura con un brazo que sobresale del transportador de palets.
- Control de calidad: comprobar 4 tornillos por cada lote de 100, revisando que tanto el dibujo como la rosca estén en buenas condiciones, así como también posibles deformaciones.

Tornillo M1:

- Misma pieza que la anterior solo que con un diámetro inferior, 1 milímetro, y que unirá el sensor a la herradura.
- Control de calidad: mismo proceso que la pieza anterior.

Cableado [18]:

- Alambrado de cobre de 0,5 milímetros con aislante PVC que interconecta eléctricamente los elementos hardware del sistema embebido.
- Control de calidad: se comprobará que el rollo del cableado esté en buen estado, o que el aislante que rodea el alambrado no tenga grietas u otros desperfectos.

10.3. Procesos

Programación:

- Se ha redactado un código en lenguaje C con el fin de controlar el microcontrolador MSP432 y poner en marcha el sensor de ultrasonidos.
- Control de calidad: una vez terminado el programa, se instala en la memoria flash del microcontrolador y se pone a prueba utilizando los sensores.

10.4. Acabados

- Montaje e instalación: junto con la compra del producto, si al cliente le interesa tiene la opción de contratar por la instalación y acoplamiento de la herradura en las máquinas.

11. Planos y esquemas

Lo primero que mostraremos serán los esquemas unifilares del sistema embebido tanto del utilizado para el desarrollo, como del que se utilizaría en una instalación de 12V de tensión de suministro.

Del esquema referido al desarrollo hecho, la conexión USB sigue el siguiente diagrama de bloques (Imagen 36), y que está basado en la representación de la ficha técnica de la tarjeta de desarrollo ([2] Pág. 8).

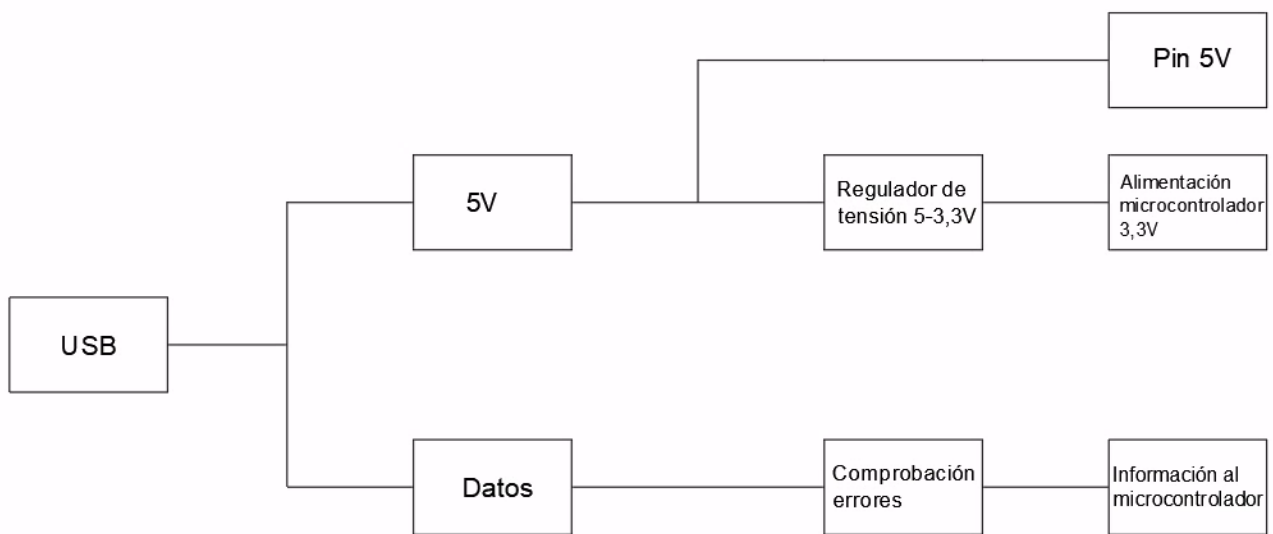
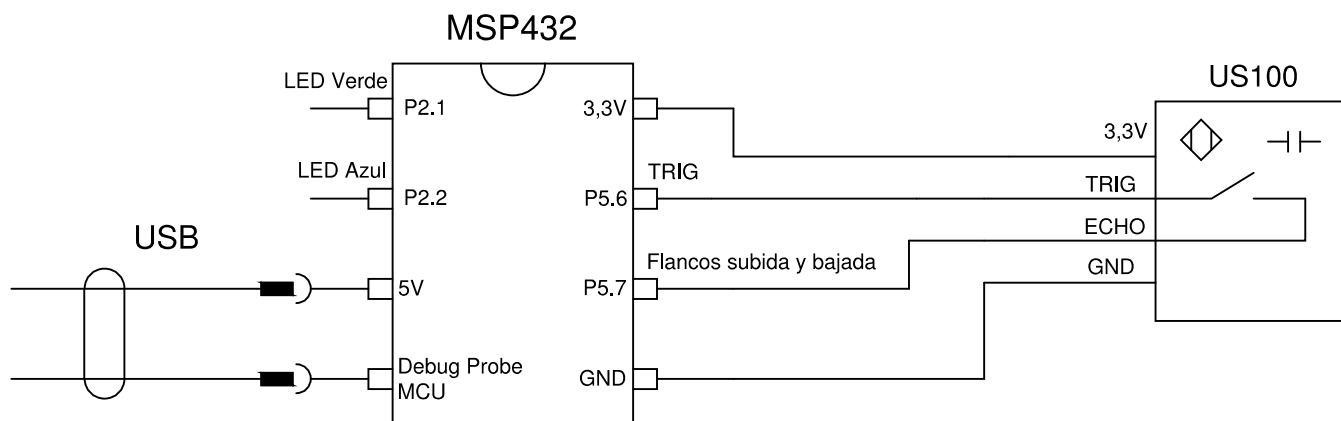
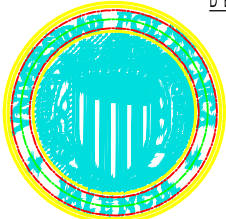
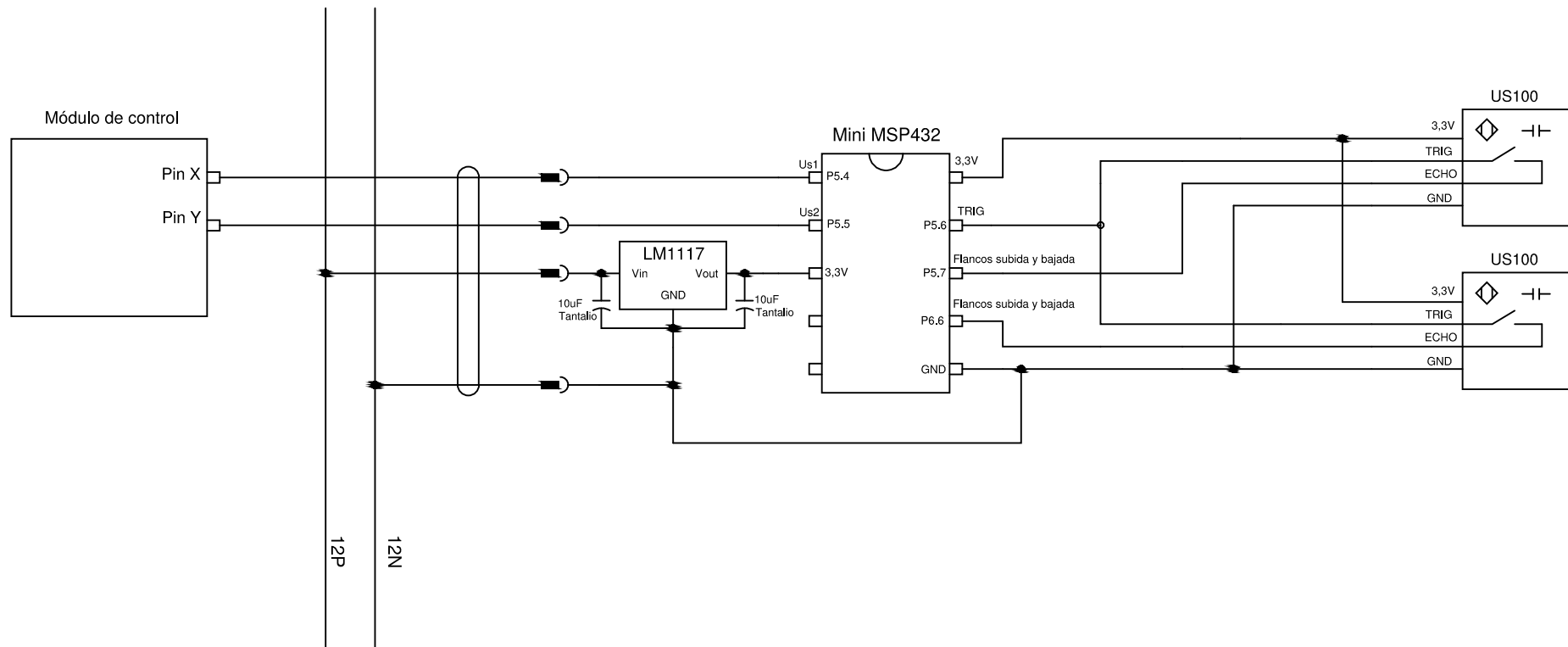
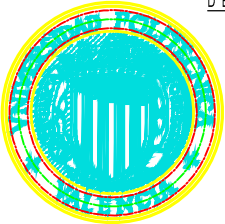


Imagen 36. Cable USB (producción propia).

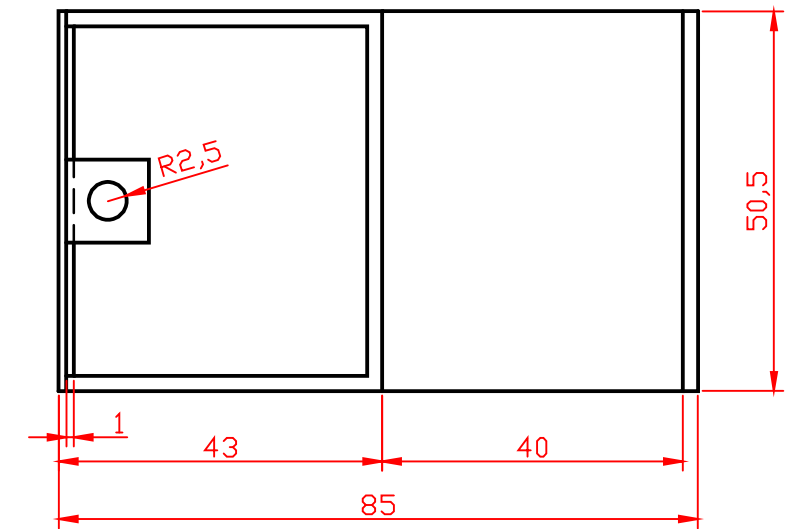
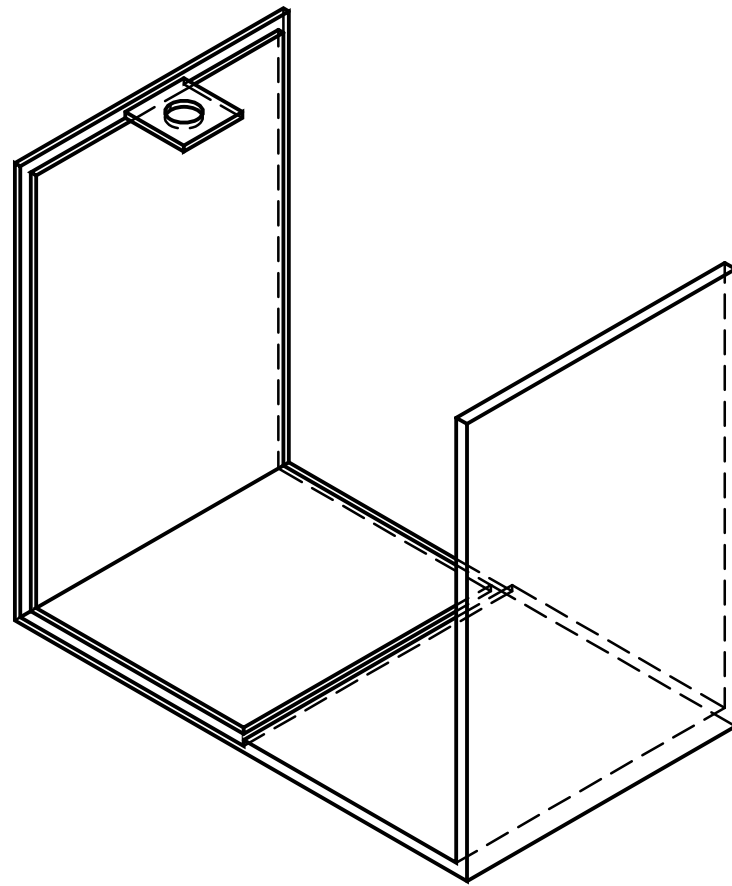
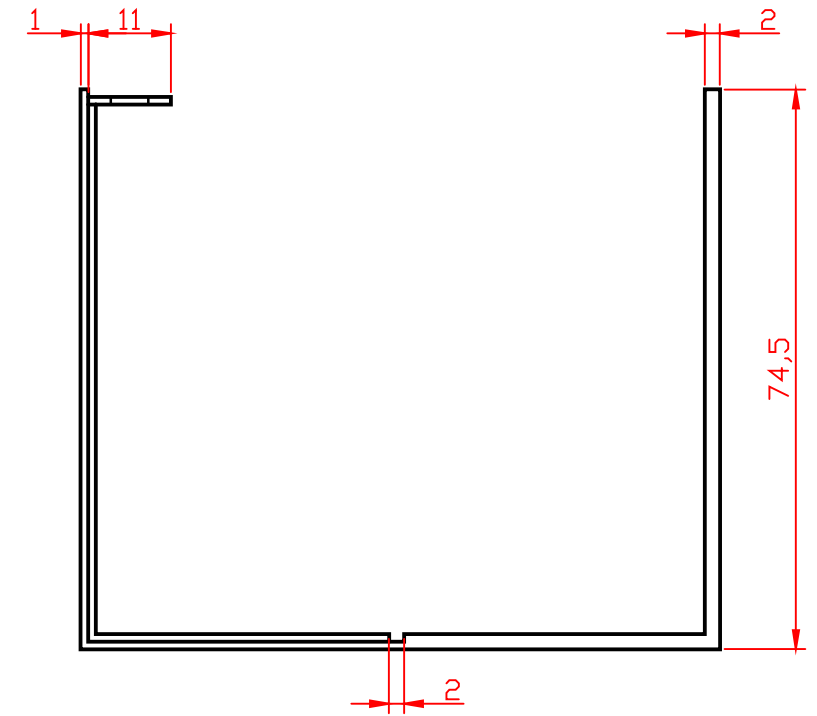
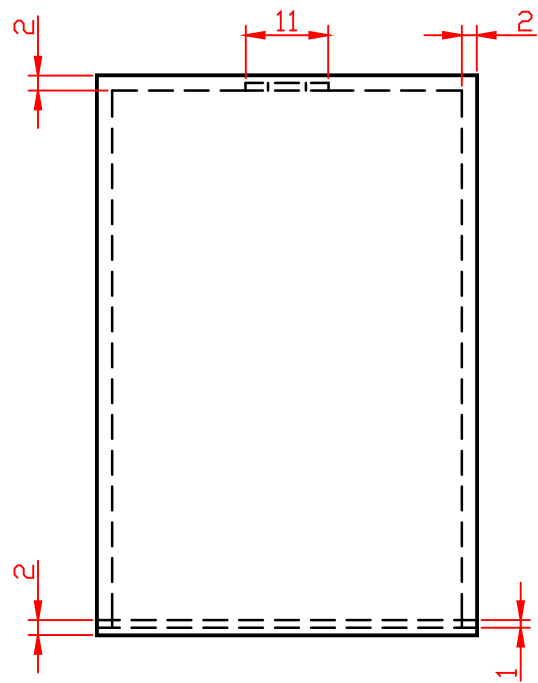


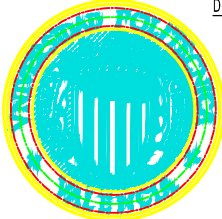
| | | |
|---|--|------------------------------------|
|  UNIVERSIDAD POLITÉCNICA DE VALENCIA | 4º Curso Ingeniería Eléctrica | Alumno: Morales Zarco, Cristian |
| | Desarrollo de un sistema embebido de bajo coste para sensor de herradura | Fecha: 07/2022 |
| | Plano unifilar sensor de herradura ultrasonido (Esquema estudio) | nº plano: 1 |



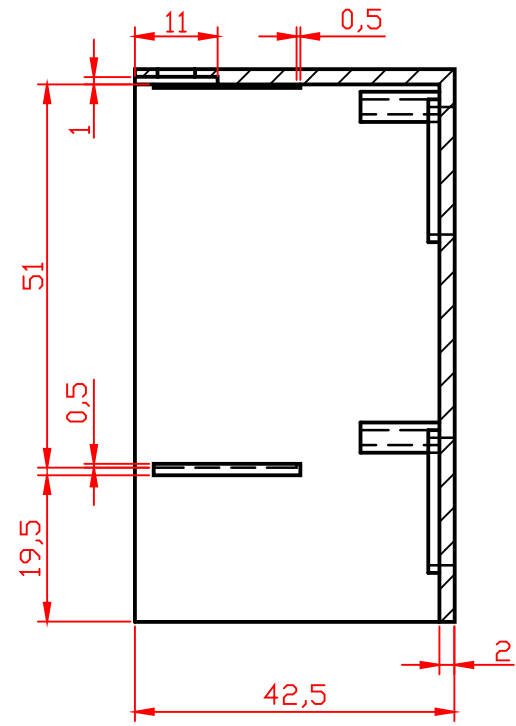
| | | | |
|--|---|--|------------------------------------|
|  | UNIVERSIDAD POLITÉCNICA DE VALENCIA | 4º Curso Ingeniería Eléctrica | Alumno: Morales Zarco, Cristian |
| | | Desarrollo de un sistema embebido de bajo coste para sensor de herradura | Fecha: 07/2022 |
| | | Plano unifilar sensor de herradura ultrasonido (Esquema instalación) | nº plano: 2 |

Por otro lado, se mostrarán los modelos 3D junto con sus vistas acotadas de la herradura que se utilizaría. Además de las pletinas que se ofrecería como extra en la compra del producto.

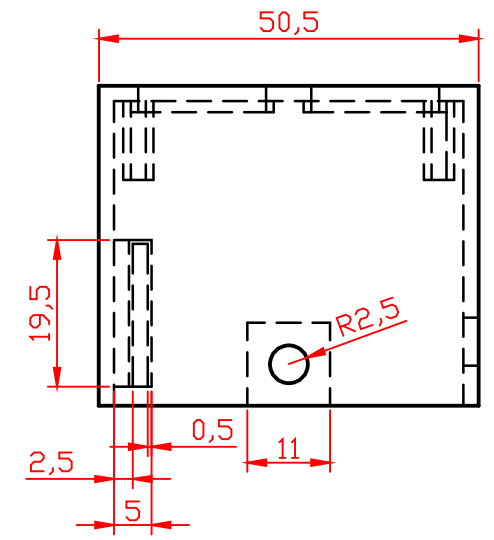
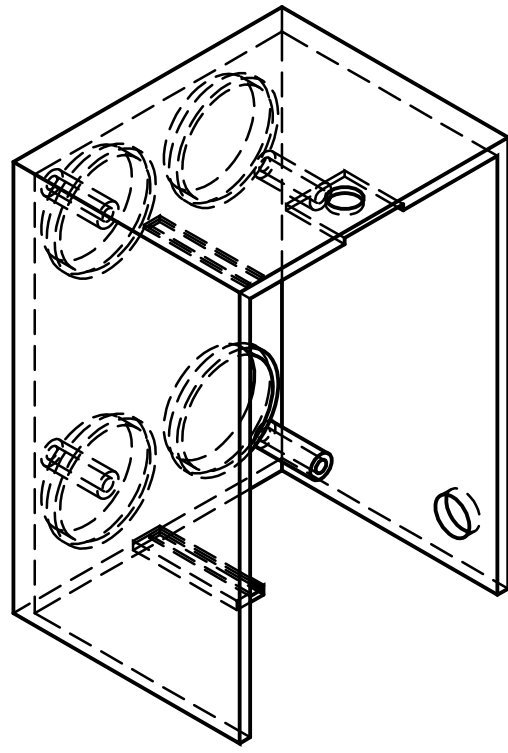
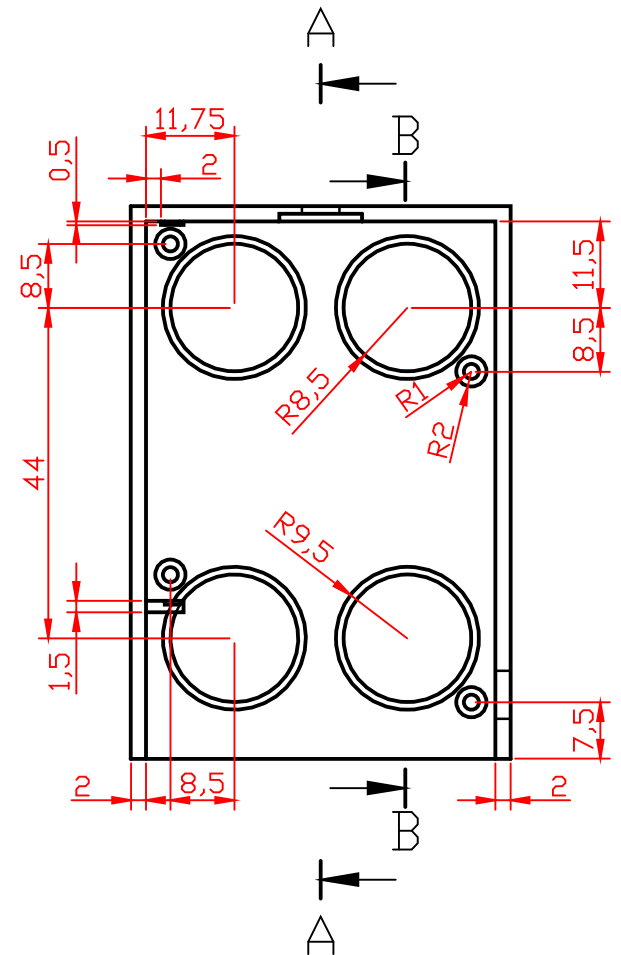
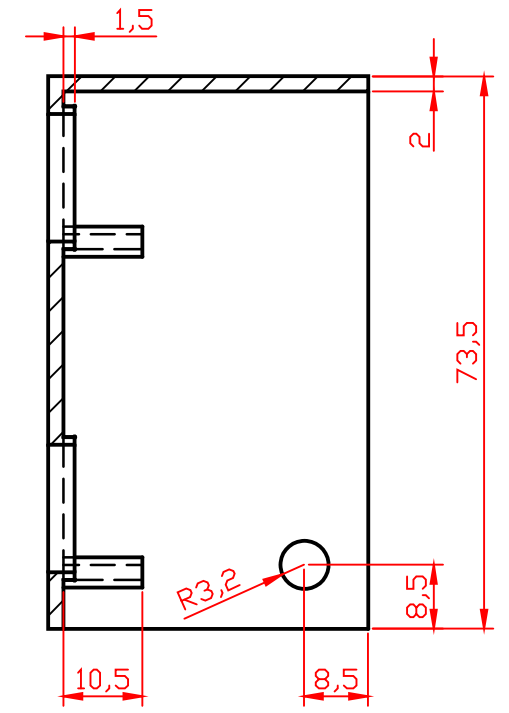


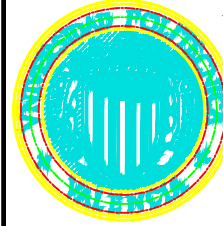
| | | | |
|---|---|--|------------------------------------|
|  | UNIVERSIDAD POLITÉCNICA DE VALENCIA | 4º Curso Ingeniería Eléctrica | Alumno: Morales Zarco, Cristian |
| | | Desarrollo de un sistema embebido de bajo coste para sensor de herradura | Fecha: 07/2022 |
| | | Carcasa 1 | Unidades: mm |
| | | | escalas: 1:1 |

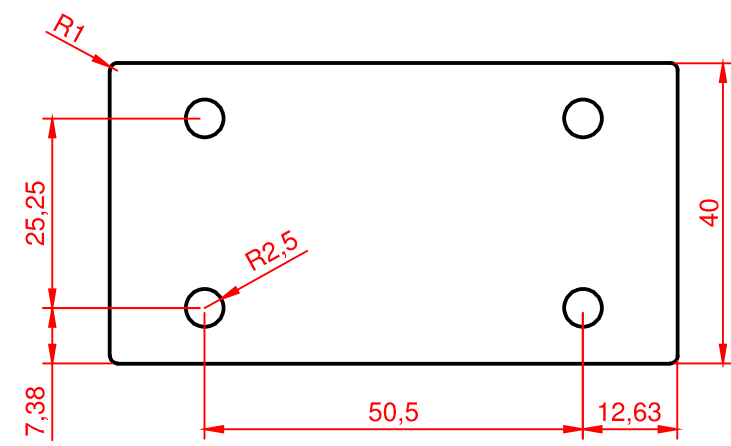
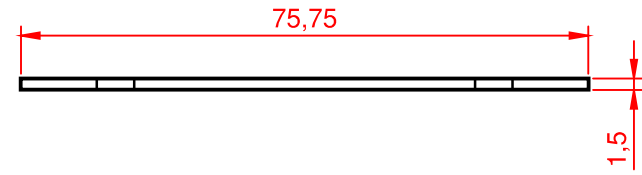
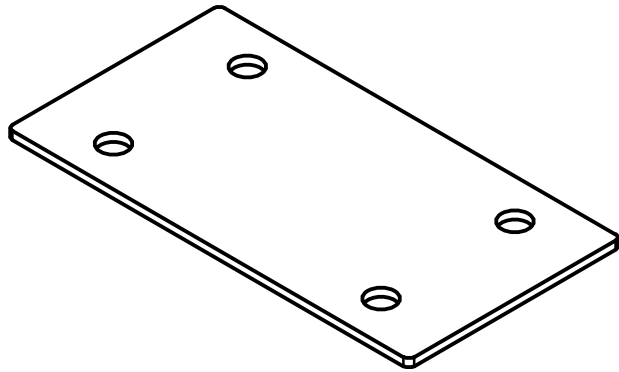
A-A (1:1)

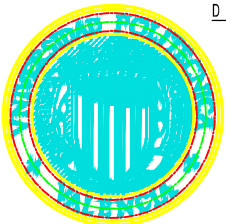


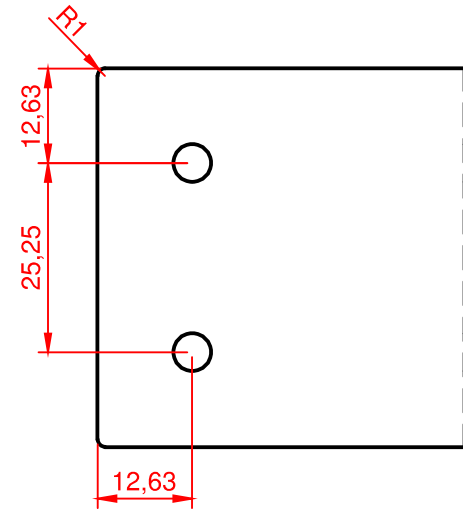
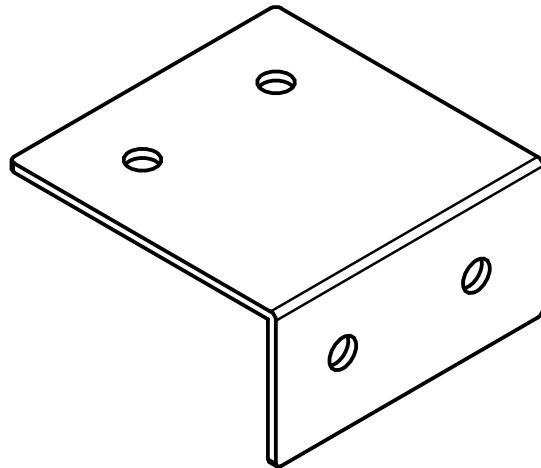
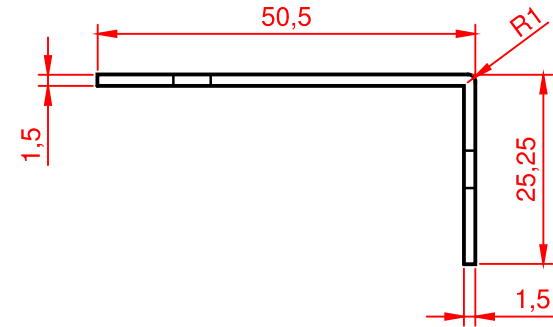
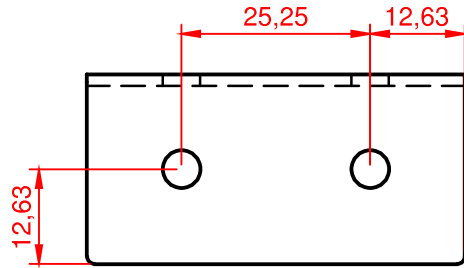
B-B (1:1)

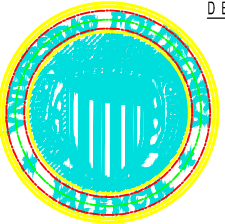


| | | |
|---|---|--|
|  | 4º Curso Ingeniería Eléctrica Desarrollo de un sistema embebido de bajo coste para sensor de herradura | Alumno: Morales Zarco, Cristian |
| | Carcasa 2 | Fecha: 07/2022 Unidades: mm escalas: 1:1 |



| | | | |
|--|---|--|------------------------------------|
|  | UNIVERSIDAD POLITÉCNICA DE VALENCIA | 4º Curso Ingeniería Eléctrica | Alumno: Morales Zarco, Cristian |
| | | Desarrollo de un sistema embebido de bajo coste para sensor de herradura | |
| | | Pletina 1 | Fecha: 07/2022 |
| | | | Unidades: mm |
| | | | escalas: 1:1 |



| | | | |
|--|---|---|------------------------------------|
|  | UNIVERSIDAD POLITÉCNICA DE VALENCIA | 4º Curso Ingeniería Eléctrica | Alumno: Morales Zarco, Cristian |
| | | Desarrollo de un sistema embebido de bajo coste para sensor de herradura | |
| | <h2>Pletina 2</h2> | | Fecha: 07/2022 |
| | | | Unidades: mm |
| | | | escalas: 1:1 |

12. Bibliografía

- [1] DIEEC, «Industrial Systems Engineering,» [En línea]. Available: http://www.ieec.uned.es/investigacion/dipseil/pac/archivos/informacion_de_referencia_ise5_3_1.pdf. [Último acceso: 10 5 2022].
- [2] T. Instruments, «MSP432P401R LaunchPad. Development Kit,» [En línea]. Available: <https://www.electronicaplugandplay.com/component/edocman/?task=document.viewdoc&id=41&Itemid=>. [Último acceso: 24 5 2022].
- [3] Keyence, «Métodos de detección y Características de los sensores fotoeléctricos,» [En línea]. Available: <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/photoelectric/info/>. [Último acceso: 9 5 2022].
- [4] I. y. Control, «Módulos de entrada y salida de datos,» [En línea]. Available: <https://instrumentacionycontrol.net/modulos-de-entrada-y-salida-de-datos/>. [Último acceso: 19 5 2022].
- [5] «Detector por ultrasonidos,» [En línea]. Available: https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/IEA/AI/AI04/es_IEA_AI04_Contenidos/website_44_detector_por_ultrasonidos.html. [Último acceso: 9 5 2022].
- [6] K. Walters, «Distance Measurement with Ultrasound,» [En línea]. Available: https://www.mouser.es/datasheet/2/737/distance_measurement_ultrasound_hcsr04-2489278.pdf. [Último acceso: 30 5 2022].
- [7] Texas Instruments, «MSP432P4xx SimpleLink Microcontrollers. Technical Reference Manual,» [En línea]. Available: https://schaumont.dyn.wpi.edu/ece4703b21/_downloads/8bf98313124444641502c686bb90dbaa/msp432p401r-trm.pdf. [Último acceso: 1 6 2022].
- [8] Mikro Elektronika, «Mini-M4 for MSP432 schematic,» [En línea]. Available: <https://download.mikroe.com/documents/starter-boards/mini/msp432/mini-m4-msp432-schematic-v100-a.pdf>. [Último acceso: 25 5 2022].
- [9] Texas Instruments, «MSP432P401x Mixed-Signed Microcontollers,» [En línea]. Available: <https://datasheet.octopart.com/MSP-EXP432P401R-Texas-Instruments-datasheet-66100013.pdf>. [Último acceso: 30 5 2022].
- [10] Teknka4, «Aluminio 5052,» [En línea]. Available: <https://www.teknika4.com/es/aluminio-5052>. [Último acceso: 14 6 2022].

- [11] Boletín Oficial del Estado, «Ministerio de trabajo y economía social,» [En línea]. Available: <https://www.boe.es/boe/dias/2021/03/22/pdfs/BOE-A-2021-4477.pdf>. [Último acceso: 5 7 2022].
- [12] Adafruit, «US-100 Ultrasonic Distance Sensor - 3V or 5V Logic,» [En línea]. Available: <https://www.adafruit.com/product/4019>. [Último acceso: 10 5 2022].
- [13] Mikroe, «Mini-MSP432,» [En línea]. Available: <https://www.mikroe.com/mini-msp432>. [Último acceso: 1 6 2022].
- [14] Texas Instruments, «LM1117 800-mA, Low-Dropout Linear Regulator,» [En línea]. Available: <https://www.ti.com/lit/ds/symlink/lm1117.pdf>. [Último acceso: 29 6 2022].
- [15] KEMET, «T491C106M010AT,» [En línea]. Available: <https://api.kemet.com/component-edge/download/specsheet/T491C106M010AT.pdf>. [Último acceso: 26 6 2022].
- [16] RS Components, «conector Binder macho serie 719 de 4 vías macho, montaje aéreo, IP40,» [En línea]. Available: <https://es.rs-online.com/web/p/conectores-circulares-industriales/0464375>. [Último acceso: 26 6 2022].
- [17] RS Components, «Conector Binder hembra serie 719 de 4 vías hembra, Montaje en Panel, IP40,» [En línea]. Available: <https://es.rs-online.com/web/p/conectores-circulares-industriales/0464460>. [Último acceso: 26 6 2022].
- [18] Automation24, «Cable unipolar H05V-K 1x0,5,» [En línea]. Available: https://www.automation24.es/cable-unipolar-azul-oscuro-100-m-lapp-4510141-h05v-k-1x0-5-dbu?previewPriceListId=1&gclid=CjwKCAjw-8qVBhANEiwAfjXLRnvr8gu-nW_rMJlDsBoRxqFTQYXgpmeKQvTtKXupG860w1Wa7z1EHBoCRo4QAvD_BwE. [Último acceso: 26 6 2022].