Universidad Tecnológica Nacional
Facultad Regional Villa María
Ingeniería Electrónica

Adiestrador Canino

Autor/es:
Piva, Micol

Tutor: Ing. en Electrónica José Luis Catalano

Director: Ing. en Electrónica Fabián Marcelo Sensini

Co-Director: Esp. Ing. Electricista-Electrónico Héctor Diego Ferrari

Fecha de Defensa: 29/07/2022

Dedicatorias

A mi familia que me apoyó durante toda la carrera, siempre alentándome y dándome fuerzas para seguir adelante. Especialmente a mis padres que me dieron los valores que tengo hoy como persona y que sin ellos no hubiera logrado muchas cosas de las que hoy estoy orgullosa.

A mis compañeros, amigos que me regaló la universidad, con quienes compartí incontables horas de estudio y de trabajo en proyectos; con quienes aprendí que ayudándonos llegamos más lejos y que fueron un gran soporte para seguir adelante entre todos.

A Shiro, mi perro, quién me enseñó de lo que era capaz y fue el motivo de inspiración para el proyecto.

Agradecimientos

A mi familia nuevamente, por el apoyo incondicional en todo momento.

A mi universidad UTN, que me recibió con los brazos abiertos desde el primer día; a los docentes que no solo nos enseñaban sus respectivas materias con gran dedicación, sino que también estaban dispuestos a resolver cualquier consulta que tuviéramos.

A Germán, adiestrador de profesión, que me asesoró a la hora de realizar el prototipo.

A Franco y Santiago quienes ayudaron en el diseño 3D del dispositivo funcional.

Resumen

El proyecto consiste en un dispositivo a control remoto que libera una pelota o un objeto pequeño, con la finalidad de lograr una mayor eficacia en el entrenamiento de perros de búsqueda y rescate.

El dispositivo en cuestión ha sido conformado por dos compartimentos, cada uno para una de las modalidades disponibles: lanzar o dejar caer. Esta modalidad se determina por un sensor infrarrojo ubicado en sendos compartimentos. Los mecanismos de accionado se realizaron con un servomotor para la modalidad Launcher que lanza la pelota hacia arriba, mediante un resorte, y un motor paso a paso para la modalidad Dropper que suelta la pelota, en el caso de que el dispositivo se colocara a cierta altura.

Todo el sistema de control se realizó con un microcontrolador, el cual se conecta mediante un módulo de Wi-Fi con un celular donde se controla el dispositivo mediante una aplicación móvil.

En dicha interfaz se puede observar el estado de la conexión con el dispositivo y el modo en el que se acciona. También permite la activación del lanzamiento del aparato.

Por otra parte, dentro de la misma interfaz gráfica, se generó un registro del tiempo que tarda el canino en encontrar el dispositivo, para control del usuario.

Palabras claves: launcher, dropper, sensor infrarrojo, ultrasonido, comunicación serial, servidor local, protocolo http, peticiones GET, base de datos, comunicación inalámbrica.

ÍNDICE (Ejemplo)

Título **Página**

Dedicatorias 3
Agradecimientos 4
Resumen 5
Introducción 7
Análisis del problema 7
Análisis de sistemas existentes 7
Descripción de las actividades del proyecto 8
Objetivos 9
Objetivos generales 9
Objetivos particulares 9
Diseño del Proyecto 10
Revisión de requerimientos y parámetros de operación 10
Selección de componentes y dispositivos 10
Diagrama en bloques del dispositivo 14
Diagrama de flujo 16
Desarrollo de la aplicación móvil 19
Desarrollo y Evaluación Final del Sistema 24
Diseño de la placa y desarrollo 24
Presentación final del prototipo 28
Mejoras 33
Conclusiones 34
Anexo I 35
Anexo II 42
Anexo III 43
Anexo IV 45
Bibliografía 47

Introducción

Análisis del problema

El proyecto se ideó con la idea de poder brindar una herramienta que ayude a bomberos, policías y adiestradores caninos a entrenar a sus perros en el área de búsqueda, ya sea de personas o sustancias. El problema que se les presenta en el entrenamiento, es en el momento en que se debe premiar al perro cuando encuentra el objeto de la búsqueda. En el aprendizaje del perro, la precisión en el tiempo de premiación es muy importante para que el perro entienda lo que el entrenador quiere que haga.

En el caso del entrenamiento de los perros de búsqueda, lo que se necesita es lograr que el perro, habiendo encontrado su objetivo, marque la posición donde está el objeto hasta que sea premiado. Es por esto que el premio deberá provenir de este lugar, lo que generará expectativa en el canino y que, por ello, mire el lugar por el que va a salir su premio: el lugar donde está el olor que rastreó.

Hasta hoy, una persona era la encargada de estar lista, junto al objetivo, para premiar en cuanto el perro lo detectara. Pero, con el tiempo, el perro comenzaba a anticiparse y, una vez detectado el objetivo, ponía toda su expectativa en la persona que tenía el premio. Esto hacía que dejara de marcar el objeto de búsqueda.

Con este dispositivo, la persona que lo entrene podrá mejorar los tiempos de premiación, ya que el dispositivo se activa remotamente y podrá lograr que el mantenga su atención en el objetivo. A su vez, el entrenamiento se facilitará al solo necesitar una persona que controle el dispositivo para concluir el objetivo.

Análisis de sistemas existentes

Por un lado, en el mercado en general, se pueden encontrar dispositivos lanzadores de pelotas, que no tienen la finalidad de entrenamiento del canino, sino que simplemente es de entretenimiento, lo cual implica que el lanzamiento sea desmedido y, por ende, no ideal para este tipo de entrenamiento.

Por otro lado, existe en el mercado norteamericano un dispositivo similar al de este proyecto, orientado al entrenamiento de perros de búsqueda, pero se encuentra a un precio inalcanzable.

Los motivos que llevaron a concretar este proyecto fueron la falta de accesibilidad a un dispositivo de estas características y la importancia que conlleva la actividad a la que está destinado el entrenamiento de estos perros. A su vez, se buscó hacerlo lo más económicamente accesible posible, sin dejar de brindar un producto de calidad.

Las cualidades que lo distinguirán de los demás productos del mercado son las siguientes: un automatizado en la detección del modo en el que se dispara, la conexión mediante Wi-Fi y un registro del tiempo en que tarda el perro en encontrar el dispositivo, lo que permitirá tomar decisiones a futuro con respecto al entrenamiento. A su vez, se controla mediante una aplicación móvil con gráfica intuitiva y de fácil uso para cualquier usuario.

Descripción de las actividades del proyecto

Para comenzar con el desarrollo se realizó la programación y prueba de cada componente por separado, siendo estos el módulo de ultrasonido, sensores infrarrojos y manejo de motores.

Posteriormente se implementó el funcionamiento básico del dispositivo mediante el accionamiento de llaves. Una vez logrado esto, se utilizó el módulo de Wi-Fi ESP8266 como punto de acceso para crear una red de Wi-Fi a la que el dispositivo móvil se conectaría. Luego se generó un servidor local con el cual se establecería la comunicación entre el módulo y la aplicación y después, se estableció una comunicación serial entre el módulo y el microcontrolador PIC.

En cuanto al control remoto, se desarrolló una aplicación móvil utilizando la plataforma Xamarin Froms en el entorno de Visual Studio. En dicha aplicación se puede visualizar, por un lado, el estado de conexión del móvil con el dispositivo y el modo en que se hará el lanzamiento. Por otro lado, se encuentran dos botones, uno que habilita el proceso y otro que acciona el lanzamiento. A su vez, se registra en una base de datos local el tiempo que tarda el perro en localizar el dispositivo que se muestra en otra pestaña.

La comunicación entre la aplicación móvil y el módulo ESP8266 se realizó por medio del servidor local creado por el este último, donde el móvil, conectado a la red Wi-Fi creada, utiliza el protocolo http para generar peticiones del tipo GET a la cual el módulo le responde con la información precisada.

Una vez que se logró tener el funcionamiento del dispositivo, se realizó el diseño de la placa con el software de diseño y luego se la implementó en físico.

Para la carcasa, se realizaron los cálculos aproximados del resorte para el lanzamiento vertical de la pelota y, en base a dichas medidas, se diseñó el prototipo y los mecanismos de lanzamiento.

Objetivos

Objetivos generales

- Implementar un dispositivo que lance una pelota u objeto pequeño mediante el control remoto a partir de una aplicación móvil.

Objetivos particulares

- Realizar una programación fiable sin fallas.

- Utilizar un módulo de Wi-Fi robusto y confiable para que se lleve a cabo la conexión con la aplicación móvil de la mejor manera.

- Realizar una aplicación móvil intuitiva.

Diseño del Proyecto

Revisión de requerimientos y parámetros de operación

A la hora de desarrollar el proyecto, se contemplaron las falencias existentes en el entrenamiento de los perros de búsqueda y los problemas que se tenían, para evaluar posibles soluciones.

La principal necesidad era la de tener un dispositivo accionado remotamente para liberarse de inconvenientes en la precisión de la premiación del perro.

A su vez, se requería que esta precisión buscada se realizara con el dispositivo mediante el uso de componentes de rápida respuesta, siempre considerando el factor económico.

Luego, era necesario una buena comunicación entre el dispositivo y el control remoto (en este caso, un celular) para aumentar la exactitud con la que se premia al perro.

Finalmente, se agregó el registro de tiempos para que, a partir de este, el usuario pueda tomar decisiones pertinentes que mejoren el rendimiento.

Selección de componentes y dispositivos

Para la implementación del prototipo, se buscaron los componentes necesarios tratando de lograr un equilibrio entre la accesibilidad económica y el buen rendimiento, así como también el trabajo eficaz

A continuación, se detallan los componentes seleccionados:

Un microprocesador Microchip PIC18F2455 (Fig.1) destinado al control y procesamiento de todas las operaciones realizadas. Se empleó este microprocesador de la familia 18F debido a su tamaño ideal para el proyecto, ya que cuenta con 28 pines de entradas y de salidas, tanto analógicas como digitales; cuenta con dos timers utilizados para el sensor ultrasónico y para el conteo utilizado en el registro de tiempo. Para la comunicación, era necesario que contase con un puerto serial que funcione a 115200 baudios para poder sincronizarse con el módulo de Wi-Fi que viene por defecto con esa velocidad de comunicación.

Fig. 1 - Microcontrolador PIC18F2455

Se utilizaron dos pares de LEDs infrarrojos emisor-receptor (Fig. 2) para realizar la detección de la pelota que se coloque en el compartimento seleccionado, y así, identificar el modo de lanzamiento.

La función de detección se cumple cuando la luz infrarroja emitida por el LED emisor es detectada por el receptor (fototransistor), el cual envía una señal digital alta (1 digital = 5v) al microcontrolador. Cuando se coloca una pelota en alguno de los dos compartimentos, se cortará la conexión entre los LEDs, por lo que el fototransistor enviará una señal digital baja (0 digital = masa). El microcontrolador, al detectar este cambio, anuncia en la aplicación móvil el modo de lanzamiento elegido por el usuario.

Se seleccionaron estos LEDs por ser económicos; prácticos, a la hora de utilizarlos y ensamblarlos; y robustos, cualidad útil dada la naturaleza trasladable del dispositivo.

Fig. 2 - LED infrarrojo emisor y receptor

Para generar el registro de tiempo es necesario saber con certeza cuándo el perro encuentra el dispositivo, por lo que se precisó de un sensor que detecte su acercamiento. Esto se resolvió con un módulo de ultrasonido HC-SR04 (Fig. 3). Si bien es de bajo costo, permite medir distancias de hasta 450 cm de manera confiable y con un margen de error muy pequeño. Solo precisa dos pines de alimentación, y dos pines de control del microcontrolador: uno de ellos se activa por 10 uS para que se envíen 8 pulsos de onda de ultrasonido, y el segundo pin es el que informa el tiempo en que tardó el ultrasonido en ir y venir, según sea el ancho de pulso de la señal de este pin.

Conociendo este tiempo y sabiendo que la velocidad del sonido es de 340 m/s, se puede obtener la distancia que recorrió la onda. A este valor se lo debe dividir por dos, ya que la distancia recorrida por el sonido es de ida y vuelta.

Para el cálculo de la distancia se utilizó la siguiente fórmula:

$$Distancia = \frac{velocidad * tiempo}{2} \qquad (Ecu.\,1)$$

Fig. 3 - Módulo del sensor de ultrasonido HC-SR04

En el caso de los accionamientos, para el mecanismo de lanzamiento vertical, Launcher, se utilizó un servomotor mg996 (Fig.4) de 10 kg. La razón por la que se lo eligió fue porque no se necesitaba mucha precisión para accionarlo, pero sí un buen torque.

En un principio, se había optado por utilizar un motor paso a paso, pero luego, considerando que uno de los objetivos es la accesibilidad económica, y comparando los tipos de motores que podrían satisfacer la necesidad de un buen torque a un costo no muy elevado, se optó por este servomotor. El control lógico se realiza mediante un PWM generado por un pin digital del microcontrolador.

Fig. 4 - Servomotor mg996

Para el mecanismo del modo de soltar la pelota, Dropper, se utilizó un motor paso a paso 28BYJ-48 (Fig.5), ya que tiene precisión en el movimiento, necesario para la parte mecánica realizada. En cuanto a su control, se utilizó un módulo driver de motores paso a paso al cual se le conectan 4 pines del microcontrolador, activando las bobinas del motor para poder generar el movimiento requerido.

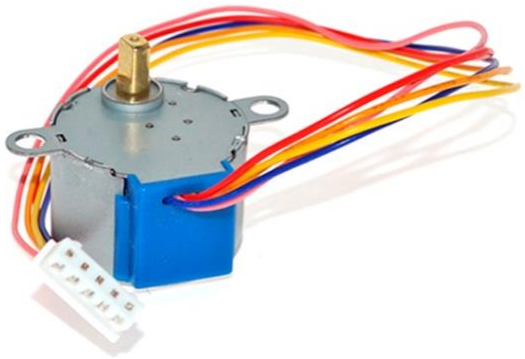


Fig. 5 - Motor paso a paso 28BYJ-48

Debido a que el dispositivo se utilizará en un lugar remoto donde puede que no haya electricidad o señal telefónica, se optó por utilizar una comunicación con un dispositivo celular mediante una red de Wi-Fi generada por un módulo de Wi-Fi. La razón por la que se eligió este tipo de comunicación fue por el gran alcance que tiene y, a su vez, por la posibilidad, como mejora a futuro, de agregarle más de un dispositivo controlado por un mismo móvil, a través de la aplicación.

Para dicha conexión Wi-Fi, se escogió el módulo ESP8266. Este módulo es uno de los más utilizados en el mercado y da la posibilidad de utilizarlo en modo Access-Point, lo cual crea una red de Wi-Fi a la que el usuario se puede conectar con su dispositivo móvil. En el caso del prototipo presentado, se empleó el módulo ESP8266 NodeMCU (Fig.6), pero con las conexiones mínimas que serían utilizadas con el ESP8266-01. Este último es el módulo que se utilizará en el dispositivo final debido a su pequeño tamaño e igual alcance de señal de Wi-Fi.

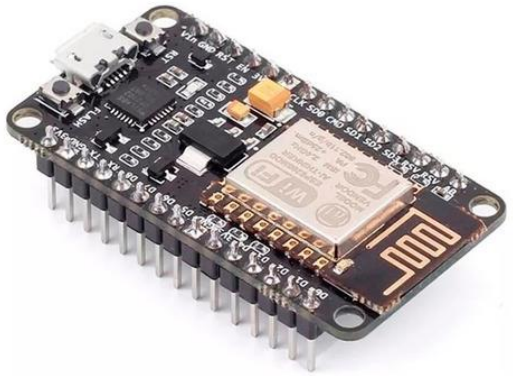


Fig. 6 - Módulo de Wi-Fi ESP8266 NodeMCU

En cuanto a la alimentación del dispositivo, se seleccionó una batería de gel pequeña teniendo en cuenta que el prototipo se utilizará en un lugar probablemente sin fuentes de alimentación cercanas y, por ende, con esta batería se podrá movilizar fácilmente. También, se tuvo en cuenta la potencia necesaria para que todo funcione correctamente y, por esto, se seleccionó una batería de 6V, con una capacidad de 2.4Ah, recargable.

Como la mayoría de los componentes que se utilizan se alimentan con 5 V, se colocó un regularizador LM7805 con salida de 5 V y 1 A y para alimentar el módulo de Wi-Fi con 3,3 V se utilizó el regularizador LM7111.

Debido a que el regulador LM7805 tiene una salida de 5 V si se alimenta con una tensión de entre 7 V-33 V, se utilizó un convertidor boost o step-up para elevar la tensión de la batería a 10 V. De esta manera se obtuvo 5 V para alimentar todo el circuito. Se decidió utilizar este módulo ya que se consideró innecesario colocar una batería de mayor voltaje (12 V) y mayor tamaño y peso.

Diagrama en bloques del dispositivo

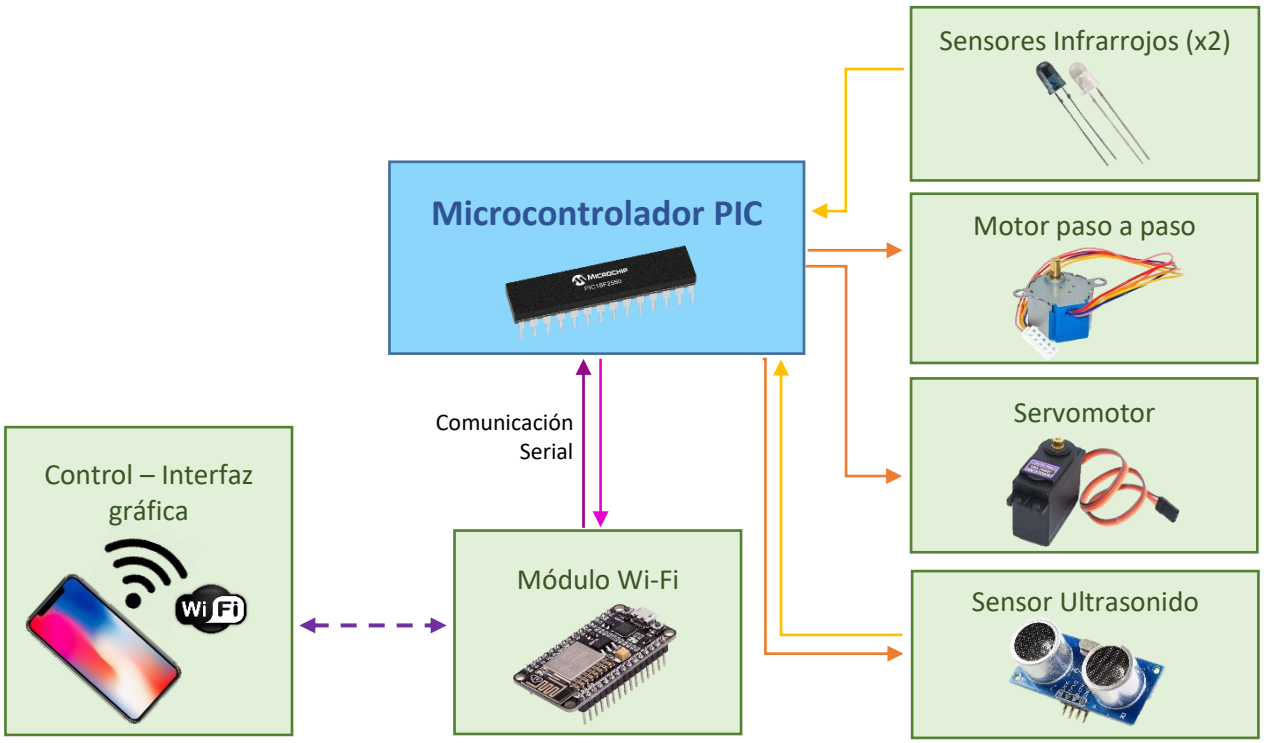


Fig. 7 - Diagrama en bloques del dispositivo

Como se observa en el diagrama en bloques, se dividió el dispositivo en las diferentes etapas y componentes que se utilizan para lograr el objetivo de este proyecto.

Para empezar, se tiene un microcontrolador PIC18f2455, el cual se encarga de controlar todo el sistema y, a su vez, de realizar las operaciones lógicas y aritméticas necesarias, que se irán detallando a medida que se expliquen los demás bloques que componen al sistema.

Otro de los componentes principales que se muestra en el diagrama en bloques es el módulo Wi-Fi ESP8266 NodeMCU. Este módulo es el intermediario entre el microcontrolador y el control remoto creado mediante la aplicación móvil. La comunicación con el microcontrolador es bilateral, a través de un puerto serial y se la puede observar en el

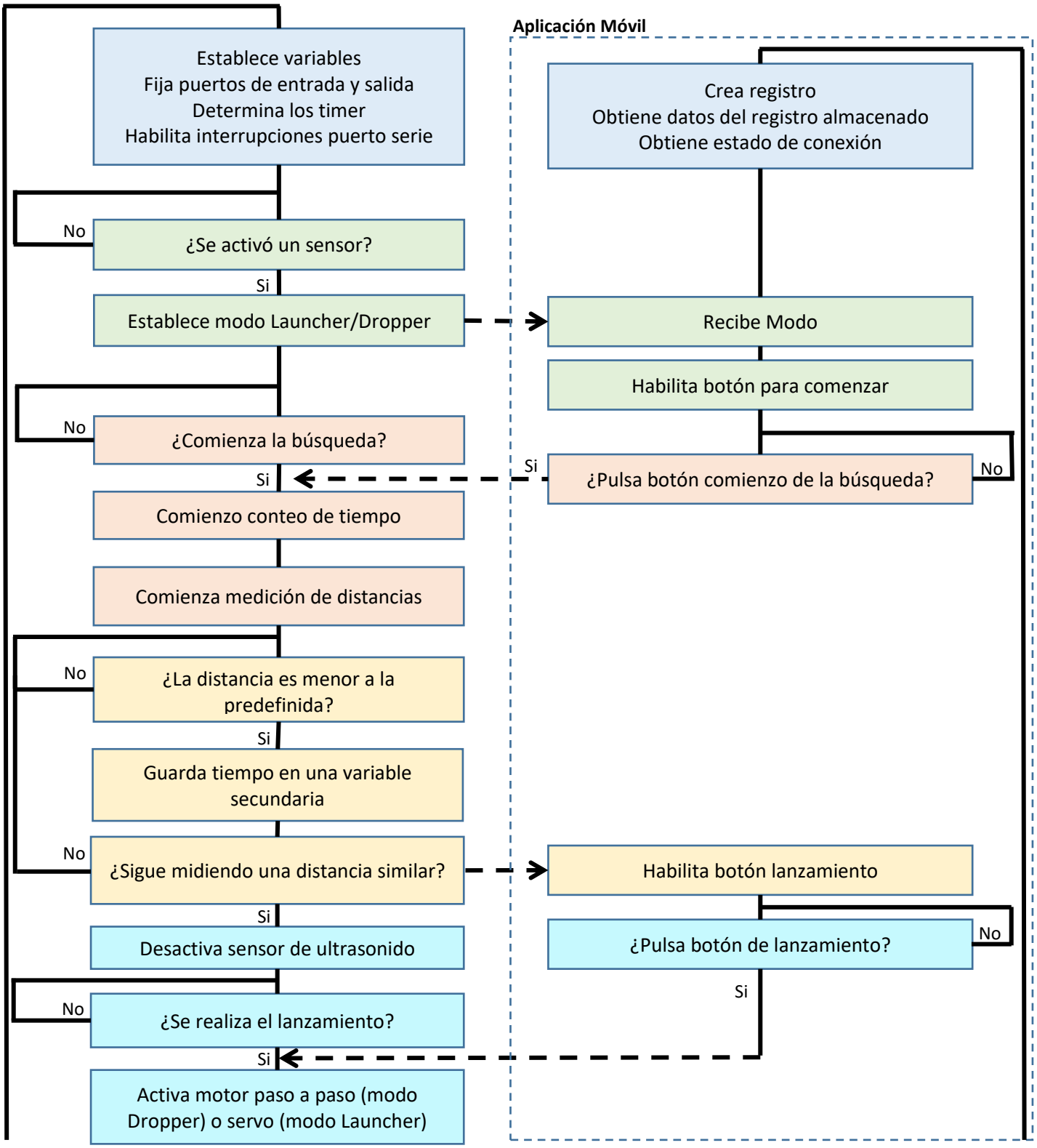
diagrama (Fig.7) con las dos flechas que va desde el módulo al microcontrolador y viceversa. El componente ESP8266 es, entonces, el encargado de interpretar todas las indicaciones brindadas por el microcontrolador y, a su vez, de informarle las decisiones que se toman en la aplicación móvil. Para realizar la comunicación con el móvil, el módulo genera una red de Wi-Fi a la cual el usuario se debe conectar. Luego genera un servidor local, donde la aplicación móvil envía peticiones del tipo GET, utilizando el protocolo HTTP, con la información requerida.

Cuando se coloca una pelota en alguno de los dos compartimentos disponibles para los diferentes tipos de lanzamiento (Launcher/Dropper), el par de LEDs infrarrojos son los encargados de determinar en cuál sección se colocó. Cuando uno de los dos pares de LEDs infrarrojos emisor-receptor son interrumpidos, le manda una señal al microcontrolador que interpreta para luego seleccionar el modo en el que se realizará el lanzamiento. Al mismo tiempo, le informa de esto al usuario a través de la aplicación, por lo que el microcontrolador le envía el dato al módulo de Wi-Fi, a través del puerto serial.

Los lanzamientos son realizados con motores accionados por el microcontrolador bajo la orden dada por el usuario mediante el dispositivo móvil. Para el modo de lanzamiento vertical, se utilizó un servomotor con la fuerza suficiente para poder accionar el mecanismo de resorte implementado. Dicho motor recibe una señal de PWM desde el microcontrolador, de manera tal que se mueva en un ángulo predeterminado. Para el modo donde deja caer la pelota, se utiliza un motor paso a paso pequeño que se mueve una cantidad necesaria de pasos para dejar caer la pelota. Este movimiento se lleva a cabo con el microcontrolador que va encendiendo y apagando paulatinamente las bobinas del motor en la secuencia requerida, para hacerlo girar la distancia previamente fijada.

Por último, el prototipo cuenta con un sensor de ultrasonido HC-SR04 que es activado por el microcontrolador cuando el usuario aprieta el botón “Comenzar la búsqueda” en la aplicación móvil. El sensor está compuesto por un transductor que emite 8 pulsos de ultrasonido y otro transductor receptor que lo detecta, tomando el tiempo que tarda desde que sale hasta que llega. Conociendo la velocidad con la que viaja el sonido en el aire y el tiempo que tardó la onda de ultrasonido en ir y volver, es posible calcular la distancia entre el módulo y el objeto sobre el cual la onda rebotó.

Diagrama de flujo



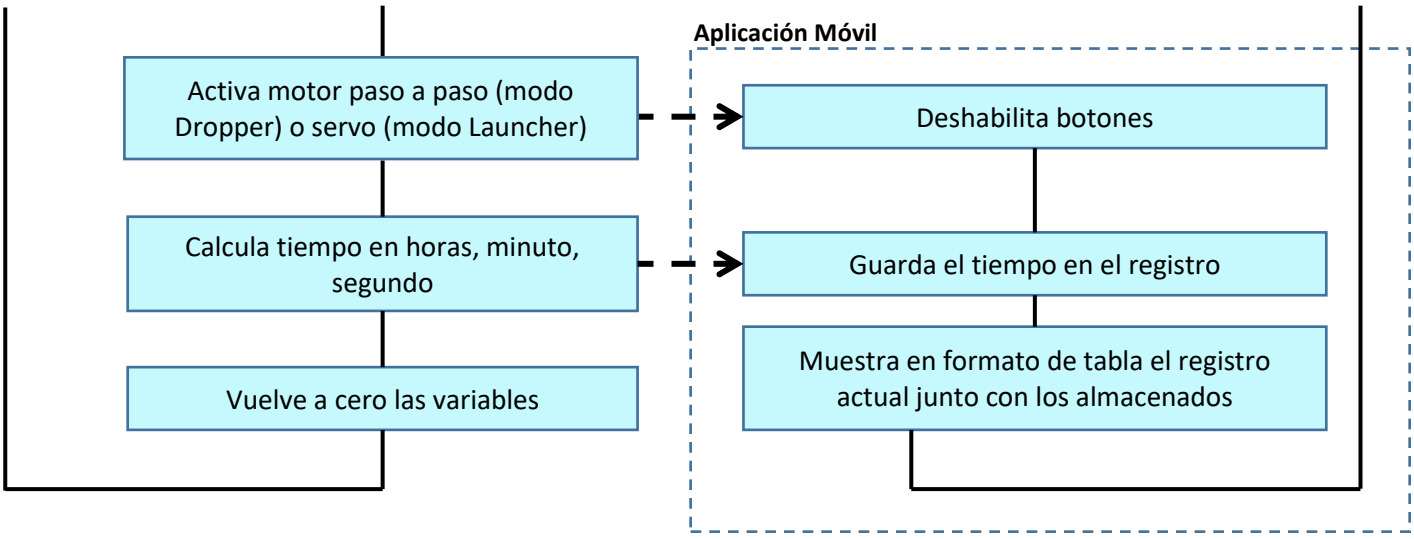


Fig. 8 - Diagrama de flujo de la programación del microcontrolador y la aplicación

En el precedente gráfico, se muestra el diagrama de flujo de la programación, tanto del microcontrolador como de la aplicación web para así, junto con la explicación que se detalla a continuación, poder aclarar las acciones que se llevan a cabo en todo el proceso.

Para empezar, cuando se enciende el dispositivo, lo primero que hace el microcontrolador es determinar los pines y variables, establecer los timers que se utilizarán, además de habilitar las interrupciones del puerto serial, definen los puertos de entrada y salida, e iniciar los motores en la posición requerida.

Una vez todo listo, entra en un bucle infinito en el que verifica constantemente si los sensores infrarrojos fueron activados, es decir, si se colocó alguna pelota en algunos de los compartimentos.

Paralelamente, cuando se enciende el dispositivo, el módulo ESP8266 genera una red de Wi-Fi con ID y contraseña a la cual se conectará el celular. Cuando se logra la conexión, la aplicación móvil informa el estado al usuario, colocando una etiqueta que diga “conectado”. Mientras, genera un registro del tipo SQLite donde almacenará el registro del tiempo. En el caso de que ya existiera ese registro, simplemente carga los datos de dicho archivo.

Cuando se inserta una pelota en uno de los dos compartimentos, ya sea en modo “Launcher” o en modo “Dropper”, se le envía la señal al microcontrolador que, mediante el módulo de Wi-Fi, le comunicará a la aplicación móvil el modo de lanzamiento.

Luego, el microcontrolador queda en espera a que el usuario le avise cuando da por comenzada la búsqueda, es decir, que envía al perro a buscar. Para esto, se debe apretar el botón “Comenzar la búsqueda”. Cuando el microcontrolador recibe esta información, inicia un conteo de tiempo real generado con un timer. Este tiempo indicará cuánto tardó el perro en encontrar el objetivo.

Por otra parte, el sensor de ultrasonido es activado y comienza a medir distancias. Para obtener esta medida, se utilizó el timer1 que realiza el conteo de tiempo transcurrido desde que se habilita la parte del sensor que envía un sonido ultrasónico hasta que rebota en alguna superficie, si la hubiere, y es captada por la otra parte del sensor. Sabiendo este tiempo y teniendo la velocidad con la que viaja el sonido, se utiliza la *Ecu. 1* para el cálculo de la distancia existente entre el dispositivo y la superficie interferente.

Si la distancia medida es igual o menor a la seteada (unos 50 cm), el microcontrolador almacena el valor del tiempo registrado y cuenta unos segundos para verificar que realmente sea el perro el que arribó. Pasados unos segundos, si la distancia medida sigue siendo menor a los 50 cm, entonces se desactiva el contador al igual que el sensor de ultrasonido y se le avisa al usuario del arribo del canino mediante la habilitación del botón de lanzamiento en la aplicación.

En el momento en el que el usuario lo considere oportuno, puede pulsar el botón de lanzamiento, lo que hará que el microcontrolador reciba la orden y active el motor seleccionado para el modo de lanzamiento elegido.

Para el caso del servo motor, el cual se acciona en el modo de lanzamiento Launcher, se creó una señal PWM con un pin digital debido a que, con el cristal de 20 MHz que se utilizó, no se podía obtener el PWM indicado desde el pin propio de PWM. Por lo tanto, conociendo que, para que el motor esté en 0° (posición inicial) en un ciclo de 20 ms se debe generar un pulso de 1 ms, y para moverlo a la posición requerida de 90° se debe generar un pulso de 1,5 ms, tal como se puede ver en la Fig. 9.

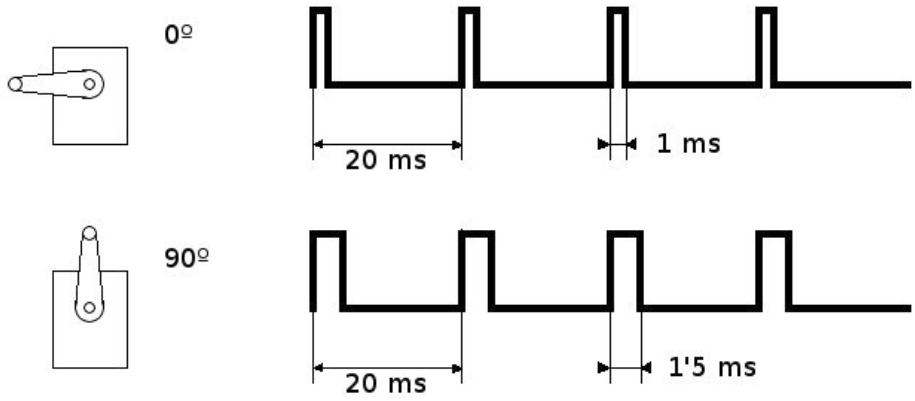


Fig. 9 - Onda de PWM generada para el funcionamiento del servo

Si el lanzamiento fuera en modo Dropper, se activan las 4 salidas digitales alternándolas secuencialmente para que se vayan encendiendo las bobinas del motor paso a paso, de manera que genere el movimiento que se necesita. Esta secuencia se realiza las veces necesarias para dejar espacio a que la pelota caiga, y luego regresa a la posición inicial. En la Fig.10 se puede observar la secuencia y las bobinas que se van activando en el programa del microcontrolador, de manera tal que genere una secuencia de un paso donde se active una sola bobina a la vez.

Secuencia Paso a Paso Unipolar

Paso	A	B	C	D
Paso 1	1	0	0	0
Paso 2	0	1	0	0
Paso 3	0	0	1	0
Paso 4	0	0	0	1

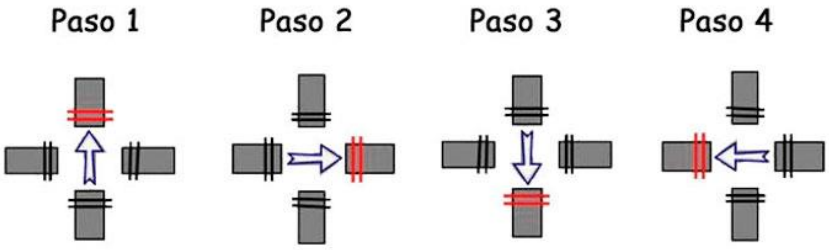


Fig. 10 - Secuencia de activación de las bobinas del motor paso a paso

Finalmente, luego de haber realizado el lanzamiento, la aplicación deshabilita todos los botones y pide el valor del tiempo almacenado al microcontrolador. Por esto, el micro calculará la conversión del tiempo en segundos, minutos y horas según sea necesario y procede a enviarlo a la aplicación, mediante el módulo ESP8266.

Cuando la aplicación móvil haya obtenido el dato, lo almacena en el registro realizado en un principio y muestra un cartel avisando de esta acción, siempre que se haga de manera exitosa. Luego, vuelve a iniciarse ambos programas.

Si el usuario quisiera revisar los valores almacenados, tiene que cambiar de pestaña dentro de la misma aplicación, donde se muestran todos los registros ordenados por fecha en formato de tabla.

Desarrollo de la aplicación móvil

La aplicación se realizó utilizando la plataforma Xamarin Froms en el entorno de Visual Studio. Se decidió utilizar este entorno debido a la familiaridad que se tenía con él, al igual que con lenguaje de programación utilizado (C#). El lenguaje C# es el encargado de realizar toda la parte lógica de la aplicación, mientras que la parte gráfica fue realizada con el lenguaje Xmal.

Al iniciar la aplicación, aparece una pantalla de presentación e inmediatamente después se visualiza la pantalla principal (Fig.11). En esta última, se observa el estado de la conexión con el dispositivo, el modo de lanzamiento y los botones deshabilitados. A su vez, en la parte inferior se observan la pestaña actual y la pestaña del registro de tiempo. Para cambiar de pestaña se debe pulsar el icono del registro o, simplemente, deslizar la pantalla hacia la izquierda.

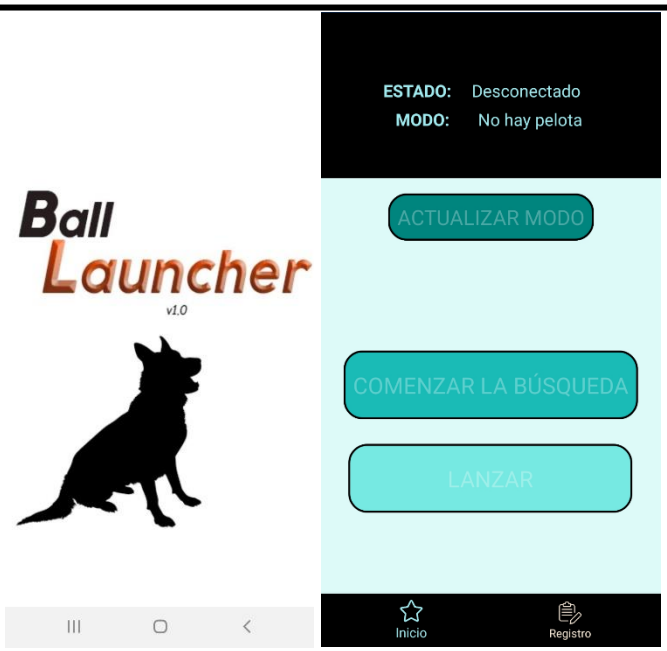


Fig. 11 - Aplicación: vista de presentación y pantalla principal

Una vez que el dispositivo móvil se conecta al Wi-Fi generado por el lanzapelotas, el estado cambia a "Conectado" (Fig.12).

Fig. 12 - Aplicación: estado conectado

La comunicación entre la aplicación y el módulo de Wi-Fi se realiza mediante el protocolo HTTP. Este protocolo permite hacer un intercambio de datos mediante peticiones realizadas a un servidor. En este caso, el módulo ESP8266 crea un servidor local al cual la aplicación envía peticiones del tipo GET, las que responde dicho servidor con la información solicitada.

Cuando el usuario coloca una pelota en uno de los compartimentos, en este caso la del Dropper (Fig.13), el microcontrolador informa al módulo de Wi-Fi y este responde la petición que envió la aplicación, con el modo de lanzamiento seleccionado. Por lo tanto, el usuario es capaz de ver que en la etiqueta de 'Modo' cambia de "No hay pelota" a "Dropper". Al mismo tiempo, se habilitan los botones de "Actualizar modo" y "Comenzar la búsqueda".

Fig. 13 - Aplicación: modo Dropper

El botón "Actualizar modo" tiene la función de enviar una petición al módulo para que actualice el modo en caso de haber cambiado, es decir, si se cambia la pelota de compartimento. Se colocó dicho botón en la interfaz debido a que las peticiones del tipo GET del protocolo HTTP son unilaterales, entonces no era posible actualizar el modo de lanzamiento sin que el celular lo pidiera primero. Como el módulo de Wi-Fi tiene un búfer en el puerto serie, es capaz de guardar el dato del modo de lanzamiento recibido del microcontrolador, evitando así que se pierda la información hasta que la aplicación se la pida.

En el caso de que se cambiara la pelota de compartimento, al pulsar el botón "Actualizar modo", automáticamente la etiqueta cambia al último modo utilizado. Por ejemplo, si el usuario cambiara de modo Dropper a modo Launcher, la etiqueta cambiaría al modo "Launcher" (Fig. 14).

Fig. 14 - Aplicación: cambio modo a launcher

En el caso de que se pulse el botón “Actualizar modo” y que no se haya puesto la pelota nuevamente, la etiqueta vuelve a su estado inicial de “No hay pelota” y los botones se desactivan.

Continuando con el ejemplo en modo Launcher, si se pulsa el botón “Comenzar la búsqueda”, este se desactiva y envía la orden al módulo de que le comunique al microcontrolador que comience el conteo del tiempo y active el sensor de ultrasonido.

Una vez que el perro encuentra el dispositivo y se confirma luego de unos segundos que sigue ahí, la aplicación recibe la respuesta de la petición con esta información. Acto seguido, se habilita el botón “Lanzar” (Fig. 15).

Fig. 15 - Aplicación: se comienza la búsqueda

Cuando el usuario pulsa el botón “Lanzar”, el microcontrolador activa el motor según el modo elegido. Una vez realizado el lanzamiento, el motor vuelve a su posición inicial y el microcontrolador enviará el valor del tiempo medido al módulo. Al recibirlo, la aplicación lo almacena en una base de datos local del tipo SQLite que crea al comenzar a utilizar la aplicación. En caso de haber sido creada previamente, agrega el dato a la lista de valores ya almacenados. En el momento en que logra realizarlo de manera exitosa, lo informa en una ventana de alerta (Fig. 16).

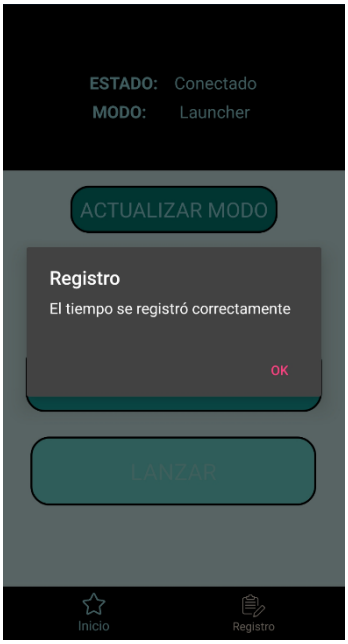


Fig. 16 - Aplicación: advertencia de que se realizó el registro

Para poder ver los datos almacenados, el usuario debe pulsar el icono de registro ubicado en la parte inferior de la pantalla o directamente desplazar la pantalla hacia la izquierda (Fig. 17). Los datos aparecen en una tabla ordenado por fecha, desde la más antigua a la más reciente.

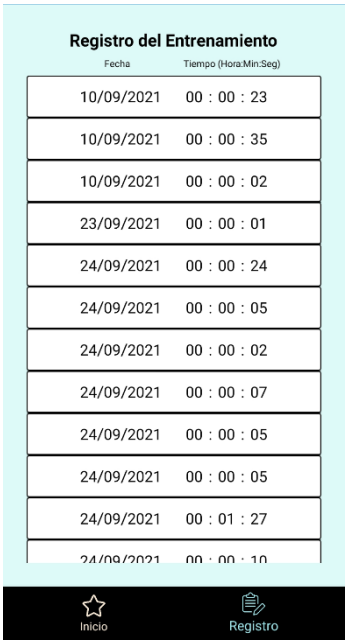


Fig. 17 - Aplicación: segunda pantalla de registro de tiempos de entrenamiento

Desarrollo y Evaluación Final del Sistema

Diseño de la placa y desarrollo

Para el diseño de la placa se utilizó el software de diseño específico para la tarea. Se partió de la idea de realizar una placa de tamaño pequeño, de 6 cm x 9 cm, para que no requiera de mucho espacio en el prototipo y así facilite el ensamblado.

Para la alimentación, se colocó una bornera de dos entradas a la cual le llegan los 6 V de la batería de gel, escogida debido al tamaño de esta última, ya que tiene la potencia ideal para alimentar todos los componentes y ofrece la posibilidad de recarga. Junto a la alimentación, se colocaron dos pines que corresponden a un interruptor de encendido/apagado, ubicado en la parte superior de la carcasa. Inmediatamente luego del interruptor, se colocaron tres capacitores para filtrar la tensión de entrada. Con el fin de informar al usuario si el dispositivo está encendido y si el microcontrolador se ha iniciado de manera correcta, se colocó un LED sobre la carcasa, al lado del interruptor.

Como la mayoría de los componentes se alimentan con 5 V, se utilizó un regulador de tensión (LM7805) para disminuir ambos parámetros (voltaje y amperaje) a los que son necesarios. Para obtener los 5 V de salida del regulador, se lo alimentó con 10 V, resultado de la tensión de la batería elevada con un conversor dc-dc boost o step-up.

Cerca del regulador, se colocaron los pines que van directo a la alimentación de los

motores, tanto del servo como del paso a paso.

En cuanto al microcontrolador, se incorporó un circuito de reseteo que consta de un botón pulsador que lleva al pin de MCLR a masa, tal como se observa en la Fig.18 obtenida del datasheet [1]. Para generar el clock, se utilizó un cristal de 20 MHz conectado a los pines del OSC1 y OSC2 junto con dos capacitores cerámicos.

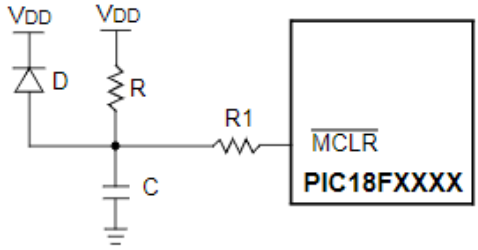


Fig. 18 – Circuito de reseteo del microcontrolador [1]

Se colocó el microcontrolador en un zócalo ubicado sobre la misma placa. Para el resto de los componentes utilizados, se colocaron los pines correspondientes, los cuales están conectados a sus respectivos cables, ya que cada uno tiene una ubicación determinada dentro del prototipo.

En cuanto al módulo de Wi-Fi ESP8266 NodeMCU, se utilizaron los pines para la comunicación serial: Tx y Rx; y dos pines de alimentación: 5 V y masa.

Como el objetivo para el dispositivo final es utilizar el módulo ESP8266-01, en la placa se implementaron los pines hembra necesarios para poder conectar un módulo ESP8266-01, y se reguló la tensión de alimentación de dichos pines a 3.3 V con el regulador LM7111. Para la comunicación serial de este módulo, se realizó un divisor de tensión (Fig.19) para redimensionar la señal que ingresa por el pin Rx a 3.3 V. El cálculo utilizado para el divisor de tensión fue el siguiente:

$$Vsal = Vin * \frac{R12}{R11+R12} \quad \textit{(Ecu. 2)}$$

$$3.3\,V = 5\,V * Rt \Rightarrow Rt = \frac{3.3\,V}{5\,V} = 0.66 \quad \textit{(Ecu. 3)}$$

$$\frac{R12}{R11+R12} = 0.66 \Rightarrow R11 = \frac{R12}{0.66} - R12 \quad \textit{(Ecu. 4)}$$

$$Si\ R12 = 2.2k\Omega \quad \textit{(Ecu. 5)}$$

$$\Rightarrow R11 = 1.13k\Omega \approx 1k\Omega \quad \textit{(Ecu. 6)}$$

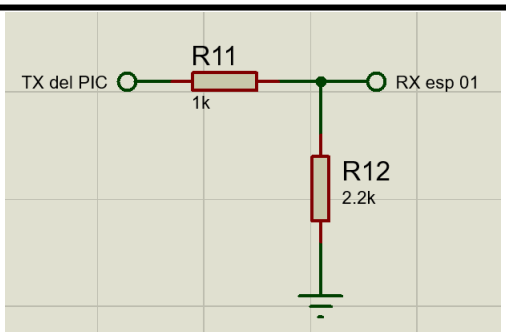


Fig. 19 - Circuito divisor se tensión

A continuación, se puede observar el esquemático completo[20]. En él se encuentran todas las conexiones que se llevaron a cabo ya sean de señales o alimentación. Entre ellos, se incluyeron los pines para programar el PIC, en caso de que fuera necesario. También se incluyeron pines extra de alimentación de 5v y masa.

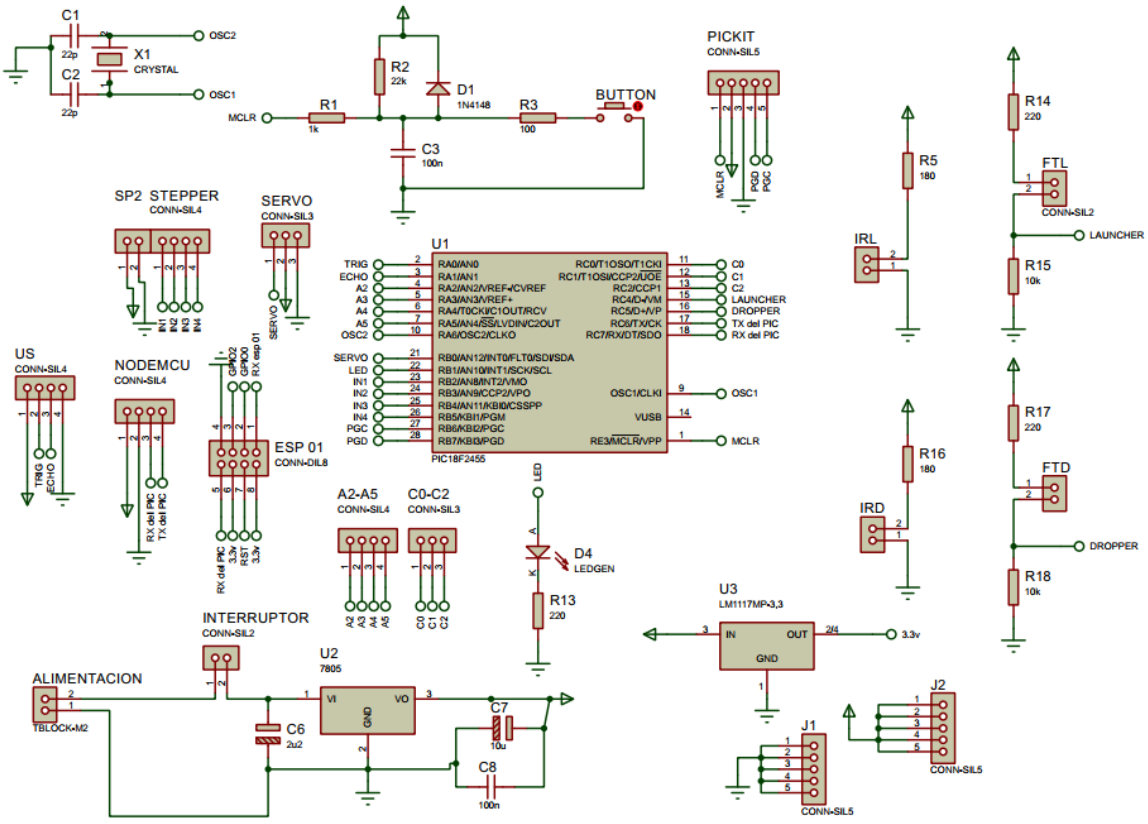


Fig. 20 - Esquemático completo

Una vez concretado el diseño del esquemático, se procedió a realizar el diseño del circuito. Las pistas se realizaron de un ancho que se suele emplear para estos circuitos,

aumentándolo un poco para la parte de potencia en la que se alimentan el motor paso a paso y el servomotor. La colocación de los componentes se hizo de manera tal que se pueda ensamblar cómodamente, con el espacio suficiente para que no se obstruyan entre sí. A su vez, se ubicaron primero las conexiones de potencia más cercanas a la alimentación y, luego, el resto de los elementos.

Mediante el mismo programa, con la opción de visualizar la placa en 3D, se puede observar si la disposición de todos los componentes se adecúa con los requisitos previstos (Fig. 21 y Fig. 22).

Fig. 21 - Vista 3D superior de la placa

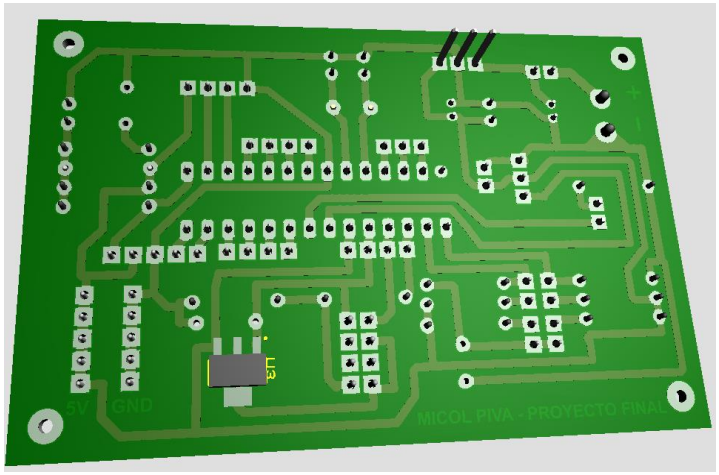


Fig. 22 - Vista 3D inferior de la placa

La placa que se utilizó fue una de pertinax. Se le imprimió el diseño para proteger el cobre que conforma las pistas y, luego, se elimina el cobre restante con ácido para obtener el circuito. Posteriormente, se perforó y se procedió a ensamblar todos los componentes y a soldarlos. Una vez finalizado, se le roció flux para evitar la oxidación de las pistas.

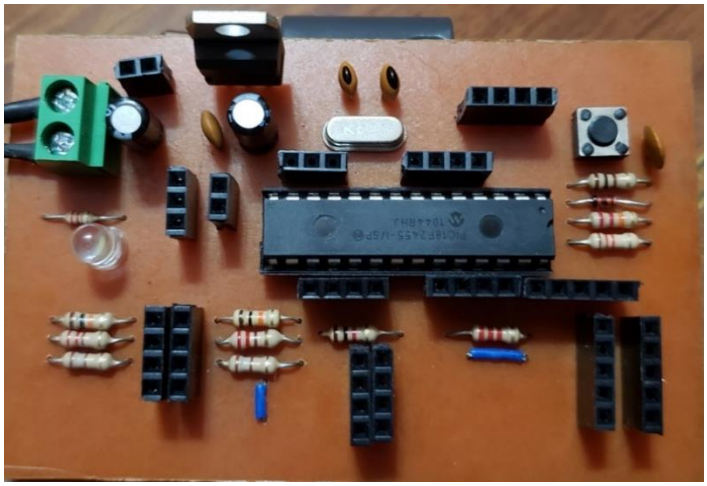


Fig. 23 – Visión superior de la placa

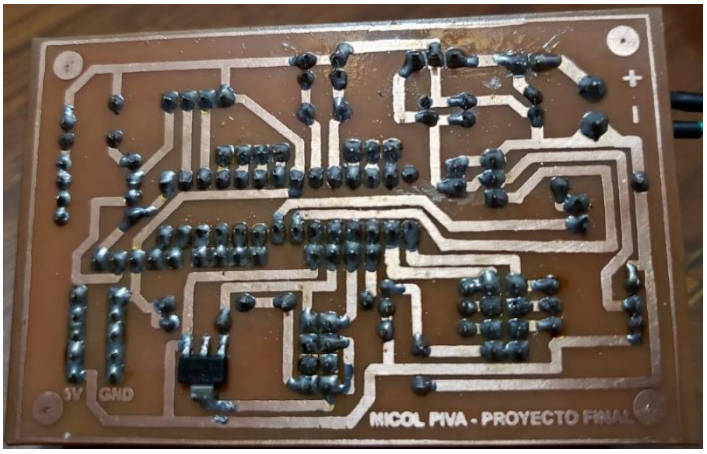


Fig. 24 – Visión inferior de la placa

Presentación final del prototipo

Para mostrar el funcionamiento que se pretende, se prosiguió a realizar un diseño de un prototipo en 2D con el software de diseño.

Previo al diseño final, y con ayuda de un software de cálculo [9], se obtuvieron los valores para el resorte que se utiliza en el modo de lanzamiento Launcher. Para esto, se midió el peso y el diámetro de la pelota de tenis que se toma como referencia: Masa=60 g; Ø=65 mm.

Luego se prosiguió a calcular la fuerza necesaria para empujar dicha pelota:

$$\mathrm{F\,[N]} = m\,[\mathrm{kg}] * 9.81\left[\frac{m}{s^2}\right] = 0.06\,[\mathrm{kg}] * 9.81\left[\frac{m}{s^2}\right] = 0.588\,[\mathrm{N}] \qquad \textit{(Ecu. 7)}$$

$$1\,[\mathrm{N}] = 0.1019\,[\mathrm{kgf}] \Rightarrow 0.588\,[\mathrm{N}] = 0.059959\,[\mathrm{kgf}] \qquad \textit{(Ecu. 9)}$$

El valor obtenido en Ecu.9 es el que se utiliza para ingresar en la calculadora y, así, obtener algunos valores del resorte que fueron necesarios para tener como referencia a la hora de confeccionarlo. Los valores obtenidos fueron los siguientes:

$$Øexterno = 50\,mm \rightarrow ØActual = 45\,mm \qquad (Ecu.\ 10)$$

$$Øhilo = 3\,mm \rightarrow ØActual = 2.5\,mm \qquad (Ecu.\ 11)$$

$$Aproximadamente\ 15\ espiras$$

A partir de esto, se obtuvo el largo del resorte de 25 cm, número que nos dará la altura del prototipo.

Ya con estos datos se pudo comenzar el diseño en un software de diseño. La idea principal era realizar una carcasa cuadrada que contuviera dos cilindros, que serían los compartimentos de cada uno de los modos de lanzamiento. Por lo tanto, se comenzó por trazar ambas secciones. Posteriormente, se diseñó la pieza móvil que forma parte del mecanismo de lanzamiento en modo Launcher junto con el resorte. A la parte superior del cilindro de esa sección se le realizó un tope para que la parte móvil no salga más allá de los límites de la carcasa y se implementó una pequeña plataforma en la base para mantener centrado al resorte.

Volviendo a la parte móvil, se le realizó una cavidad de menor diámetro que el cilindro que forma parte del mecanismo de accionamiento del motor para lanzar. Esta hendidura cilíndrica permite al cricket trabar el mecanismo cuando se ingresa la pelota y se comprime el resorte hacia la base. El cricket está sostenido por una horquilla y, en un extremo, está sujeto al vástago. Este último mantiene el cricket en su posición gracias a un pequeño resorte puesto a su alrededor. El vástago está sujetado al servo, por lo tanto, el mecanismo funciona de la siguiente manera: el usuario empuja hacia abajo la pelota hasta que la parte móvil se traba. Permanece en dicho lugar hasta que el servo se active y mueva su aleta hacia arriba y con ella eleva el vástago que hace ceder al cricket y con él, el resorte se descomprime.

En la parte del lanzamiento en modo Dropper, se realizó un cilindro con una abertura lo suficientemente ancha para que entre una paleta, controlada por el motor paso a paso. La función de esta paleta es la de sostener la pelota cuando se la coloca dentro del compartimento y cuando el microcontrolador se lo ordene, el motor se accionará para mover la paleta y dejar caer la pelota.

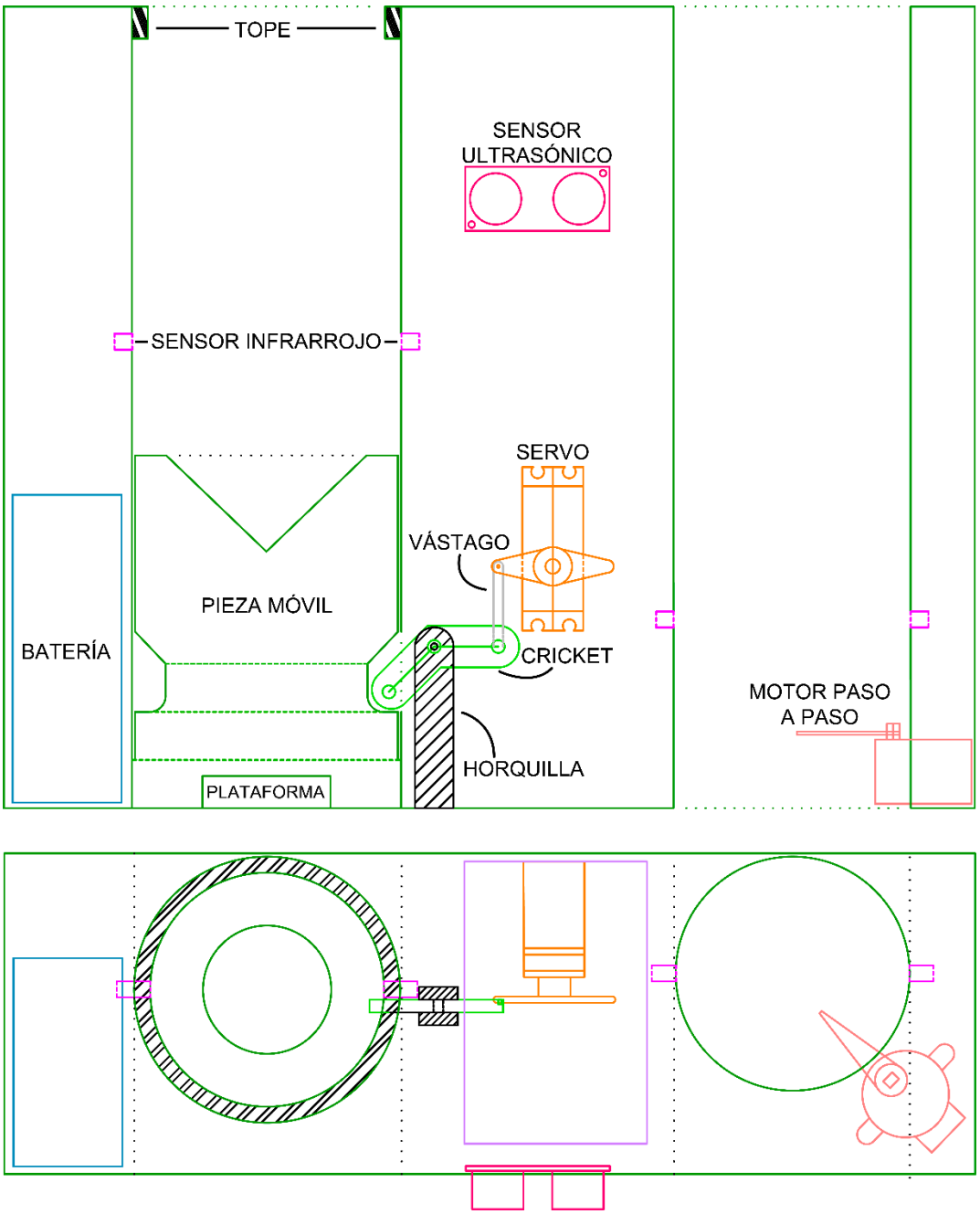


Fig. 25 - Plano del prototipo con vista frontal y superior.

Una vez finalizado el plano con las medidas reales, se prosiguió a realizar el diseño en 3D para imprimir el prototipo con una impresora 3D. Se generó todo el prototipo, por un lado, y la tapa y la base, por el otro, ya que se ajustarían luego de ensamblar los componentes.

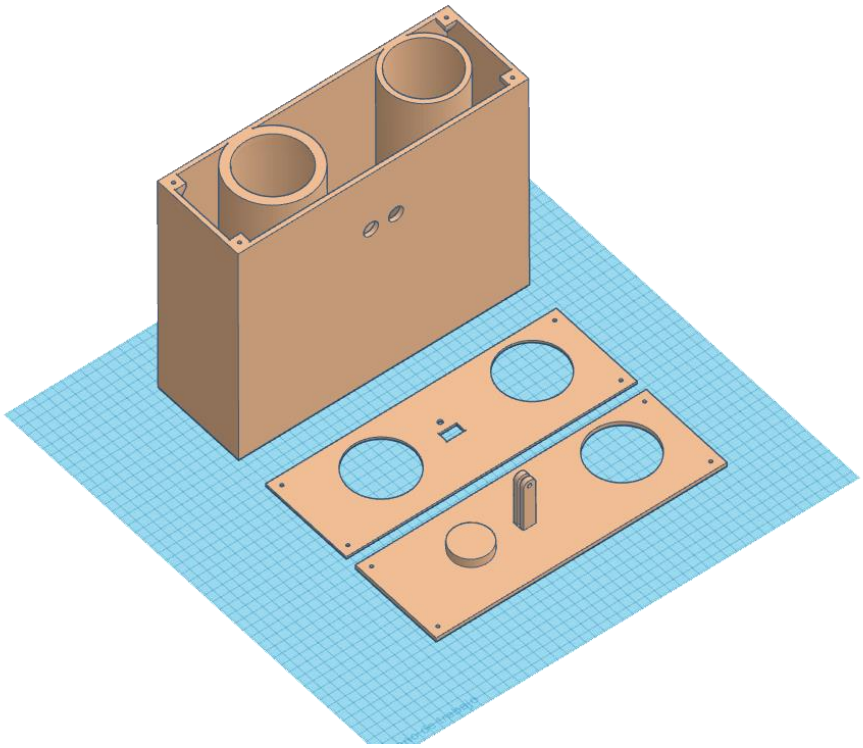


Fig. 26 – Diseño 3D del prototipo.

Paralelamente, se creó la pieza móvil que impulsaría la pelota. Esta cuenta con una cavidad del mismo diámetro del resorte para que sirva de guía, y la parte superior en forma de cono invertido ayuda a que la pelota permanezca centrada.

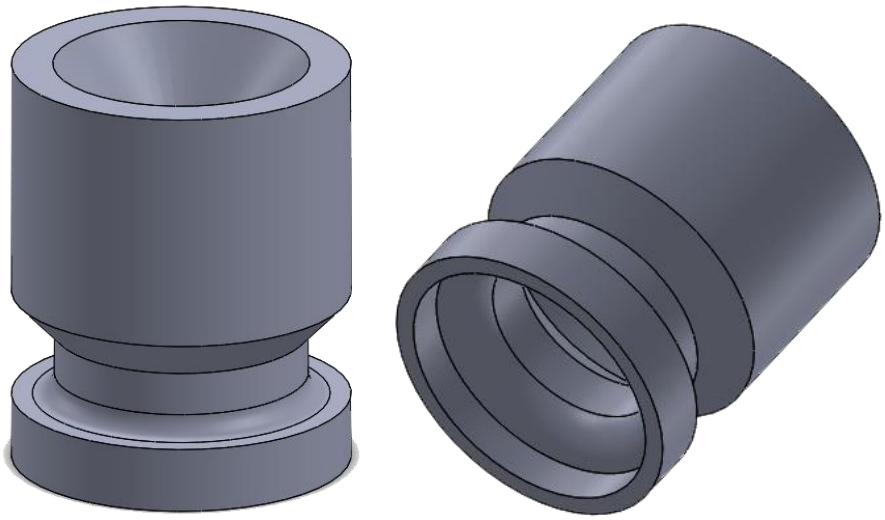


Fig. 27 – Diseño 3D de la pieza móvil

Para el mecanismo, se imprimió el cricket en 3D, mientras que el vástago se realizó de

metal, ya que en él se deposita toda la fuerza del mecanismo.

Terminada las impresiones 3D, se ensamblaron todas las partes y se obtuvo una sola pieza. Se colocaron todos los componentes, la placa y la batería en sus respectivos lugares y se puso a prueba el funcionamiento del prototipo.

Debido a que la horquilla debía soportar mucha presión, se la volvió a realizar de acero. Por otra parte, se hizo un cambio en el largo del cricket ya que no realizaba correctamente el trabado del sistema.

Posterior a estos cambios, se logró obtener el funcionamiento deseado.

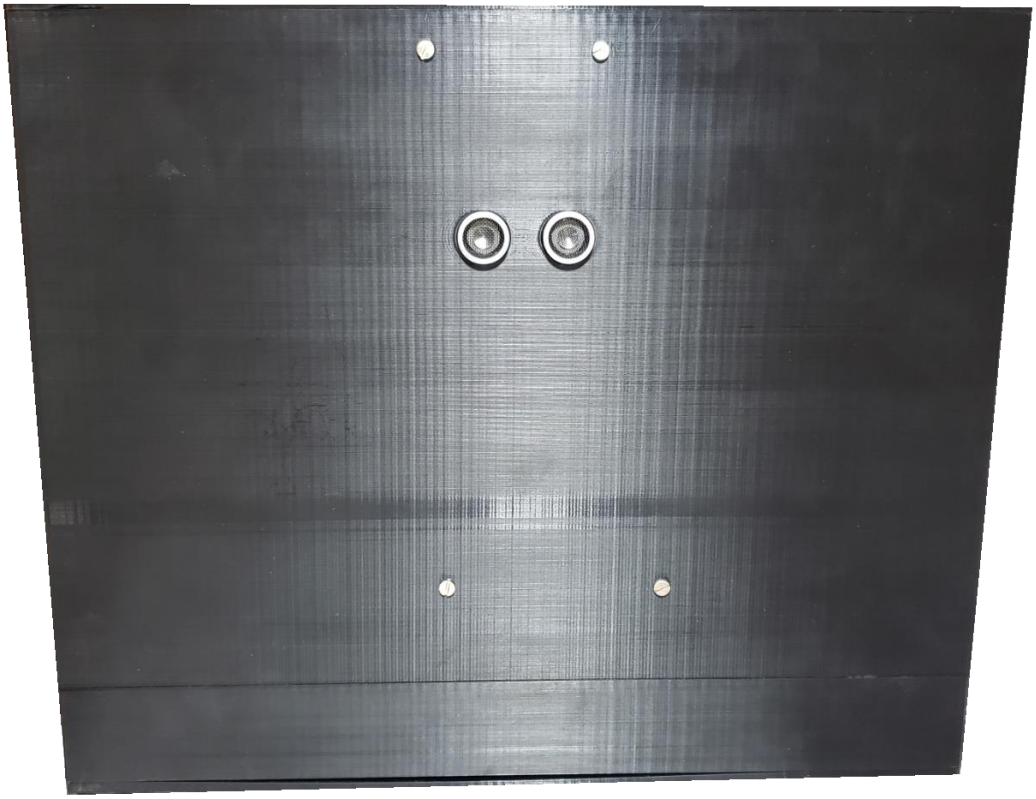


Fig. 28 - Dispositivo final – vista frontal

Fig. 29 - Dispositivo final – vista superior

Mejoras

- Agregar más dispositivos de lanzamiento controlados por una misma aplicación.
- Verificar el estado de carga de la batería y enviarlo al celular.
- Añadir más sensores ultrasonido en las diferentes caras del prototipo para poder ubicarlo en espacios abiertos.
- Brindar la posibilidad de usarlo sin sensor ultrasónico, configurable en la aplicación.

Conclusiones

El proyecto se propuso con la idea de ayudar a bomberos y entrenadores con su trabajo, haciéndolo más práctico. Se buscó brindar un producto efectivo y de calidad, pero a su vez, que fuese económicamente accesible. Debido a esto, se eligieron con cuidado los componentes que lo integran. Se innovó con el software, ya que es algo inexistente en el mercado. Se lo creó para que sea intuitivo y con la idea de que sea una herramienta útil para mejorar el entrenamiento a futuro, mediante el uso del registro.

Se logró todo lo propuesto con los elementos que fue posible conseguir. Proporciona mucha satisfacción haber podido empezar el proyecto desde cero y verlo funcionando como se había previsto.

Si bien se partió con una base fuerte del conocimiento logrado durante el cursado de la carrera, el aprendizaje siguió a lo largo de todo el proyecto requiriendo de más horas de lectura en algunas áreas y otras que se resolvieron de manera más rápida, pero todo fue igualmente gratificante.

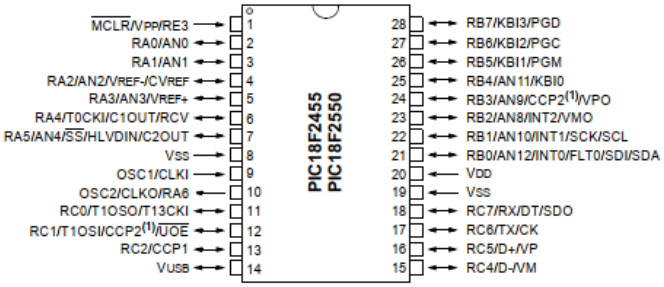
Anexo I

En este anexo se adjuntan las hojas de datos del microcontrolador PIC18f2455 utilizadas a lo largo del desarrollo del proyecto. Estas son: diagrama de pines, especificaciones del oscilador a utilizar, circuito de reinicio del controlador y especificaciones de los timers.

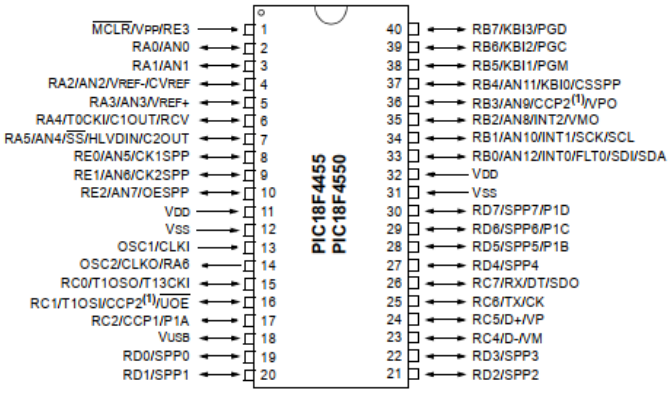
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

2.0 OSCILLATOR CONFIGURATIONS

2.1 Overview

Devices in the PIC18F2455/2550/4455/4550 family incorporate a different oscillator and microcontroller clock system than previous PIC18F devices. The addition of the USB module, with its unique requirements for a stable clock source, make it necessary to provide a separate clock source that is compliant with both USB low-speed and full-speed specifications.

To accommodate these requirements, PIC18F2455/2550/4455/4550 devices include a new clock branch to provide a 48 MHz clock for full-speed USB operation. Since it is driven from the primary clock source, an additional system of prescalers and postscalers has been added to accommodate a wide range of oscillator frequencies. An overview of the oscillator structure is shown in Figure 2-1.

Other oscillator features used in PIC18 enhanced microcontrollers, such as the internal oscillator block and clock switching, remain the same. They are discussed later in this chapter.

2.1.1 OSCILLATOR CONTROL

The operation of the oscillator in PIC18F2455/2550/4455/4550 devices is controlled through two Configuration registers and two control registers. Configuration registers, CONFIG1L and CONFIG1H, select the oscillator mode and USB prescaler/postscaler options. As Configuration bits, these are set when the device is programmed and left in that configuration until the device is reprogrammed.

The OSCCON register (Register 2-2) selects the Active Clock mode; it is primarily used in controlling clock switching in power-managed modes. Its use is discussed in **Section 2.4.1 "Oscillator Control Register"**.

The OSCTUNE register (Register 2-1) is used to trim the INTRC frequency source, as well as select the low-frequency clock source that drives several special features. Its use is described in **Section 2.2.5.2 "OSCTUNE Register"**.

2.2 Oscillator Types

PIC18F2455/2550/4455/4550 devices can be operated in twelve distinct oscillator modes. In contrast with previous PIC18 enhanced microcontrollers, four of these modes involve the use of two oscillator types at once. Users can program the FOSC3:FOSC0 Configuration bits to select one of these modes:

1. XT Crystal/Resonator
2. HS High-Speed Crystal/Resonator
3. HSPLL High-Speed Crystal/Resonator with PLL Enabled
4. EC External Clock with Fosc/4 Output
5. ECIO External Clock with I/O on RA6
6. ECPLL External Clock with PLL Enabled and Fosc/4 Output on RA6
7. ECPIO External Clock with PLL Enabled, I/O on RA6
8. INTHS Internal Oscillator used as Microcontroller Clock Source, HS Oscillator used as USB Clock Source
9. INTIO Internal Oscillator used as Microcontroller Clock Source, EC Oscillator used as USB Clock Source, Digital I/O on RA6
10. INTCKO Internal Oscillator used as Microcontroller Clock Source, EC Oscillator used as USB Clock Source, Fosc/4 Output on RA6

2.2.1 OSCILLATOR MODES AND USB OPERATION

Because of the unique requirements of the USB module, a different approach to clock operation is necessary. In previous PIC® devices, all core and peripheral clocks were driven by a single oscillator source; the usual sources were primary, secondary or the internal oscillator. With PIC18F2455/2550/4455/4550 devices, the primary oscillator becomes part of the USB module and cannot be associated to any other clock source. Thus, the USB module must be clocked from the primary clock source; however, the microcontroller core and other peripherals can be separately clocked from the secondary or internal oscillators as before.

Because of the timing requirements imposed by USB, an internal clock of either 6 MHz or 48 MHz is required while the USB module is enabled. Fortunately, the microcontroller and other peripherals are not required to run at this clock speed when using the primary oscillator. There are numerous options to achieve the USB module clock requirement and still provide flexibility for clocking the rest of the device from the primary oscillator source. These are detailed in **Section 2.3 "Oscillator Settings for USB"**.

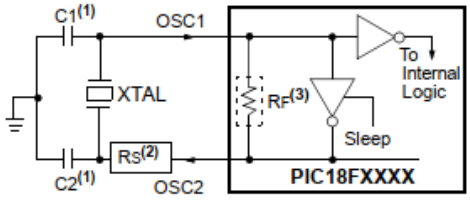
2.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In HS, HSPLL, XT and XTPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-2 shows the pin connections.

The oscillator design requires the use of a parallel cut crystal.

Note:	Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

FIGURE 2-2: CRYSTAL/CERAMIC RESONATOR OPERATION (XT, HS OR HSPLL CONFIGURATION)



Note 1: See Table 2-1 and Table 2-2 for initial values of C1 and C2.
2: A series resistor (RS) may be required for AT strip cut crystals.
3: RF varies with the oscillator mode chosen.

TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	4.0 MHz	33 pF	33 pF
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF

Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. **These values are not optimized**.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes following Table 2-2 for additional information.

Resonators Used:
4.0 MHz
8.0 MHz
16.0 MHz

When using ceramic resonators with frequencies above 3.5 MHz, HS mode is recommended over XT mode. HS mode may be used at any VDD for which the controller is rated. If HS is selected, the gain of the oscillator may overdrive the resonator. Therefore, a series resistor should be placed between the OSC2 pin and the resonator. As a good starting point, the recommended value of RS is 330 Ω.

TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF
Capacitor values are for design guidance only. These capacitors were tested with the crystals listed below for basic start-up and operation. **These values are not optimized.** Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. See the notes following this table for additional information.			
Crystals Used:			
4 MHz			
8 MHz			
20 MHz			

Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time.

2: When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

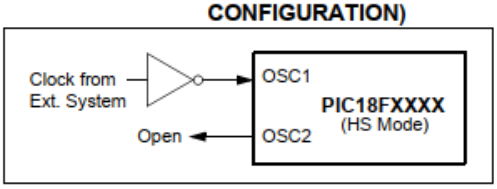
4: Rs may be required to avoid overdriving crystals with low drive level specification.

5: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

An internal postscaler allows users to select a clock frequency other than that of the crystal or resonator. Frequency division is determined by the CPUDIV Configuration bits. Users may select a clock frequency of the oscillator frequency, or 1/2, 1/3 or 1/4 of the frequency.

An external clock may also be used when the micro-controller is in HS Oscillator mode. In this case, the OSC2/CLKO pin is left open (Figure 2-3).

FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)

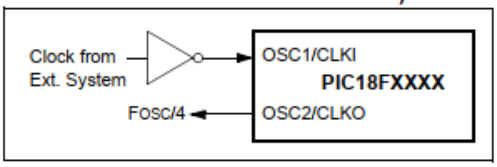


2.2.3 EXTERNAL CLOCK INPUT

The EC, ECIO, ECPLL and ECPIO Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

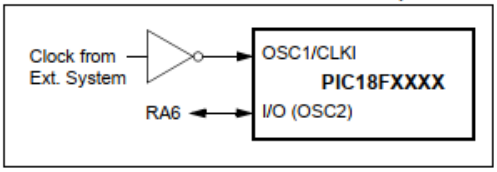
In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (EC AND ECPLL CONFIGURATION)



The ECIO and ECPIO Oscillator modes function like the EC and ECPLL modes, except that the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 2-5 shows the pin connections for the ECIO Oscillator mode.

FIGURE 2-5: EXTERNAL CLOCK INPUT OPERATION (ECIO AND ECPIO CONFIGURATION)



The internal postscaler for reducing clock frequency in XT and HS modes is also available in EC and ECIO modes.

4.2 Master Clear Reset (MCLR)

The MCLR pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the MCLR Reset path which detects and ignores small pulses.

The MCLR pin is not driven low by any internal Resets, including the WDT.

In PIC18F2455/2550/4455/4550 devices, the MCLR input can be disabled with the MCLRE Configuration bit. When MCLR is disabled, the pin becomes a digital input. See **Section 10.5 "PORTE, TRISE and LATE Registers"** for more information.

4.3 Power-on Reset (POR)

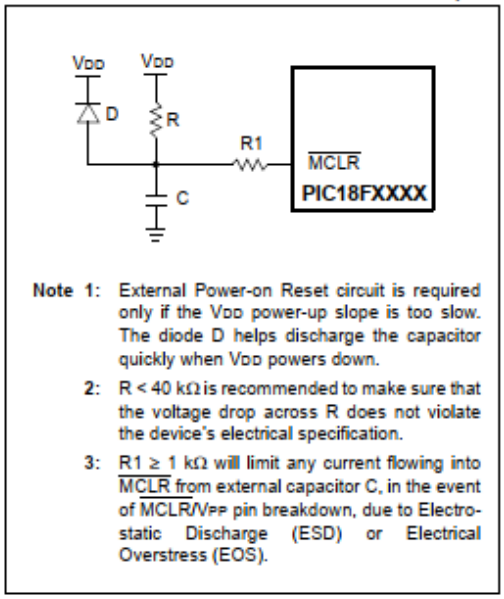
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the MCLR pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004, **Section 28.1 "DC Characteristics"**). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the POR bit (RCON<1>). The state of the bit is set to '0' whenever a POR occurs; it does not change for any other Reset event. POR is not reset to '1' by any hardware event. To capture multiple events, the user manually resets the bit to '1' in software following any POR.

FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 ≥ 1 kΩ will limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

11.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt on overflow

The T0CON register (Register 11-1) controls all aspects of the module's operation, including the prescale selection. It is both readable and writable.

A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 11-1. Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

12.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCP Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 12-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 12-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 12-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from RC0/T1OSO/T13CKI pin (on the rising edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

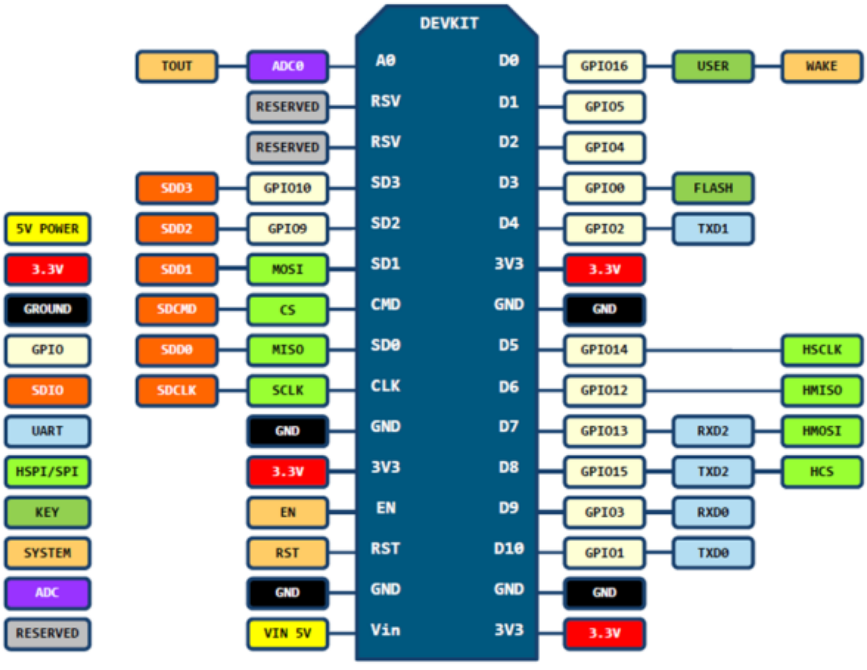
Anexo II

En cuanto al módulo de Wi-Fi NodeMCU ESP8266, se utilizó un manual con sus especificaciones.

1. Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

2. Pin Definition:



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

3. Using Arduino IDE

Anexo III

En este anexo se muestra el datasheet utilizado del sensor ultrasonido, el cual no solo contiene las especificaciones de los pines y alimentación, sino también la ecuación que se necesitó para realizar las pertinentes mediciones.

ELEC Freaks Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

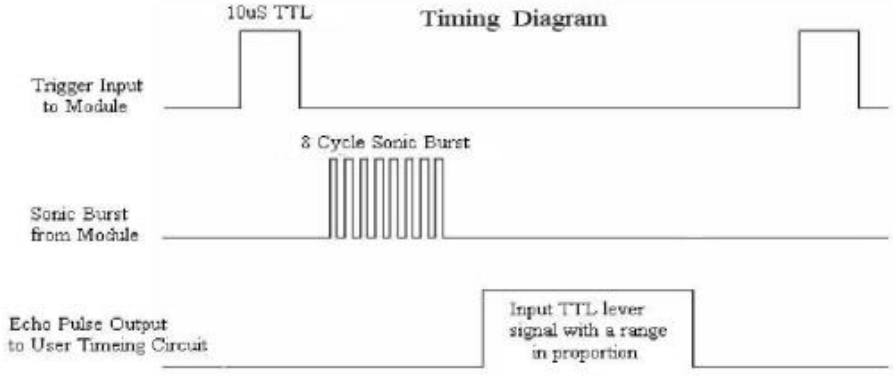
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Anexo IV

Para la correcta alimentación del prototipo se consultó el datasheet del regulador de voltaje LM7805.

TEXAS INSTRUMENTS

LM340, LM340A and LM7805 Family Wide V_{IN} 1.5-A Fixed Voltage Regulators

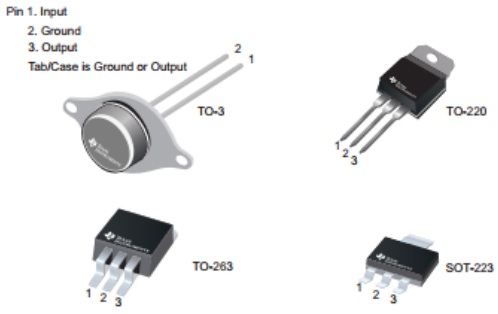
1 Features

- Output Current up to 1.5 A
- Available in Fixed 5-V, 12-V, and 15-V Options
- Output Voltage Tolerances of ±2% at T_J = 25°C (LM340A)
- Line Regulation of 0.01% / V of at 1-A Load (LM340A)
- Load Regulation of 0.3% / A (LM340A)
- Internal Thermal Overload, Short-Circuit and SOA Protection
- Available in Space-Saving SOT-223 Package
- Output Capacitance Not Required for Stability

2 Applications

- Industrial Power Supplies
- SMPS Post Regulation
- HVAC Systems
- AC Inventors
- Test and Measurement Equipment
- Brushed and Brushless DC Motor Drivers
- Solar Energy String Invertors

Available Packages



3 Description

The LM340 and LM7805 Family monolithic 3-terminal positive voltage regulators employ internal current-limiting, thermal shutdown and safe-area compensation, making them essentially indestructible. If adequate heat sinking is provided, they can deliver over 1.5-A output current. They are intended as fixed voltage regulators in a wide range of applications including local (on-card) regulation for elimination of noise and distribution problems associated with single-point regulation. In addition to use as fixed voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents.

Considerable effort was expended to make the entire series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

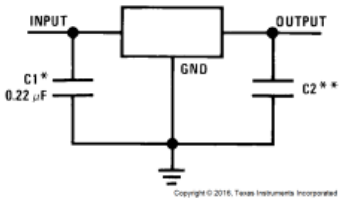
LM7805 is also available in a higher accuracy and better performance version (LM340A). Refer to LM340A specifications in the *LM340A Electrical Characterisitcs* table.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM340x LM7805 Family	DDPAK/TO-263 (3)	10.18 mm × 8.41 mm
	SOT-223 (4)	6.50 mm × 3.50 mm
	TO-220 (3)	14.986 mm × 10.16 mm
	TO-3 (2)	38.94 mm x 25.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Fixed Output Voltage Regulator



*Required if the regulator is located far from the power supply filter.
**Although no output capacitor is needed for stability, it does help transient response. (If needed, use 0.1-µF, ceramic disc).

Typical Characteristics (continued)

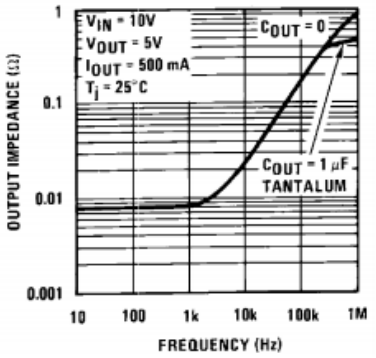


Figure 7. Output Impedance

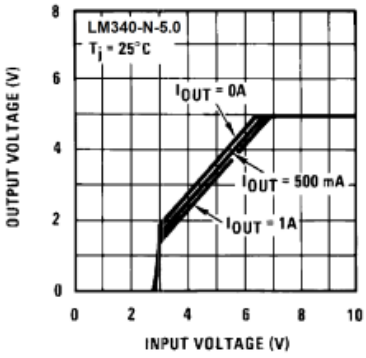
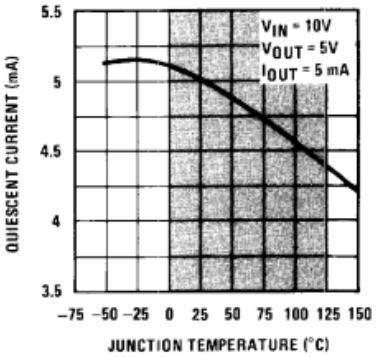


Figure 8. Dropout Characteristics



Shaded area refers to LM340A/LM340, LM7805, LM7812, and LM7815.

Figure 9. Quiescent Current

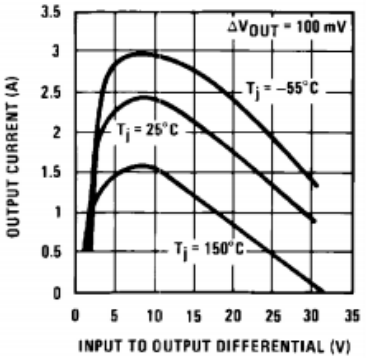
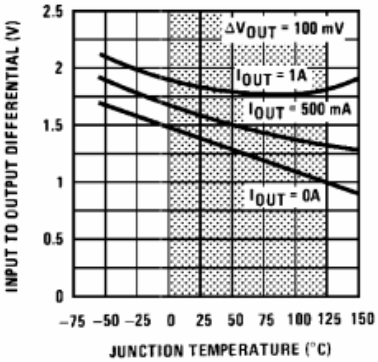


Figure 10. Peak Output Current



Shaded area refers to LM340A/LM340, LM7805, LM7812, and LM7815.

Figure 11. Dropout Voltage

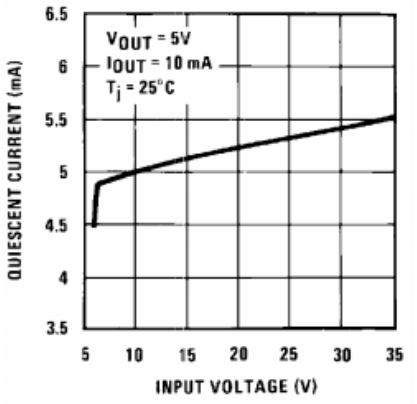


Figure 12. Quiescent Current

Bibliografía

[1] Microchip Technology Inc; PIC18F2455/2550/4455/4550 datasheet; DS39632E; 2009.

[2] ELECFreaks; HC-SR04 datasheet; 2020.

[3] Handson Technology; ESP8266 NodeMCU WiFi Development Board; 2019.

[4] Eduardo García Breijo; *Compilador C Ccs Y Simulador Proteus Para Microcontroladores Pic*; Alfaomega; 2008.

[5] Irving L. Kosow; *Máquinas Eléctricas y Transformadores*; Prentice Hall; 1991.

[6] Hector O. Pueyo, Carlos Marco; *Análisis de modelos circuitales Tomo 1*; Arbó; 1993.

[7] William Stalling; *Comunicaciones y redes de computadores*; Pearson Prentice Hall; 2004.

[8] Leonard Marks, James A. Caterina; *Printed Circuits Assembly Design*; Mc Graw-Hill; 2000.

[9] Software de cálculo de muelles
https://www.muellestock.com/es/producto/search?tipo=Compresi%C3%B3n

[10] Texas Instruments; regulador de voltaje LM7805 datasheet; 2016