

Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Utilización de un Radar Jtron HB100 y un microcontrolador Arduino para la medida de velocidad de blancos móviles.

Grado en Ingeniería Mecánica

ALUMNO: Cristóbal González-Aller Cornago.

DIRECTORES: Norberto Fernández García.

Paula Gómez Pérez.

CURSO ACADÉMICO: 2014-2015

Universidad de Vigo



Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Utilización de un Radar Jtron HB100 y un microcontrolador Arduino para la medida de velocidad de blancos móviles

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General

Universidad de Vigo

RESUMEN

El objetivo de este TFG consiste en montar físicamente un sistema que:

1) Lleve a cabo una acción (por ejemplo: encendido de un LED) cuando detecte movimiento dentro de un alcance. Este modo de funcionamiento sería similar al mecanismo utilizado para encender una luz o abrir automáticamente una puerta al detectar movimiento.

2) Sea capaz de estimar la velocidad con la que un objetivo se aproxima o aleja, de un modo similar a cómo lo haría un velocímetro de la policía.

Para ello se utilizará un sensor radar de microondas HB100 de onda continua, baja potencia y bajo coste, capaz de obtener la frecuencia Doppler de una señal por medio de la comparación entre la señal transmitida por el radar y la recibida. Se montará un circuito acondicionador, para amplificar la señal del radar hasta un nivel fácilmente procesable. Por último, se utilizará un microcontrolador Arduino que procesará la señal recibida y, por un lado será capaz de decidir si hay o no presencia de movimiento y llevar a cabo una acción, y por otro lado, de la frecuencia Doppler obtendrá un valor de la velocidad y lo presentará en un *display* LCD (*Liquid Crystal Display*).

Como resultado del proyecto se pretende evaluar la viabilidad de utilizar el módulo radar HB100 como herramienta docente de la asignatura de Sistemas de Control y Sensores Navales de 5º curso del grado.

PALABRAS CLAVE

Radar, Efecto Doppler, Microcontrolador Arduino, Velocímetro.

CONTENIDO

Contenido 1
Índice de Figuras 3
Índice de Tablas 5
1 Introducción y objetivos 6
1.1 Contextualización 6
1.2 Motivación 7
1.3 Objetivos 7
1.4 Organización de la memoria 8
2 Estado del arte 9
2.1 Radar 9
2.1.1 Introducción y orígenes históricos 9
2.1.2 Principios de funcionamiento y tipos de radar 11
2.1.3 Efecto Doppler 13
2.2 Sensor radar HB100 14
2.2.1 Características y elementos 14
2.2.2 Funcionamiento del Radar HB100 16
2.3 Arduino 18
2.3.1 Introducción y orígenes históricos 18
2.3.2 El Hardware de Arduino 20
2.3.3 Entorno de desarrollo 21
2.3.4 Ejemplo de uso 23
3 Desarrollo del TFG 25
3.1 Etapas de desarrollo del TFG 25
3.1.1 Etapa de Documentación 25
3.1.2 Etapa de análisis del circuito electrónico de acondicionamiento. 26
3.1.3 Montaje del circuito de acondicionamiento. 29
3.1.4 Verificación del circuito de acondicionamiento 31
3.1.5 Conexión del circuito con el microcontrolador Arduino 34
3.1.6 Independizar el montaje del PC 35
4 Validación 40
4.1 Comprobación del funcionamiento del programa 40
4.2 Estimación del umbral de detección 41
4.3 Pruebas de alcance del sistema 42

4.4 Pruebas de medida de velocidad 44
4.4.1 Prueba de velocidad “lenta” 44
4.4.2 Prueba de velocidad “rápida” 46
4.4.3 Prueba en exteriores 47
5 Conclusiones y líneas futuras 48
5.1 Conclusiones 48
5.2 Líneas Futuras 48
6 Bibliografía 49
Anexo I: Código detección/no detección 52
Anexo II: Código velocidad promedio 54
Anexo III: Código velocidad máxima 59

ÍNDICE DE FIGURAS

Figura 2-1 Imagen de las antenas del Chain Home [5] 10
Figura 2-2 Mapa esquemático de la distribución del Chain Home [5] 11
Figura 2-3 Secuencia de funcionamiento del radar 12
Figura 2-4 Esquema de Onda Electromagnética Pulsada 13
Figura 2-5 Efecto Doppler 14
Figura 2-6 Imagen física del Módulo HB100 Figura 2-7 Diagrama del Módulo HB100 15
Figura 2-8 Interior del Módulo HB100 15
Figura 2-9 Diagramas de radiación del radar HB100 (Izda: Azimut - Dcha: Elevación) 16
Figura 2-10 Circuito acondicionador para modo onda continua [11] 18
Figura 2-11 Logotipo Arduino 18
Figura 2-12 Una de las primeras placas de Arduino 19
Figura 2-13 Modelos de placa Arduino (De izda. a dcha. 1-Uno 2-Robot 3-Mega 4-Mini) 20
Figura 2-14 Microcontrolador ATmega 328 20
Figura 2-15 Explicación de los elementos de la placa Arduino UNO 21
Figura 2-16 Entorno de desarrollo de Arduino 22
Figura 2-17 Esquema de conexiones del ejemplo de uso 23
Figura 2-18 Programa utilizado en el ejemplo de uso 24
Figura 3-1 Kit de aprendizaje de Arduino 26
Figura 3-2 Representación del circuito de acondicionamiento en *PSim* 27
Figura 3-3 Señal a la salida del generador de corriente alterna (simulado en el *PSim*) 27
Figura 3-4 Señal a la salida de la primera etapa de acondicionamiento (simulado en el *PSim*) 28
Figura 3-5 Señal a la salida del circuito de acondicionamiento (simulado en el *PSim*) 28
Figura 3-6 Sensor HB100 conectado a alimentación (Izda: Esquema - Dcha: Real) 29
Figura 3-7 Primera etapa del circuito acondicionador (Izda: Esquema - Dcha: Real) 30
Figura 3-8 Segunda etapa del circuito de acondicionamiento (Izda: Esquema - Dcha: Real) 30
Figura 3-9 Tercera etapa del circuito de acondicionamiento (Izda: Esquema - Dcha: Real) 31
Figura 3-10 Circuito de acondicionamiento completo (Izda: Esquema - Dcha: Real por etapas) ... 31
Figura 3-11 Izda: Alimentación de 5 V - Dcha: Señal generada de amplitud 0,05 V y frecuencia 100,00 Hz 32
Figura 3-12 Señal generada medida con osciloscopio 32
Figura 3-13 Señal generada medida con osciloscopio a la salida de la primera etapa 33
Figura 3-14 Señal generada medida con osciloscopio a la salida final del circuito 33
Figura 3-15 Señal del radar medida con osciloscopio a la salida final del circuito 34

Figura 3-16 Circuito acondicionador montado sobre una única *protoboard* 34
Figura 3-17 Batería con entradas USB 36
Figura 3-18 Cable USB para conexión de la placa Arduino..... 36
Figura 3-19 *Display* LCD 36
Figura 3-20 Sistema independiente del PC..... 37
Figura 3-21 Sistema detectando presencia de movimiento 38
Figura 3-22 Sistema mostrando velocidad media calculada..... 38
Figura 3-23 Sistema mostrando velocidad máxima superior al límite 39
Figura 4-1 Sistema mostrando resultados de procesamiento de señal generada 40
Figura 4-2 Gráfica que muestra resultados de prueba de umbral de detección 41
Figura 4-3 *Boxplo*t de la prueba de determinación del umbral de detección..... 42
Figura 4-4 Gráfica que muestra resultados de prueba de alcance..... 43
Figura 4-5 *Boxplot* de la prueba de alcance..... 43
Figura 4-6 Representación de resultados obtenidos en prueba de medida de velocidad “lenta”..... 44
Figura 4-7 Representación de resultados obtenidos en prueba de medida de velocidad “rápida” .. 46

ÍNDICE DE TABLAS

Tabla 2-1 Factor de Conversión en función de F_t [11]..... 17

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contextualización

El uso de sistemas radar no es algo nuevo del siglo XXI. Sus inicios se remontan incluso a finales del siglo XIX, cuando se empiezan a estudiar las propiedades de las señales electromagnéticas. Desde ese momento hasta ahora los sistemas radar han ido mejorando de manera progresiva hasta llegar a los sistemas de la actualidad, que cuentan con unas grandes capacidades y tienen muy diversas aplicaciones.

Los sistemas radar entendidos como tales, tienen un origen eminentemente militar. Los primeros fueron utilizados en la Segunda Guerra Mundial para detectar la posición de las aeronaves enemigas.

Tener unos mejores equipos de detección que el enemigo, significa obtener una gran ventaja. Por este motivo, durante la Segunda Guerra Mundial, y posteriormente a lo largo de los años, estas tecnologías han ido avanzando mucho en el ámbito militar. Sobre todo en el entorno marítimo donde hoy en día es imposible encontrar un buque de guerra sin al menos un radar de navegación.

Sin embargo, estos sistemas de navegación no sólo se usan en el entorno militar sino que su uso se extendió a la vida civil. También es prácticamente imposible encontrar un buque mercante, pesquero o de cualquier tipo que no cuente con radares de navegación. Hoy en día un sistema de control del tráfico marítimo o del tráfico aéreo sin un buen sistema de posicionamiento radar no tiene mucho sentido.

Pero el uso de radares no queda limitado al ámbito de la navegación y el control aéreo. En la vida cotidiana se pueden encontrar fácilmente tecnologías que cuentan con estos sistemas a menor escala. Como por ejemplo una puerta automática o una luz que se enciende cuando hay alguien en la habitación, los radares de policía que detectan la velocidad de los coches en circulación, etc.

Este proyecto está relacionado con estos últimos ejemplos de uso de sensores radar, ya que consiste en detectar presencia de movimiento y medir velocidad a corto alcance.

Sin embargo, el sensor radar que se va a utilizar no es capaz por sí mismo de decidir si hay movimiento o no, ni de informar de la velocidad de un blanco. Para conseguir esto, es necesario procesar la información que el radar envía, con un microcontrolador.

El uso de microcontroladores es algo mucho más reciente que el uso de los radares. No obstante, en los últimos años, se ha avanzado mucho en el uso de la electrónica y la programación informática.

En el ámbito de la educación, una plataforma muy adecuada para introducirse en el mundo de la programación informática y en el uso de la electrónica, es la plataforma Arduino. Esta plataforma

permite al usuario realizar desde pequeños proyectos o diseños muy simples hasta otros mucho más complejos.

En el presente proyecto, se utilizará un microcontrolador Arduino para procesar la información del radar detectando movimiento y estimando la velocidad del móvil.

1.2 Motivación

El motivo por el que se plantea el presente proyecto es meramente académico. Busca por un lado, mejorar la formación del alumno designado para su realización, y por otro lado, evaluar la posible utilización del sistema radar resultante en la asignatura de Sistemas de Control y Sensores Navales de 5º curso o como base para plantear posibles TFG's en el futuro.

Este proyecto abarca una rama de la ingeniería que incluye varias materias cursadas a lo largo de todo el grado, en asignaturas como Informática para la Ingeniería, Tecnología Electrónica o Sistemas de Control y Sensores Navales. Tanto en estas asignaturas como en el presente TFG se estudia programación, análisis de circuitos electrónicos y de los diferentes tipos de radar y su funcionamiento.

Este proyecto es, por tanto una culminación del proceso de aprendizaje, en el que se busca que el alumno lleve a la práctica los conocimientos adquiridos durante el estudio de las asignaturas mencionadas.

Al ser el resultado del proyecto un sistema de detección de velocidad por frecuencia Doppler, la relación con la asignatura de Sistemas de Control y Sensores Navales es muy elevada. Es por esto por lo que se pretende evaluar el módulo radar de cara a su posible utilización como herramienta docente en la asignatura del último año de titulación.

1.3 Objetivos

El objetivo fundamental de este TFG queda perfectamente recogido en el nombre del proyecto: "*Utilización de un Radar Jtron HB100 y un microcontrolador Arduino para la medida de velocidad de blancos móviles"*.

Para llegar a este objetivo principal, es necesario ir avanzando a través de una serie de etapas intermedias, marcándose metas cortas. Estas etapas se pueden definir como objetivos segundarios del proyecto, y son:

- Conocer bien el sensor radar *Jtron* HB100. Cuáles son sus características y sus capacidades.
- Conocer la plataforma Arduino, y familiarizarse con el lenguaje de programación que éste utiliza.
- Analizar y posteriormente realizar el montaje del circuito electrónico de acondicionamiento de la señal de salida del sensor.
- Conectar el módulo radar con el circuito de acondicionamiento al microcontrolador Arduino para procesar la señal de salida y ser capaces de:
    - Tomar decisiones sobre la detección o no detección de un blanco en movimiento.
    - Estimar la velocidad del blanco a partir de la frecuencia Doppler.
- Realizar un montaje manejable e independiente del PC, haciendo uso de un *display* LCD (*Liquid Crystal Display*) y una batería.

Como se puede observar tanto en el objetivo principal como en los secundarios, este proyecto no busca la creación de algún nuevo tipo de sistema o equipo ni la mejora de otros sistemas usados en la actualidad para incrementar sus capacidades. El objetivo es evaluar el módulo radar de cara a su posible utilización como herramienta docente en la asignatura de Sistemas de Control y Sensores Navales.

1.4 Organización de la memoria

El resto de la memoria se estructura como sigue:

- En el capitulo 2 (“Estado del arte”) se habla de la historia del radar, qué tipos hay y cuál es su funcionamiento. Así como también se habla sobre la plataforma Arduino, historia, tipos y partes de la plataforma.

- En el capítulo 3 (“Desarrollo del TFG”) se explica por etapas cómo se ha ido desarrollando el proyecto a lo largo del tiempo.

- En el capítulo 4 (“Validación”) se mencionan las pruebas que se realizaron una vez el sistema radar estaba terminado. Se explican y se sacan algunas conclusiones.

- En el capítulo 5 (“Conclusiones”) finalmente se cierra el proyecto con unas conclusiones obtenidas de la realización del mismo.

- En el capítulo 6 (“Bibliografía”) se citan todas las referencias que se han utilizado para el desarrollo de la memoria del presente TFG.

- Por último se anexan al final del presente documento los código utilizados para el microcontrolador Arduino

2 ESTADO DEL ARTE

2.1 Radar

2.1.1 Introducción y orígenes históricos

"El RADAR es un sistema que utiliza radiaciones electromagnéticas reflejadas por un objeto para determinar la localización o velocidad de este" [1].

Aunque los primeros radares empezaron a aparecer en los años 30, el inicio de su desarrollo tecnológico se remonta al año 1864, cuando el inglés James Clerk Maxwell describe las leyes del electromagnetismo. Más tarde, en 1886, el físico Heinrich Herz, basándose en las ecuaciones de Maxwell, consigue demostrar que las ondas electromagnéticas se reflejan en objetos sólidos [2] [3].

En el año 1904, un investigador alemán, Christian Hulsmeyer, desarrolló un equipo llamado "*Telemobiloskop*" capaz de detectar las ondas electromagnéticas que se reflejaban en los buques, con el inconveniente de que no era capaz de determinar la posición del blanco y que solo conseguía un alcance de aproximadamente una milla [2] [3] [4].

En el año 1917, Nicola Tesla habla sobre los principios del futuro radar. "*Por su uso podemos producir a voluntad, de una estación de envío, un efecto eléctrico en cualquier región del mundo en particular; podemos determinar la posición relativa o el curso de un objeto en movimiento, tal como un buque en el mar, la distancia atravesada por el mismo, o su velocidad*" [3].

Unos años más tarde, dos científicos norteamericanos, Hoyt Taylor y L. Young realizaron un experimento emitiendo una señal de onda continua, y descubrieron que cuando pasaba un blanco por delante, se producían alteraciones en la señal recibida [2] [3].

En 1935, el ingeniero inglés Watson-Watt logró la primera detección de un avión a 15 millas de distancia. Y más tarde en ese mismo año, lo logró a una distancia de 40 millas. El dispositivo con el que lo consiguió se llamó RDF (*Radio Detection Finding)* [2] [3] [4].

En el año 1936 los norteamericanos Page y Young desarrollan el primer radar pulsado llegando a tener un alcance de 25 millas. Posteriormente, estos sistemas de detección pasarían a conocerse por su nombre actual, RADAR (*RAdio Detection And Ranging*) [2].

Winston Churchill aprueba el traspaso de tecnología con EEUU y comienza entonces un trabajo conjunto en esta materia. Consiguieron grandes avances, durante los años de pre-guerra y durante la guerra [2] [5].

El siguiente paso que adoptaron los ingleses fue el de montar un radar a bordo de una aeronave, lo cual se consiguió en el año 1937 [2] [5].

En el año 1938 los ingleses instalaron en sus costas este y sur una cadena de antenas conocidas como "*Chain Home*" (*Figuras 2-1 y 2-2*). Fue el primer radar de vigilancia aérea de la historia y desempeñó un papel crucial en la "Batalla de Inglaterra", ya que este sistema era muy superior a los sensores alemanes [2] [4] [6].

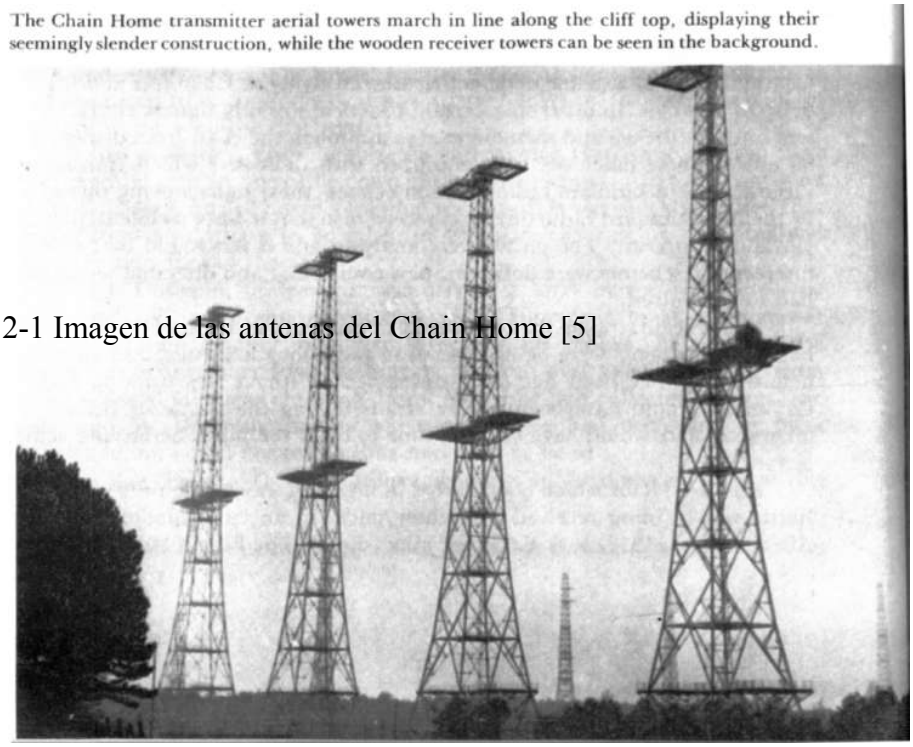


Figura 2-1 Imagen de las antenas del Chain Home [5]

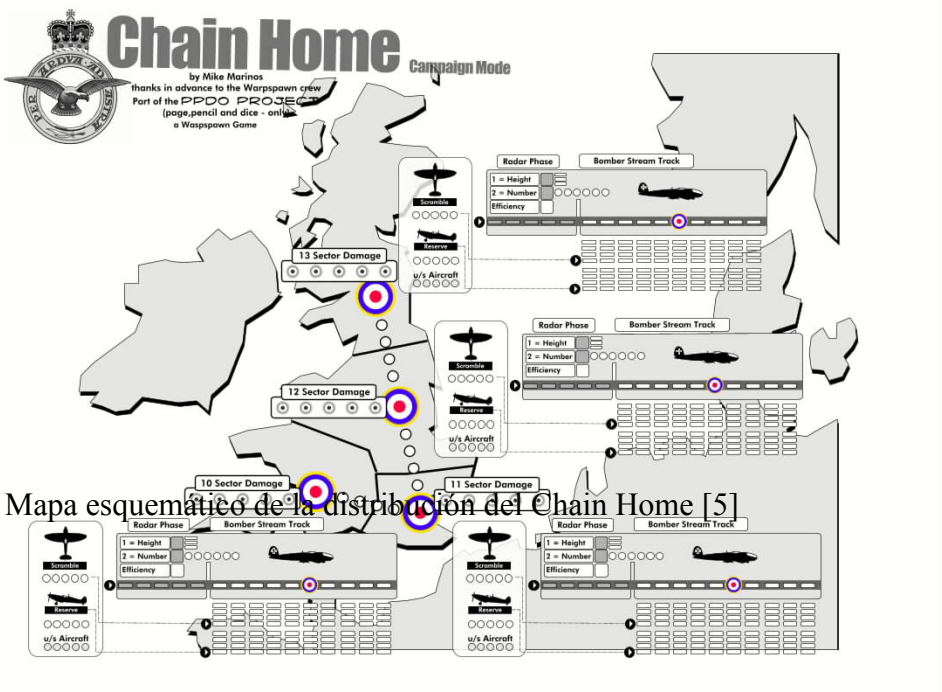


Figura 2-2 Mapa esquemático de la distribución del Chain Home [5]

En el año 1940, los científicos Henry Boot y John Randall consiguen crear un oscilador resonante para ondas centimétricas (conocidas hoy en día como microondas); y el físico Roger Sutton, inventa un amplificador para este tipo de ondas [2].

Los grandes países que participaron en la guerra (EEUU, Inglaterra, Alemania, Italia, Rusia, Japón) fueron desarrollando diferentes sistemas radar, aportando grandes avances para llegar hasta lo que hoy en día entendemos por radar.

2.1.2 Principios de funcionamiento y tipos de radar

El funcionamiento de un radar se basa fundamentalmente en el principio de reflexión de las ondas electromagnéticas. Cuando hay un cambio significativo de las constantes dieléctricas o diamagnéticas, las ondas electromagnéticas se ven afectadas. Básicamente, lo que quiere decir esto es que las ondas electromagnéticas rebotan en los objetos sólidos [7] [8].

Ésta es la razón fundamental por la que somos capaces de detectar un objeto con señales electromagnéticas. La secuencia de funcionamiento es la siguiente: Primero el radar emite al exterior una onda electromagnética. Esta señal electromagnética se refleja en los diferentes objetos que estén dentro del alcance del radar. Por último, parte de la señal reflejada irá en dirección a un receptor, que detectará esa señal, obteniendo así información sobre los objetos existentes en el medio de propagación. Esta secuencia se puede observar en la *figura 2-3* [7] [8].

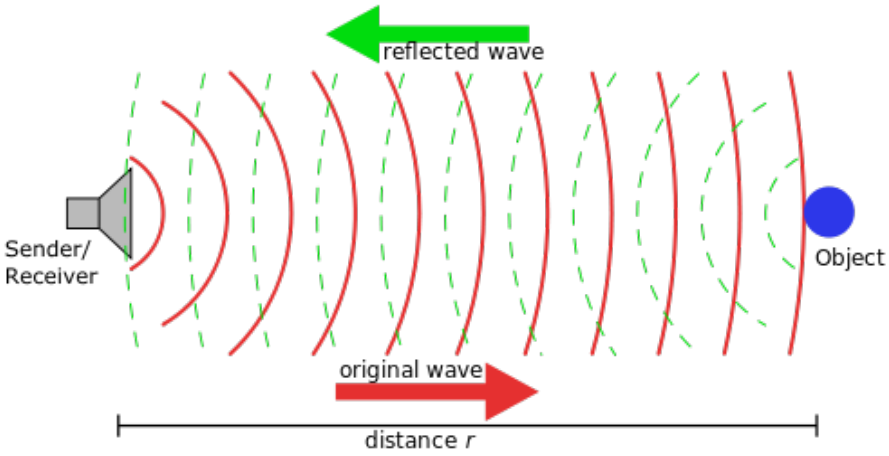


Figura 2-3 Secuencia de funcionamiento del radar

Durante el camino que recorre la señal electromagnética, ésta se atenúa. La potencia de la señal recibida no es igual a la de la señal transmitida, sino que varía en función de las características de la antena, la superficie, el material del objeto y de la distancia al mismo. Se muestra a continuación la ecuación fundamental radar (*ecuación (1)*):

$$\mathrm{Pr} = \frac{\mathrm{Pt\ Gt\ Ae\ \sigma}}{(4\pi)^2 R^4} \quad (1)$$

Según [9], los radares se pueden clasificar de diversas maneras:

Según el número de antenas:

- Mono-estáticos: *"Radar convencional, en el cual el transmisor y el receptor están en el mismo lugar y comparten la misma antena; en contraste con el radar biestático".*
- Bi-estáticos: *"Radar en el cual el receptor está a alguna distancia del transmisor, con antenas separadas para cada uno".*
- Multi-estáticos: *"Radar en el cual los lóbulos sucesivos de la antena están conectados secuencialmente para proporcionar una capacidad de rastreo sin movimiento físico de la antena".*

Según el tipo de blanco:

- Primario: *"Radar en el cual el haz incidente se refleja desde el blanco para formar la señal de retorno".*
- Secundario: *"Radar en el que el receptor analiza las ondas emitidas por un transpondedor asociado con el objetivo".*

Según su finalidad:

- De búsqueda: *"Radar ideado fundamentalmente para cubrir una vasta región del espacio y para visualizar blancos tan pronto como entran en la región; utilizado para detección lejana,*

en conexión con la aproximación e interceptación controlada desde tierra y en el control del tráfico aéreo".

- De seguimiento: *"Radar capaz de seguir el movimiento de un blanco".*

Pero la quizás la clasificación más importante, es según el tipo de onda con la que el radar transmite:

- Onda pulsada: Los radares de este tipo, emiten una señal electromagnética en pequeños pulsos, cuya duración es del orden de las millonésimas de segundo (τ). Entre pulso y pulso, el radar "escucha". Sabiendo la velocidad de propagación de las ondas electromagnéticas y midiendo el tiempo desde que se transmite el pulso hasta que se recibe, el radar es capaz de determinar la distancia al blanco. Emitiendo varios pulsos separados un determinado periodo de tiempo PRI (*Pulse Repetition Interval)*, el radar es capaz de hacer un seguimiento y de esta manera determinar la velocidad del blanco y su dirección [7] [8].

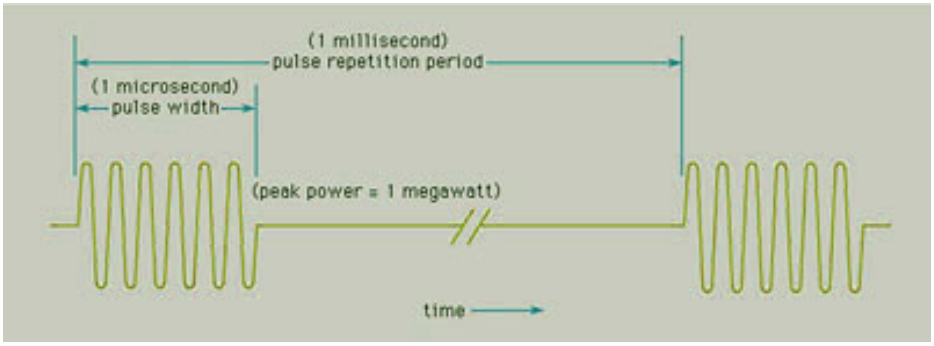


Figura 2-4 Esquema de Onda Electromagnética Pulsada

- Onda continua: Este tipo de radar emite una señal electromagnética de manera continua, a diferencia de los radares pulsados. El funcionamiento de los radares de onda continua, se basa principalmente en el efecto Doppler [7] [8].

2.1.3 Efecto Doppler

El efecto Doppler es la diferencia aparente que se produce entre la frecuencia con que se emite un sonido o una onda electromagnética y la frecuencia con que un observador la recibe, debido al movimiento relativo entre el emisor y el observador. Debe su nombre al físico austriaco Christian Andreas Doppler, quien descubrió el fenómeno en 1842 [7] [8].

De esta manera, cuando la velocidad relativa entre el emisor y el observador es de acercamiento, la frecuencia de la onda recibida será mayor que la transmitida. Por el contrario, si es de alejamiento, la frecuencia recibida será menor que la transmitida. Este efecto se puede observar en la *figura 2-5* [7] [8].

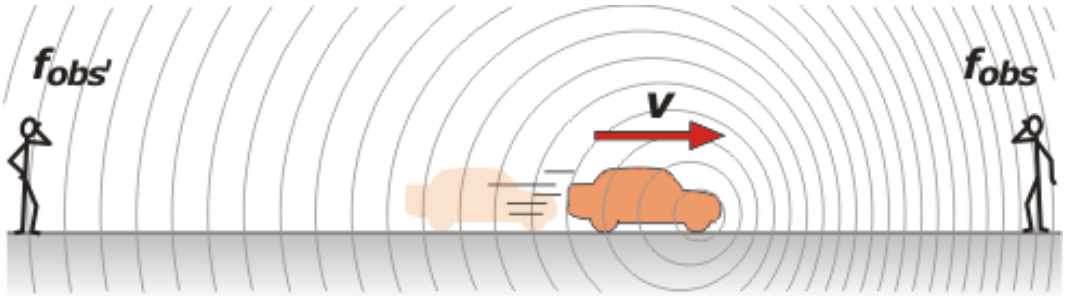


Figura 2-5 Efecto Doppler

La diferencia entre la frecuencia de transmisión y la frecuencia de recepción es la frecuencia Doppler. Gracias a este efecto podemos calcular la velocidad de un blanco, tal como se indica en la *ecuación (2)*.

$$\mathrm{Fo} = \left(\frac{V \mp Vo}{V \mp Vf}\right) \quad (2)$$

Donde:

- Fo: Frecuencia observada en el receptor.
- Ff: Frecuencia de la fuente (emisor)
- V: Velocidad de propagación de la onda.
- Vo: Velocidad del observador.
- Vf: Velocidad de la fuente.

Estos radares de onda continua son, como hemos visto, capaces de detectar la velocidad de un blanco, pero no son capaces de detectar su posición. Hay otro tipo de radares de onda continua que sí son capaces de detectar la posición del blanco aparte de la velocidad. Estos son los radares de onda continua con modulación. Consisten básicamente en añadir a la señal electromagnética una modulación de fase o de frecuencia [7] [8].

2.2 Sensor radar HB100

2.2.1 Características y elementos

El sensor de microondas HB100 es un módulo transceptor Doppler bi-estático, que trabaja en la banda X del espectro de frecuencias (7GHz – 12,5GHz). Este módulo está diseñado para detección de movimiento, en aplicaciones como: alarma de intrusos, sensor para puertas automáticas, etc. [10].

El módulo es un LPRD (*Low Power Radio Device*). Consiste en un oscilador resonador dieléctrico, un mezclador de microondas, una antena de parche de transmisión y otra de recepción, como podemos ver en la *figura 2-7* [11].



Figura 2-6 Imagen física del Módulo HB100

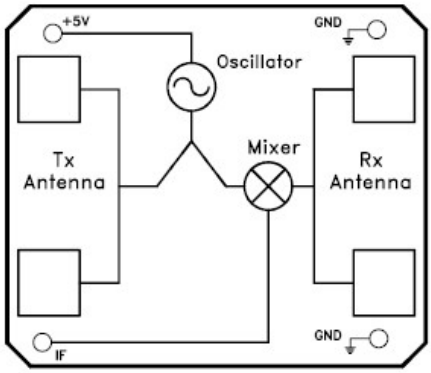


Figura 2-7 Diagrama del Módulo HB100

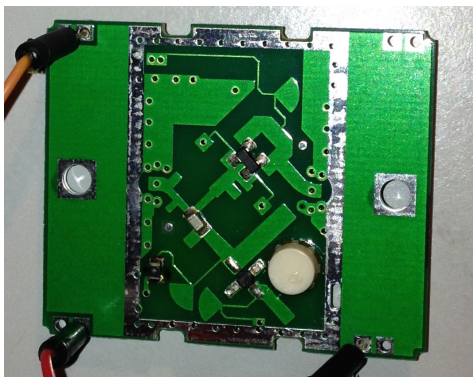


Figura 2-8 Interior del Módulo HB100

Para que quede más claro el funcionamiento del módulo radar, se describirán a continuación cada uno de los elementos fundamentales que lo forman:

- Oscilador: Un oscilador es un sistema electrónico capaz de generar una señal de salida periódica, sin necesidad de aplicar una corriente alterna a la entrada. Básicamente, es un elemento que transforma energía continua en energía alterna. El oscilador resonador dieléctrico es un tipo de oscilador que trabaja en la banda de las microondas. Este elemento es el que va a generar una frecuencia en la señal, para posteriormente emitirla en forma de ondas electromagnéticas a través de la antena de transmisión. La frecuencia de resonancia depende fundamentalmente de las dimensiones físicas del resonador y la constante dieléctrica del material [12]. En concreto, la frecuencia teórica generada por este oscilador es de 10,525GHz [10].

- Mezclador: Un mezclador es un dispositivo capaz de mezclar dos señales de entrada de diferente frecuencia, y producir a la salida una señal resultante con una frecuencia igual a una combinación lineal de las frecuencias de las señales de entrada. Un mezclador consta de tres puertos: dos puertos de entrada, uno para recibir la señal del oscilador, y otro para recibir la señal de la antena receptora, y un tercer puerto de salida para la señal resultante o puerto de frecuencia intermedia (IF- *Intermediate Frecuency*). Esta frecuencia intermedia, será la

diferencia entre la frecuencia de la señal transmitida, generada por el oscilador, y la de la señal recibida por la antena receptora, es decir, la frecuencia Doppler [13].

- Antenas: Una antena es un dispositivo electrónico diseñado para emitir al exterior, o recibir del exterior ondas electromagnéticas. Este módulo radar, cuenta con una antena transmisora, encargada de transformar una señal eléctrica en ondas electromagnéticas y transmitirlas; y una antena receptora, encargada de recibir las ondas electromagnéticas de vuelta y transformarlas en una señal eléctrica. Las antenas de parche son un tipo de antenas adecuadas para dispositivos de tamaño reducido, que trabajan en frecuencias elevadas, principalmente en los rangos de microondas y ondas milimétricas. Como se puede ver en los diagramas de radiación[1] (*figura 2-9*), no son antenas muy direccionales, lo cual no es un problema para este módulo radar, teniendo en cuenta sus principales aplicaciones [14] [15].

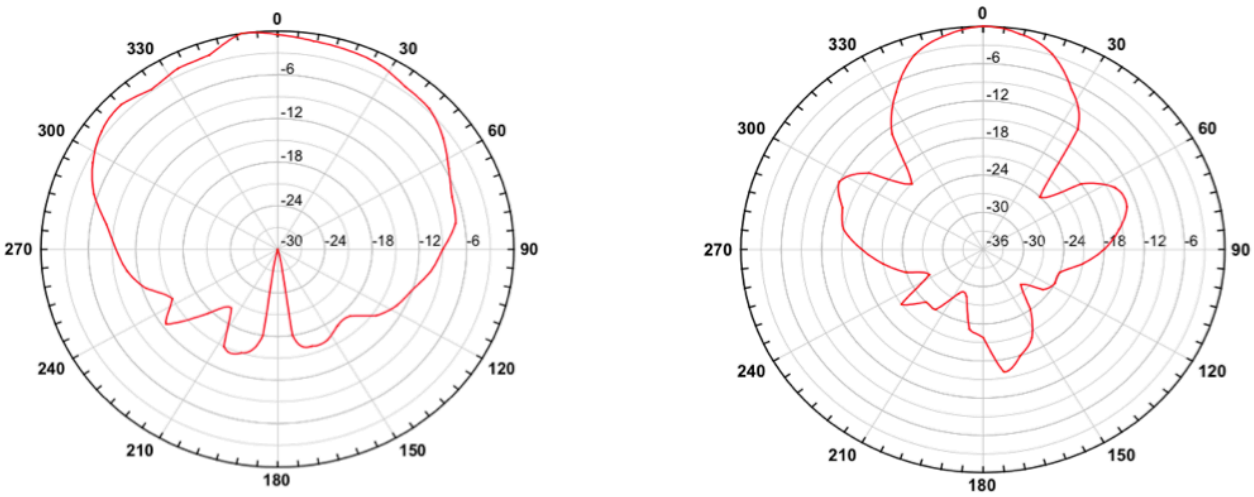


Figura 2-9 Diagramas de radiación del radar HB100 (Izda: Azimut - Dcha: Elevación)

Como se puede ver en la *figura 2-7*, el módulo cuenta con cuatro terminales: por un lado, uno de +5V y dos tomas a tierra (GND). Estos terminales serán utilizados como fuente de alimentación. Por otro lado, un terminal IF, que es el puerto de salida de la señal a procesar con objeto de obtener la velocidad del blanco. El módulo puede trabajar tanto en onda continua como en onda pulsada, en función de la alimentación a la que se conecte. En este proyecto, se trabajará en el modo onda continua. Por tanto, para calcular la velocidad, hay que basarse en las ecuaciones Doppler [10] [11].

2.2.2 Funcionamiento del Radar HB100

El funcionamiento del módulo radar HB100 es simple. El oscilador recibe un voltaje de +5V generando una señal eléctrica con una frecuencia de 10,525 GHz. Al ser la alimentación de +5V e invariable, la frecuencia que genera el oscilador, y por tanto con la que emitirá el radar, será continua [10] [11].

[1] Diagrama de radiación: Representación gráfica de las características de radiación en función de la dirección angular. Puede ser en azimut y en elevación.

Esta señal eléctrica es enviada a la antena transmisora y al mezclador. La antena transmisora se encarga de transformarla en una señal de ondas electromagnéticas y la emite al exterior de manera continuada.

La señal viaja por el aire hasta que se encuentra con un blanco, donde rebota y vuelve al módulo radar. Esta señal de vuelta es recibida por la antena receptora, que la transforma de nuevo en una señal eléctrica y la manda al mezclador.

El mezclador recibe del oscilador la señal con la frecuencia de transmisión y de la antena receptora la señal con la frecuencia de recepción. La señal a la salida del mezclador tendrá una frecuencia que será la diferencia entre la frecuencia de transmisión y la de recepción, es decir la frecuencia Doppler (F_d) y por tanto tiene una relación directa con la velocidad del blanco. De acuerdo con la hoja de características del módulo HB100, esta relación viene expresada por la *ecuación (3)* [10] [11].

$$\mathrm{Fd} = 2\mathrm{V}\left(\frac{\mathrm{Ft}}{c}\right) Cos\,(\theta) \quad (3)$$

- F_d: Frecuencia Doppler.
- F_t: Frecuencia de transmisión.
- V: Velocidad del blanco.
- c: Velocidad de la luz.
- Cos (θ): Coseno del ángulo de aproximación/alejamiento del blanco.

Si el objeto a detectar se mueve hacia el radar o se aleja del radar directamente, es decir, $\theta = 0$, la fórmula se simplifica a: F_d = 19.49V (Velocidad en Km/hora) para F_t = 10,525 GHz. Se puede ver en la *tabla 2-1* una relación de valores de F_d en función de la F_t.

Freq. De Transmisión (F_t)	**Freq. Doppler (F_d) – V(Km/h)**
9.35 GHz	17.31 V
9.9 GHz	18.33 V
10.525 GHz	19.49 V
10.587 GHz	19.60 V
10.687 GHz	19.69 V
24.125 GHz	44.68 V

Tabla 2-1 Factor de Conversión en función de F_t [11]

Esta señal a la salida del mezclador es la que será procesada por el microcontrolador, para obtener la velocidad del blanco. El inconveniente es que esta señal es del orden de los microvoltios (μV), un nivel muy bajo para procesarlo con comodidad. Por esta razón a la salida del radar, conectaremos un circuito acondicionador (*Figura 2-10)*, que ampliará y recortará esta señal hasta que sea fácilmente procesable por el microcontrolador (entre 0 y 5V) [10] [11].

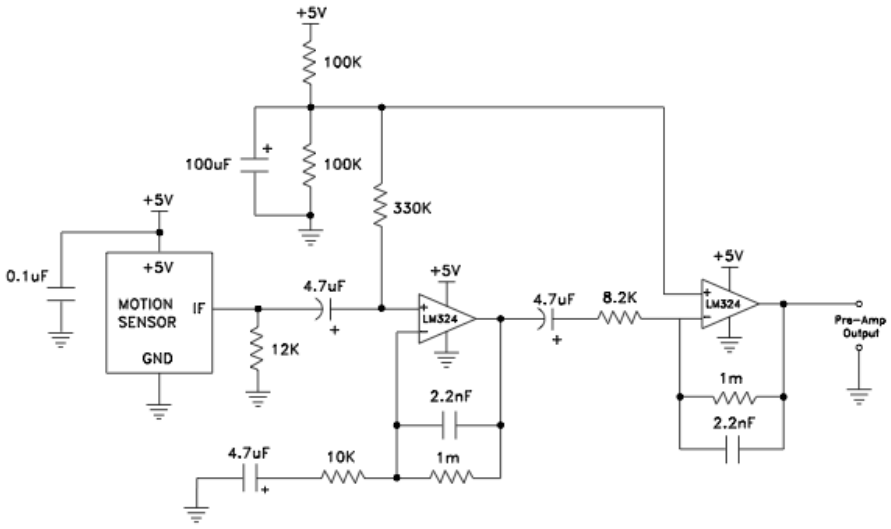


Figura 2-10 Circuito acondicionador para modo onda continua [11]

2.3 Arduino

2.3.1 Introducción y orígenes históricos

El Arduino es un pequeño sistema de procesamiento, una placa programable. Algunos lo definen como una plataforma de hardware libre; la cual consiste, básicamente, en una placa formada por un microcontrolador y una serie de entradas y salidas configurables. El software del Arduino es un entorno de trabajo en el cual, mediante el uso de un código de programación podemos decirle al microcontrolador cómo debe procesar la información y cómo debe actuar en función de la misma. En resumen, es un pequeño ordenador capaz de leer información de diferentes sensores y capaz de controlar luces, motores y muchas otras cosas [16] [17] [18].

Figura 2-11 Logotipo Arduino

El Arduino es una herramienta relativamente moderna. El proyecto se inició en el año 2005 en el instituto de Ivrea, en Italia. Por aquel entonces, un microcontrolador costaba alrededor de los $100, un precio demasiado elevado y prácticamente inaccesible para la mayoría de los estudiantes [18].

Massimo Banzi, estudiante del instituto de Ivrea, con el objetivo de crear una herramienta de aprendizaje en el campo de la computación y la electrónica inició como proyecto, la creación de una placa electrónica con la cual no solo economizó la creación de proyectos escolares, sino que también ayudó a evitar la quiebra de la escuela con los beneficios que obtenía al vender cada placa [18].

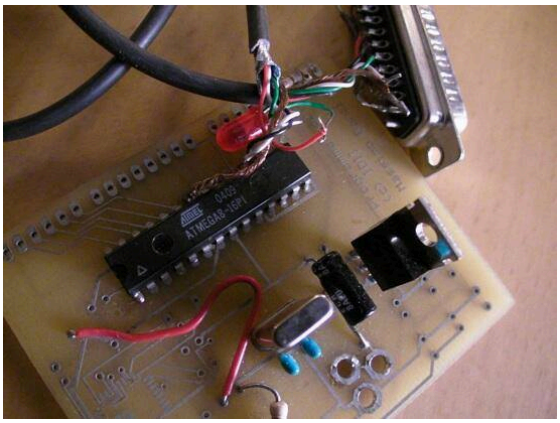


Figura 2-12 Una de las primeras placas de Arduino

Inicialmente, como se puede ver en la *figura 2-12*, esta placa de circuitos eléctricos estaba formada por un microcontrolador simple y algunos componentes discretos y ni siquiera contaba con un entorno de programación para manipularla [18].

Años más tarde se incorporó al proyecto Hernándo Barragán, estudiante de la universidad de Colombia que, junto a David Mellis, contribuyó al desarrollo de un entorno para la programación del procesador de esta placa [18].

El estudiante español, David Cuartielles, se integró más tarde al "equipo Arduino", ayudando a mejorar la interfaz hardware de la placa. Añadió los microcontroladores necesarios para dar soporte al proceso de programación. Con este proceso de programación se podía introducir en la placa códigos de instrucciones para controlar el funcionamiento de la plataforma [18].

Poco después, un estudiante estadounidense llamado Tom Igoe se interesó por el proyecto visitando las instalaciones en el instituto de Ivrea. Una vez de vuelta en su tierra natal recibió una invitación del fundador de la placa para unirse al equipo y mejorar así Arduino. Consiguieron hacerlo más potente, y le añadieron un puerto USB para poder conectarlo a un ordenador [18].

Comenzaron entonces a distribuir el producto a nivel mundial. Empezando por Italia y después por España, hasta que se ha convertido en una de las mejores herramientas de aprendizaje para el desarrollo de sistemas automáticos [18].

Hoy en día Arduino cuenta con varios modelos de placas a elegir, según las necesidades del usuario. Cada una tiene unas características específicas. En la *figura 2-13* se muestran algunos de los tipos de placas Arduino más comunes [19].

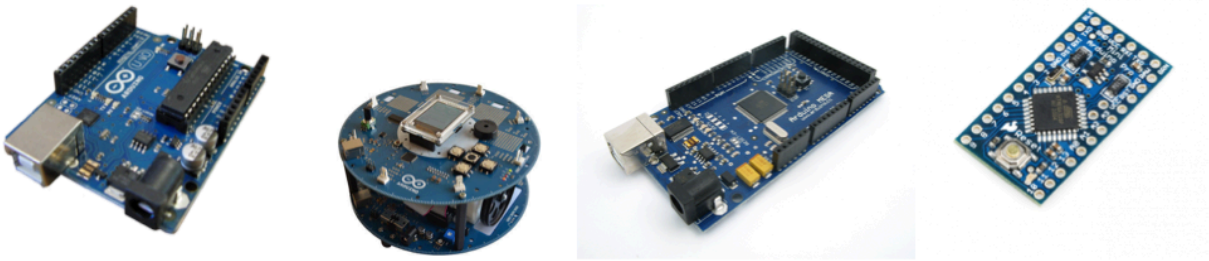


Figura 2-13 Modelos de placa Arduino (De izda. a dcha. 1-Uno 2-Robot 3-Mega 4-Mini)

Además de estos cuatro ejemplos, Arduino cuenta con muchos más tipos de placas así como con diferentes accesorios entre ellos *Shields,* Kits de trabajo, pantallas LCD (*Liquid Crystal Display)*, etc. El modelo con el que se trabaja en este proyecto es el Arduino Uno (*Figura 2-13*).

2.3.2 El Hardware de Arduino

El hardware del Arduino se basa en tres elementos fundamentales, los puertos de entrada o *inputs,* los puertos de salida o *outputs,* y el microcontrolador, que es básicamente el "cerebro" de Arduino, el que analiza la información recibida de las entradas, y da ordenes a las salidas. En concreto el Arduino Uno, cuenta con 14 puertos digitales que se pueden configurar tanto de entrada como de salida; 6 entradas y 6 salidas analógicas y un microcontrolador ATmega328 [16] [20].

El microcontrolador ATmega328 (*Figura 2-14*) es un microcontrolador simple, de bajo consumo y bajo coste. Es muy utilizado en múltiples proyectos y sistemas autónomos. Tiene una memoria de 32 KB para almacenar código y una velocidad de reloj de 20 MHz [21].

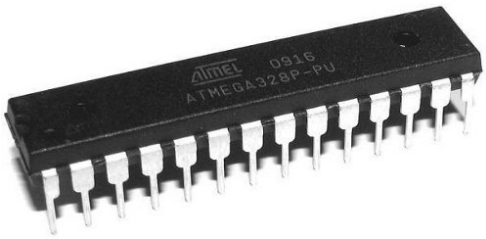


Figura 2-14 Microcontrolador ATmega 328

El microcontrolador gestiona las entradas y las salidas tanto digitales como analógicas. En las salidas digitales el microcontrolador asigna un valor "HIGH" (5V) o "LOW" (0V). En las entradas lee si el pin está "HIGH" o "LOW". En las entradas analógicas el microcontrolador lee el valor de tensión del pin especificado. Arduino posee 6 canales conectados a un conversor analógico digital de 10 bits. Esto quiere decir que convertirá el valor de tensión (entre 0V y 5V) en un número del 0 al 1023. En las salidas analógicas el microcontrolador "escribe" un valor que va del 0 al 255. Esto puede ser usado para controlar la luminosidad de un LED por ejemplo [22].

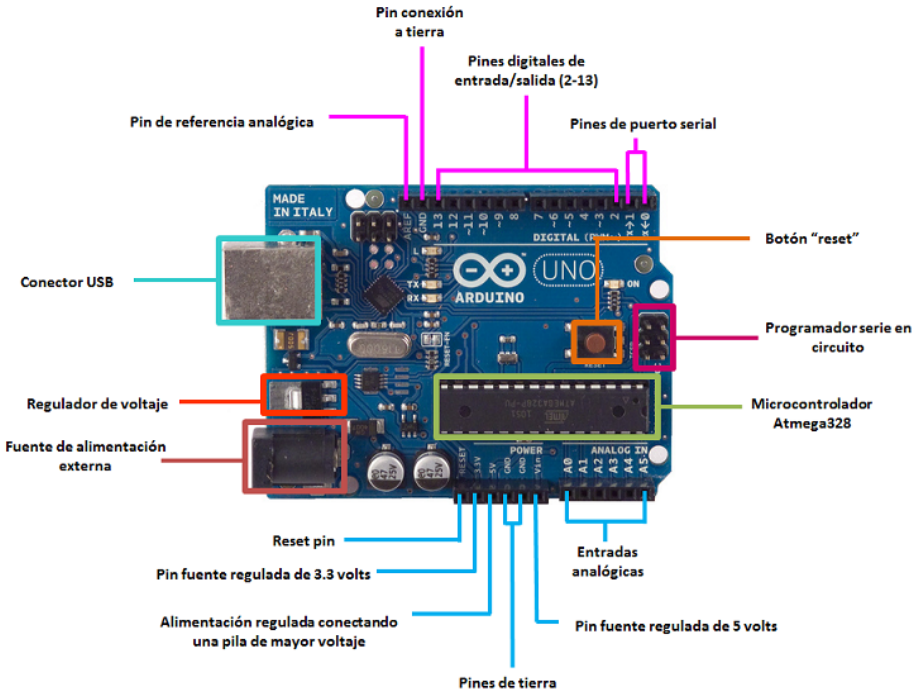


Figura 2-15 Explicación de los elementos de la placa Arduino UNO

En resumen, la diferencia entre los puertos digitales y los analógicos es que los digitales solo dan como información: "el pin está activado" o "el pin está desactivado"; mientras que los analógicos pueden dar un valor numérico. Por ejemplo, un pulsador, que solo tiene dos posiciones (pulsado, no pulsado) se conectaría a una entrada digital; en cambio un sensor de temperatura, que puede obtener un amplio rango de valores se conectaría a una entrada analógica.

El microcontrolador de la placa Arduino aparte de controlar pines digitales y analógicos puede comunicarse con un ordenador mediante la "comunicación en serie". Esto permite al usuario leer, por ejemplo, los valores de un sensor conectado a Arduino. El intercambio de información se hace a través de un puerto USB. Este puerto, se usa además para cargar código al microcontrolador y para dar alimentación a la placa [20] [22].

La placa también puede ser alimentada por otra fuente externa aparte del USB. En ese caso, puede trabajar con alimentación de entre 6 y 20 Voltios. Sin embargo, si se alimenta con menos de 7 V, los puertos de salida pueden dar menos de 5 V y la placa puede ser inestable. Si se alimenta con más de 12 V el regulador de voltaje podría sobrecalentarse y dañar la placa. Por esto, se recomienda alimentar la placa en un margen de entre 7 y 12 voltios [20].

La placa alimenta los puertos de salida con un voltaje de 5 voltios. Cada puerto I/O puede dar o recibir un máximo de 40mA. Para alimentar el circuito que queramos conectar cuenta con una salida de 3.3 V y otra de 5 V, así como tres tomas a tierra [20].

2.3.3 Entorno de desarrollo

El software del Arduino es un programa informático llamado IDE (*Integrated Development Enviromentent*), o entorno de desarrollo integrado (*Figura 2-16)*. El IDE permite al usuario escribir un código, compilarlo (convertirlo en un código binario) y cargarlo en la placa del Arduino. Este código

es una secuencia de instrucciones ejecutables por el microcontrolador. Los códigos se escriben usando un leguaje de programación tal como C, Fortran o BASIC. Arduino utiliza un lenguaje similar al C [20] [23] [24].

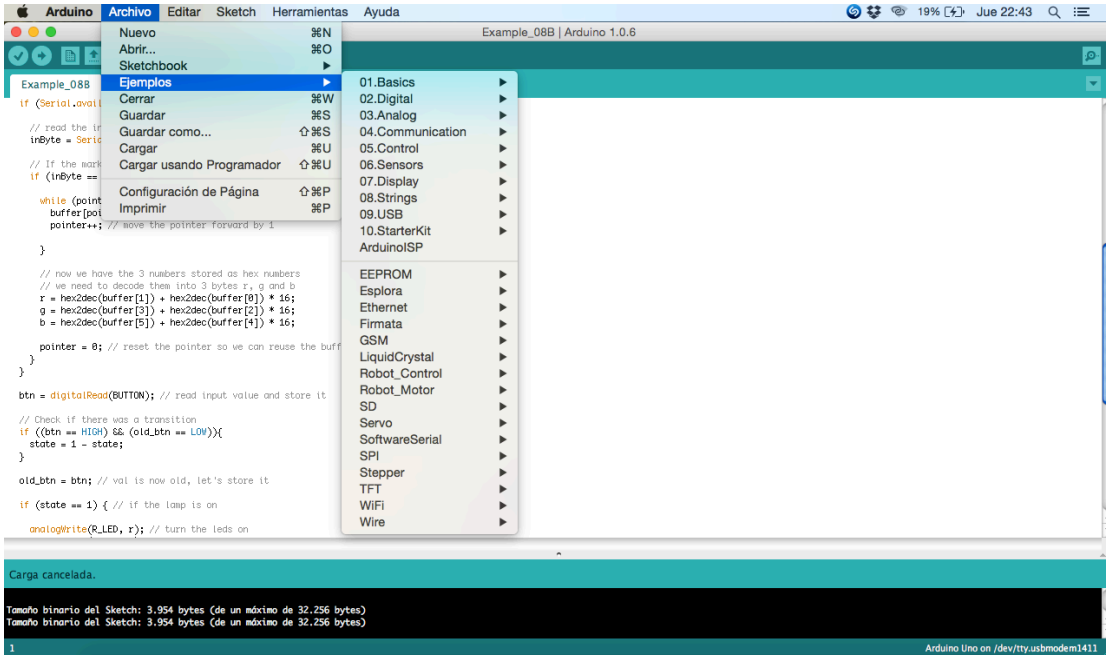


Figura 2-16 Entorno de desarrollo de Arduino

La estructura general del lenguaje de programación de Arduino es la siguiente: Primero se incluye una zona de declaración de variables. En esta parte del texto se declaran todas las variables que se van a usar en el código. Existen diferentes tipos de variables. La más típica es la definida por la palabra reservada *int,* que ocupa 2 bytes de memoria y puede abarcar valores entre -32,768 y +32,768. Una vez definidas las variables, mediante la función *setup* se configuran los pines de la placa como entradas o salidas. Esta función solo se ejecuta una vez. A continuación usando la función *loop* comenzamos el bucle de instrucciones que será ejecutado por el microcontrolador cíclicamente [23].

Para personalizar este bucle de instrucciones existen numerosos elementos tales como estructuras de control (*if, if-else, while, ...)* temporizadores (*delay, millis, ...)* instrucciones para los pines digitales (*pinMode, digitalRead, digitalWrite, ...)* instrucciones para los pines analógicos (*analogWrite, analogRead, ...),* etc [23] [24].

Estos son algunos ejemplos de comandos del lenguaje que utiliza Arduino para hacer diferentes programas. Todo el lenguaje de Arduino está recogido en la página web del mismo, en el *Arduino Programming Notebook* [25] [26].

En el IDE de Arduino se pueden encontrar librerías. Las librerías ayudan a añadir funcionalidad extra a los códigos de programación a la hora de trabajar con hardware o manipular datos. Algunas librerías vienen descargadas ya en el IDE, pero también se pueden crear o descargar [26]. En este proyecto por ejemplo se hará uso de una librería para convertir la señal que el Arduino recibe del radar (en voltios) a frecuencia y velocidad.

Según [17], el funcionamiento del Arduino consiste en tres pasos fundamentales:

- Entrada de la señal: Los sensores, conectados al Arduino a través de los puertos de entrada, envían a éste la señal que detectan del exterior.

- Procesamiento de la señal: El microcontrolador por un lado lee esta señal y la analiza en función del código con el que se haya programado. Por otro lado ordena a través de los puertos de salida realizar diferentes acciones a los actuadores, también en función del código.

- Señal de salida: Por los puertos de salida pasan las órdenes del microcontrolador y llegan a los actuadores.

2.3.4 Ejemplo de uso

Para entender más fácilmente cómo funciona el Arduino se explica a continuación un ejemplo de uso sencillo.

Se conecta a la entrada del Arduino un pulsador, y a la salida un LED, como se puede ver en la *figura 2-17*. Se carga en la placa del Arduino, a través del puerto USB un programa sencillo (véase *figura 2-18*) en el cual se le dice al microcontrolador que lea la señal del pulsador y que encienda el LED solo si el pulsador está siendo pulsado:

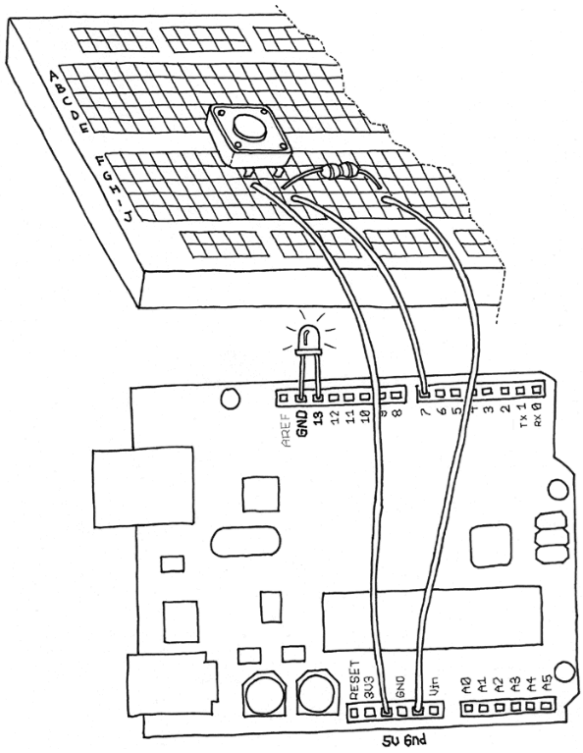


Figura 2-17 Esquema de conexiones del ejemplo de uso

```
const int LED = 13;   // the pin for the LED
const int BUTTON = 7; // the input pin where the
                      // pushbutton is connected

int val = 0; // val will be used to store the state

             // of the input pin
void setup() {
  pinMode(LED, OUTPUT); // tell Arduino LED is an output
  pinMode(BUTTON, INPUT); // and BUTTON is an input
}
void loop(){
  val = digitalRead(BUTTON); // read input value and store it

  // check whether the input is HIGH (button pressed)
  if (val == HIGH) {
    digitalWrite(LED, HIGH); // turn LED ON
  }
  else {
    digitalWrite(LED, LOW);
  }
}
```

Figura 2-18 Programa utilizado en el ejemplo de uso

El microcontrolador ejecuta una y otra vez el código incluido en la función *loop*. Mientras el pulsador no esté pulsado, no mandará al LED la orden de encender. Cuando el pulsador es accionado, a través del puerto de entrada le llega al microcontrolador esta señal. Éste la recibe y manda la orden de encenderse al puerto de salida. El LED entonces se enciende. Cuando se deja de pulsar el pulsador, por el mismo procedimiento, el LED se apaga.

Éste es un ejemplo muy simple, pero el Arduino es capaz de cosas mucho más complejas. En el caso de este proyecto la señal de entrada será detectada por el radar HB100. El microcontrolador procesará esta información y generará la señal de salida, que será la por un lado una señal indicado si se ha detectado o no un blanco, y por otro, la velocidad del blanco.

3 DESARROLLO DEL TFG

3.1 Etapas de desarrollo del TFG

En el desarrollo de este Trabajo de Fin de Grado se pueden diferenciar cinco etapas fundamentales:

1) Etapa de documentación: Etapa de búsqueda y lectura de información sobre el módulo sensor HB100, y sobre el microcontrolador Arduino.

2) Etapa de análisis del circuito electrónico de acondicionamiento de la señal de salida del sensor, por medio de simulación con el software *PSim*.[2]

3) Montaje del circuito de acondicionamiento en los laboratorios de electrónica. Caracterización de la señal de salida del circuito por medio del uso de un osciloscopio.

4) Conectar el módulo radar con el circuito de acondicionamiento al microcontrolador Arduino para procesar la señal de salida y ser capaces de:

   - Tomar decisiones sobre la detección o no detección de un blanco en movimiento.
   - Estimar la velocidad del blanco a partir de la frecuencia Doppler.

5) Independizar el montaje del PC utilizando una batería que alimenta a la placa Arduino y una pantalla LCD (*Lyquid Cristal Display*) para mostrar la información.

3.1.1 Etapa de Documentación

Durante esta etapa de documentación los objetivos fundamentales eran: informarse sobre las características y usos del módulo radar HB100 y llegar a conocer la plataforma Arduino, sus usos y aplicaciones.

Por un lado, para conocer el módulo radar se utilizó la *"Hoja de Especificaciones técnicas del módulo radar HB100"* [10] y la "*nota de aplicación"* [11] del mismo. En estos dos documentos están recogidas las características del sensor y los diferentes modos de aplicación del mismo.

[2] *PSim* es un *software* informático que permite al usuario simular circuitos electrónicos para su análisis.

Por otro lado, era necesario familiarizarse con el Arduino, y la mejor manera para ello era utilizándolo. La plataforma Arduino cuenta con unos tutoriales de uso para empezar de cero. Haciendo uso del libro "*Getting Started with Arduino*" [24] y del kit de aprendizaje (*figura 3-1),* se consiguió conocer esta plataforma y sobre todo se aprendió a utilizarla.

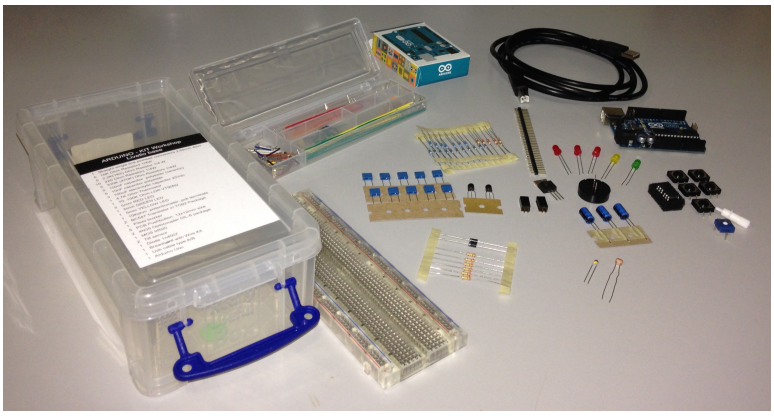


Figura 3-1 Kit de aprendizaje de Arduino

Conseguidos estos dos objetivos fundamentales, durante esta etapa se aprovechó para profundizar más acerca de la historia, tipos y funcionamiento de los radares, y de la historia, hardware y software de la plataforma Arduino. Esta documentación queda recogida en el apartado de "*Estado del Arte"*. La herramienta principal que se usó para ello fue internet. Todas las páginas visitadas están referenciadas en el apartado de "*Bibliografía"*.

3.1.2 Etapa de análisis del circuito electrónico de acondicionamiento.

Durante la etapa de documentación se llegó a la conclusión de que el módulo radar enviaba una señal muy pequeña y que, para que el Arduino pudiera procesarla bien, era necesario amplificarla primero.

Para ello, había que construir un circuito electrónico capaz de acondicionar la señal del radar para que fuera fácilmente procesable por el microcontrolador Arduino. La "*Nota de Aplicación"* [11] del módulo ofrece un circuito amplificador para ello como se vio en el apartado de *Estado del Arte* (*figura 2-10)*.

Pero antes de montar el circuito acondicionador físicamente en el laboratorio, se procedió a realizar un análisis del mismo, haciendo uso del programa informático *PSim* para comprobar sus características.

Lo primero que se hizo fue representar en el *PSim* el circuito de acondicionamiento tal y como se recomienda en la "*Nota de Aplicación"* [11], cambiando el módulo radar por un generador de corriente alterna para poder simular una señal de entrada en el circuito, como se puede observar en la *figura 3-2.*

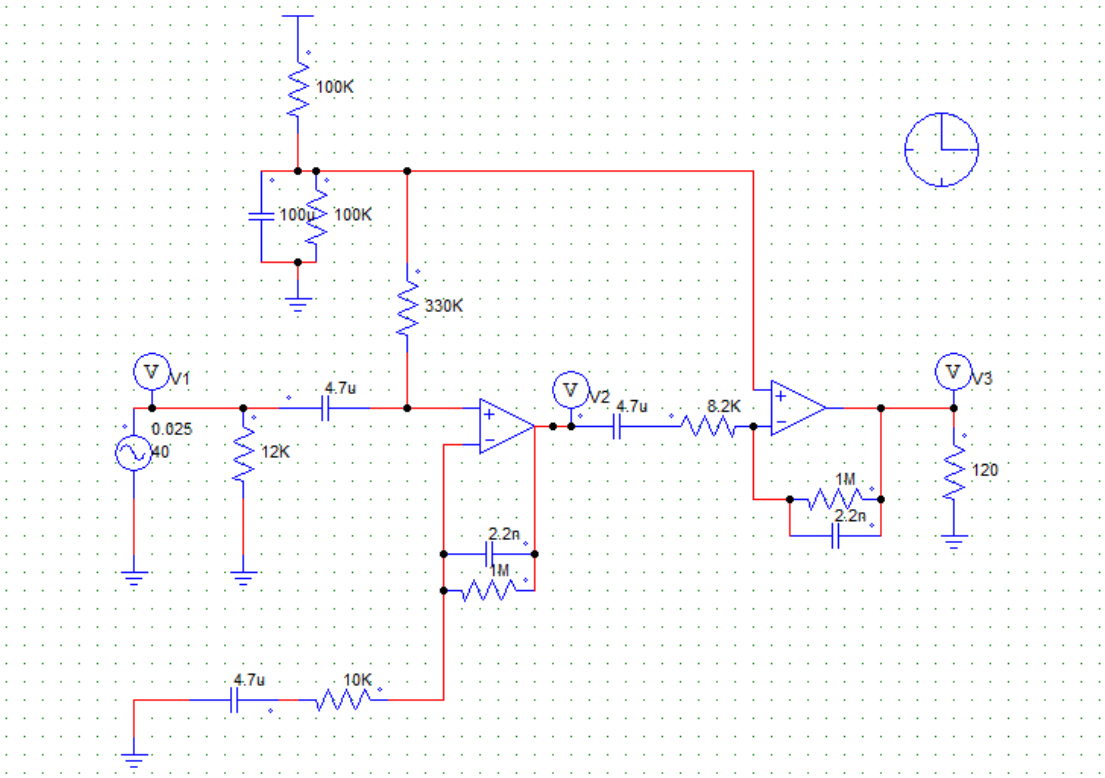


Figura 3-2 Representación del circuito de acondicionamiento en *PSim*

Para medir la señal en cada una de las diferentes etapas del circuito acondicionador, se colocaron tres voltímetros situados en diferentes puntos (ver *figura 3-2*):

- V1: A la salida del generador de corriente alterna, para ver que realmente genera la señal que se quiere introducir en el circuito. Los resultados de esta simulación se pueden observar en la *figura 3-3*. Donde se puede ver que la señal de entrada es sinusoidal con amplitud 0,05 V y frecuencia 40Hz.

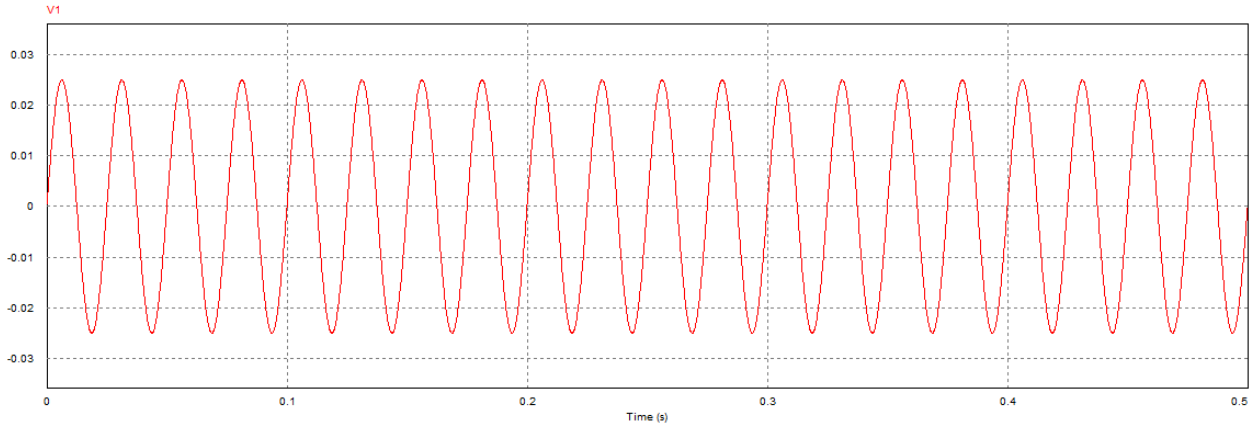


Figura 3-3 Señal a la salida del generador de corriente alterna (simulado en el *PSim*)

- V2: A la salida del primer amplificador, para comprobar si la primera etapa del acondicionador responde correctamente. Los resultados de esta simulación se pueden observar en la *figura 3-4*. Donde se puede ver que se corresponde con una señal sinusoidal amplificada y con la misma frecuencia (40 Hz).

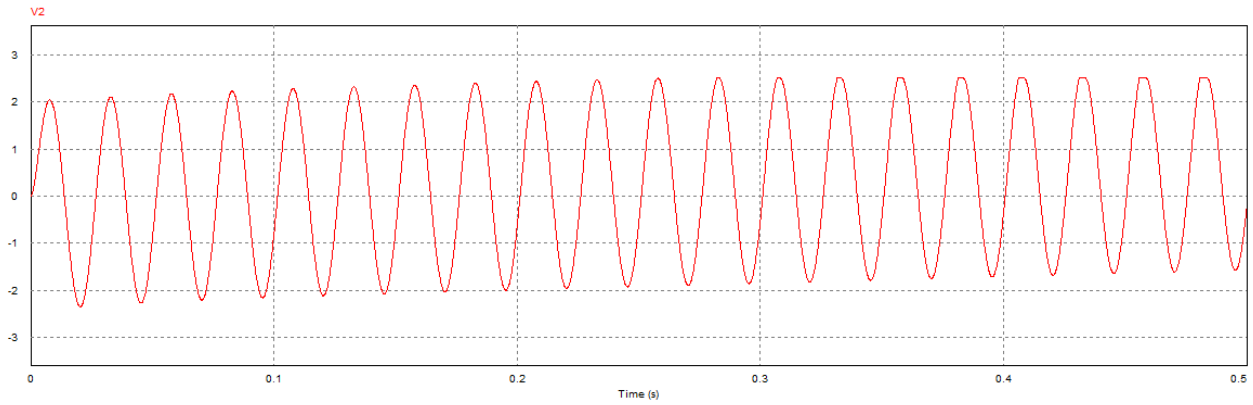


Figura 3-4 Señal a la salida de la primera etapa de acondicionamiento (simulado en el *PSim*)

- V3: A la salida final del circuito, para ver que la señal queda adecuadamente amplificada y recortada al salir del circuito. Los resultados de esta simulación se pueden observar en la *figura 3-5*. Donde se observa que la señal de salida es amplificada de nuevo y además está recortada a una amplitud máxima de 5 V.

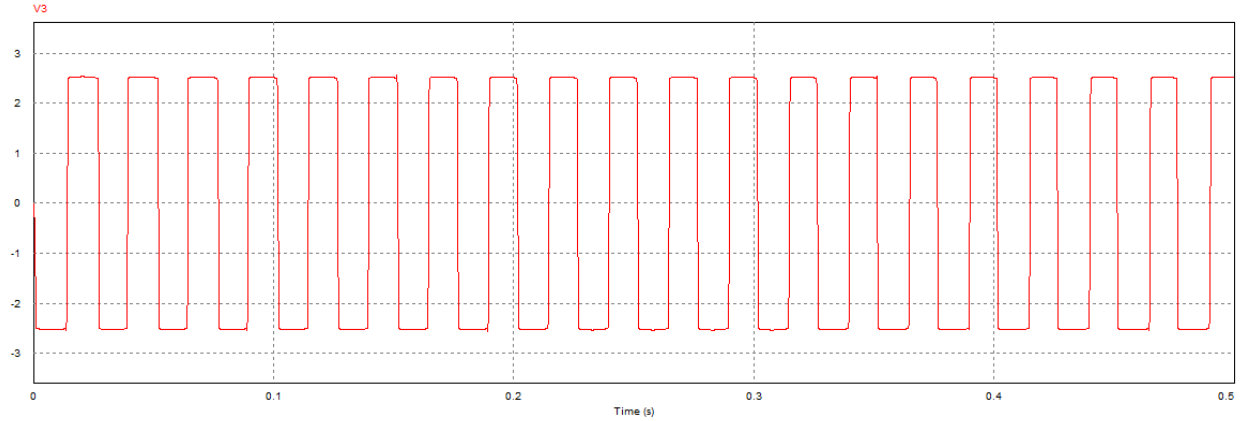


Figura 3-5 Señal a la salida del circuito de acondicionamiento (simulado en el *PSim*)

Vistos los resultados obtenidos en la simulación del circuito en el *PSim*, se observó que el circuito acondicionador amplifica y recorta la señal hasta un nivel adecuado para el microcontrolador Arduino. Por lo tanto, a continuación se procedió a realizar el montaje del mismo.

3.1.3 Montaje del circuito de acondicionamiento.

Después de analizar el circuito, se procedió al montaje del mismo. Para ello se utilizó el laboratorio de física ubicado en el edificio Isaac Peral de la E.N.M.

Antes de montar el circuito amplificador, se conectó el radar a la alimentación para medir la señal del mismo (sin amplificar) con un osciloscopio y así comprobar que efectivamente la señal tenía una amplitud de valor muy reducido, del orden de los microvoltios.

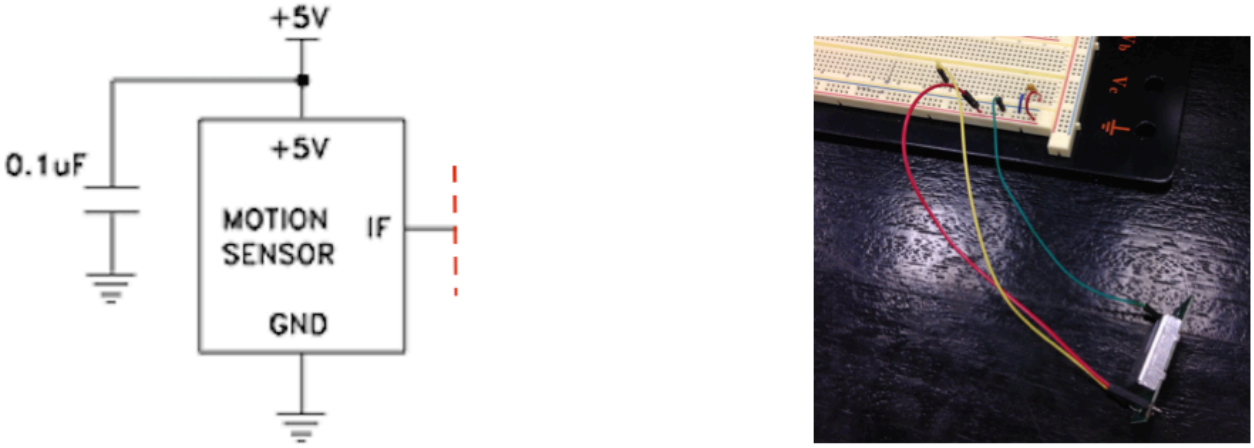


Figura 3-6 Sensor HB100 conectado a alimentación (Izda: Esquema - Dcha: Real)

Para hacer el montaje del circuito, debido a que se trata de un circuito con bastantes elementos, se decidió hacerlo paso a paso para poder analizarlo por etapas, utilizando una *protoboard*[3] por cada una de las ellas. De esta manera, en el caso de cometer un error, sería mucho más fácil de detectar y de corregir.

Para empezar se montó la primera parte del circuito, que no incluye ninguna etapa de amplificación. Simplemente un condensador de 0,1uF conectado entre la alimentación del radar y tierra; y en la salida de IF del módulo radar, por un lado una resistencia de 12 KΩ a tierra y por otro un condensador electrolítico de 4.7 uF por el que pasaría la señal antes de llegar a la primera etapa del circuito acondicionador.

[3] Una placa de pruebas (en inglés: *protoboard* o *breadboard*) es un tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares [30].

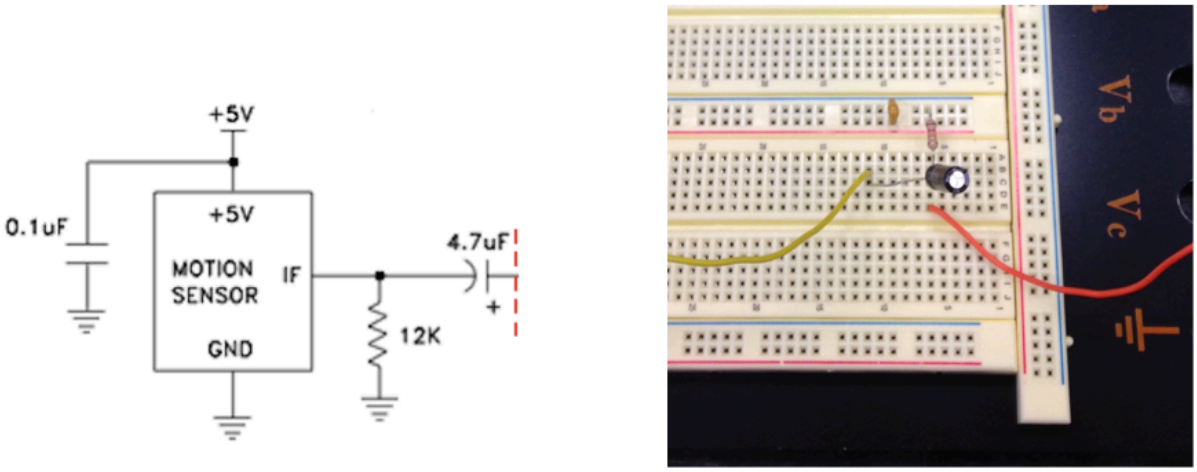


Figura 3-7 Primera etapa del circuito acondicionador (Izda: Esquema - Dcha: Real)

Una vez montada esta parte y comprobado que a la salida del condensador electrolítico llegaba la señal, se procedió al montaje de la segunda etapa del circuito, es decir, la primera etapa del circuito acondicionador. Esto se hizo usando otra *protoboard* aparte para conseguir mayor claridad durante el montaje. Conectando la salida de la etapa anterior con la entrada positiva del *amplificador LM324*[4], y uniendo la entrada negativa y la salida por medio de un condensador de 2.2 nF y una resistencia de 1 MΩ. A su vez la entrada negativa del amplificador se conecta a tierra a través de una resistencia de 10 KΩ con un condensador electrolítico de 4.7 uF en serie. El amplificador se conecta, por un lado a alimentación de 5V y por otro a tierra. La salida del amplificador lleva la señal, ya amplificada, a la siguiente etapa a través de un condensador electrolítico de 4.7 uF en serie con una resistencia de 8.2 KΩ.

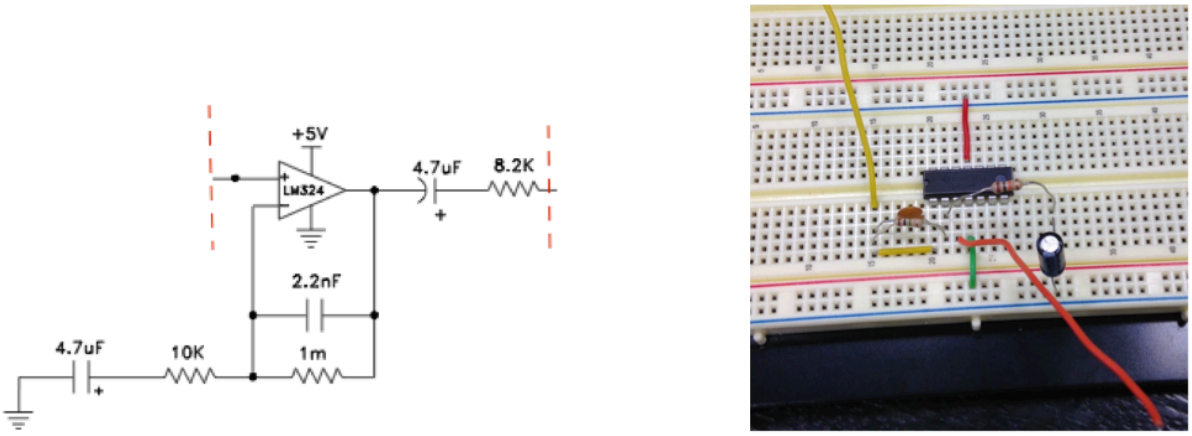


Figura 3-8 Segunda etapa del circuito de acondicionamiento (Izda: Esquema - Dcha: Real)

Por último y de nuevo haciendo uso de otra *protoboard* se realizó el montaje de la última etapa. En esta etapa del circuito se conecta una resistencia de 100 KΩ por un lado a alimentación de 5 V y por

[4] Se trata de un dispositivo electrónico que tiene dos entradas y una salida. La salida es la diferencia de las dos entradas multiplicada por un factor (G) que es la ganancia de amplificación.

otro lado dos conexiones: una a tierra, a través de un condensador de 100 uF y una resistencia de 100 KΩ en paralelo; y otra que va, por un lado a la entrada positiva del primer amplificador a través de una resistencia de 330 KΩ, y por otro a la entrada positiva del segundo amplificador, elemento conectado a 5 V y tierra para ser alimentado. Esto se hace para limitar la amplitud de la señal amplificada a 5 V, ya que es la amplitud máxima que debe recibir el microcontrolador. La salida de la primera etapa se conecta a la entrada negativa del segundo amplificador. Esta entrada negativa se une con la salida de este amplificador a través de una resistencia de 1 MΩ y un condensador cerámico de 2.2 nF conectados en paralelo. La señal quedará amplificada y recortada sin verse afectada la frecuencia de la misma, que es el dato imprescindible para calcular la velocidad.

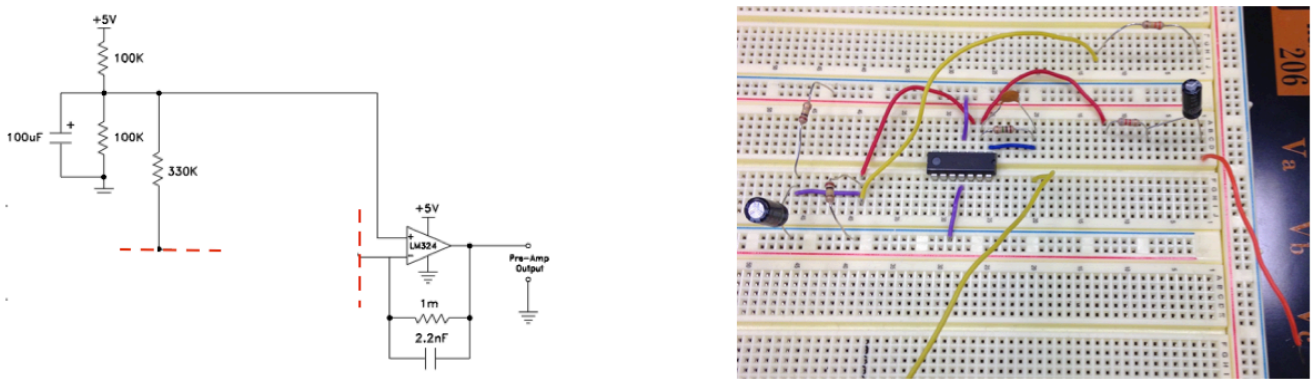


Figura 3-9 Tercera etapa del circuito de acondicionamiento (Izda: Esquema - Dcha: Real)

3.1.4 Verificación del circuito de acondicionamiento

Una vez montada la segunda etapa del circuito acondicionador se procedió a unir cada una de las partes y comprobar su funcionamiento. Para ello, a la entrada de la primera etapa, en lugar de conectar el radar, se utilizó un generador de señales para crear una señal sinusoidal con una amplitud de 0.05 V y una frecuencia de 100.00 Hz.

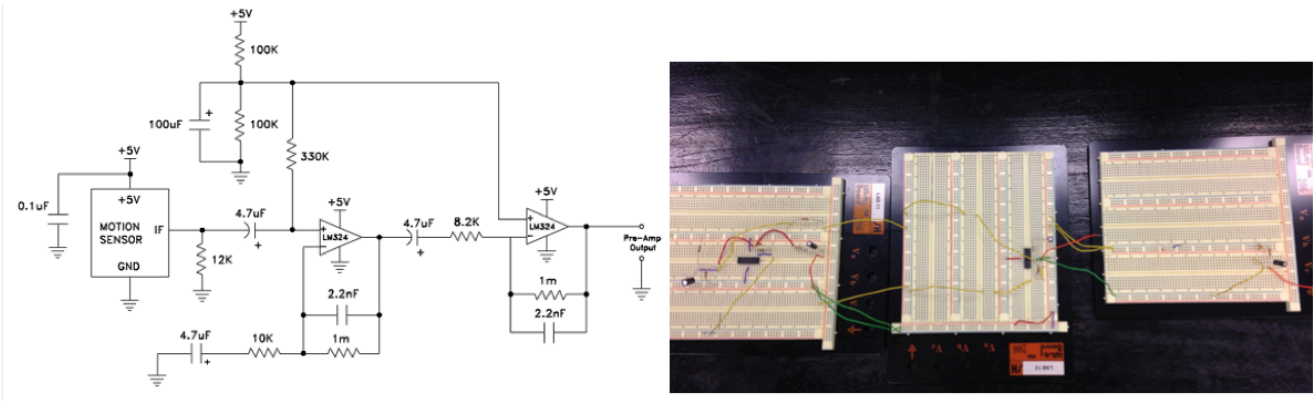


Figura 3-10 Circuito de acondicionamiento completo (Izda: Esquema - Dcha: Real por etapas)

Alimentando el circuito con 5 V y metiendo en la entrada del circuito la señal generada (*figura 3-11*), haciendo uso de un osciloscopio se tomaron mediciones en tres puntos fundamentalmente:



Figura 3-11 Izda: Alimentación de 5 V - Dcha: Señal generada de amplitud 0,05 V y frecuencia 100,00 Hz

1. En la entrada del primer amplificador, para comprobar que la señal introducida en el circuito era la generada *(figura 3-12)*.

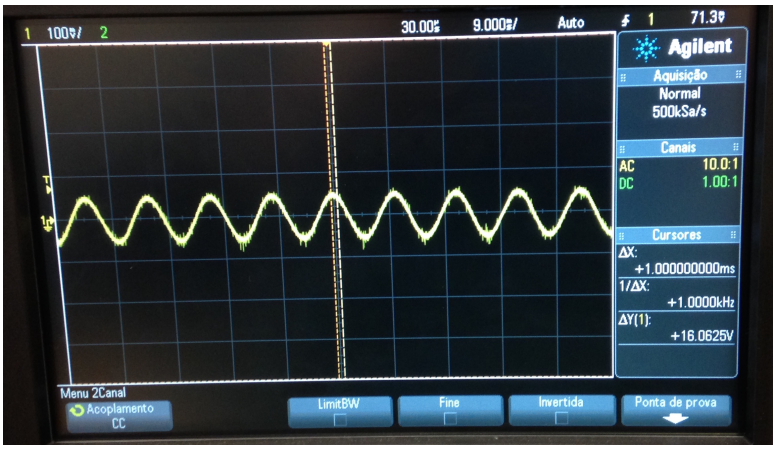


Figura 3-12 Señal generada medida con osciloscopio

2. En la salida del primer amplificador para comprobar el correcto funcionamiento de la primera etapa de amplificación *(figura 3-13)*.

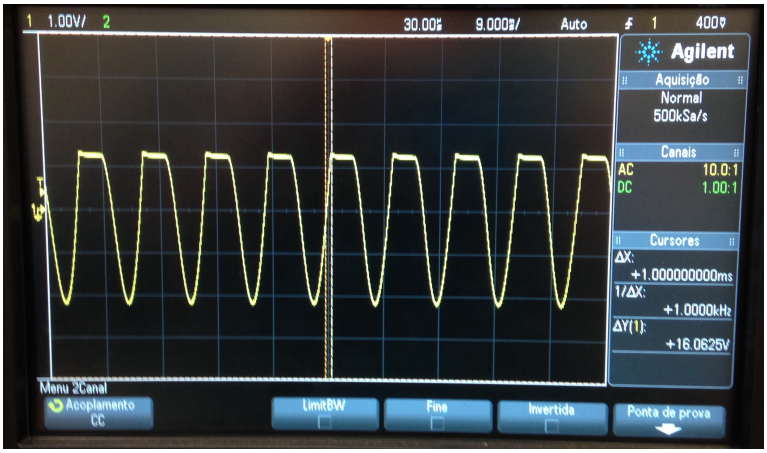


Figura 3-13 Señal generada medida con osciloscopio a la salida de la primera etapa

3. Por último en la salida del segundo amplificador, comprobando que la señal era recortada de manera correcta además de amplificada *(figura 3-14)*.

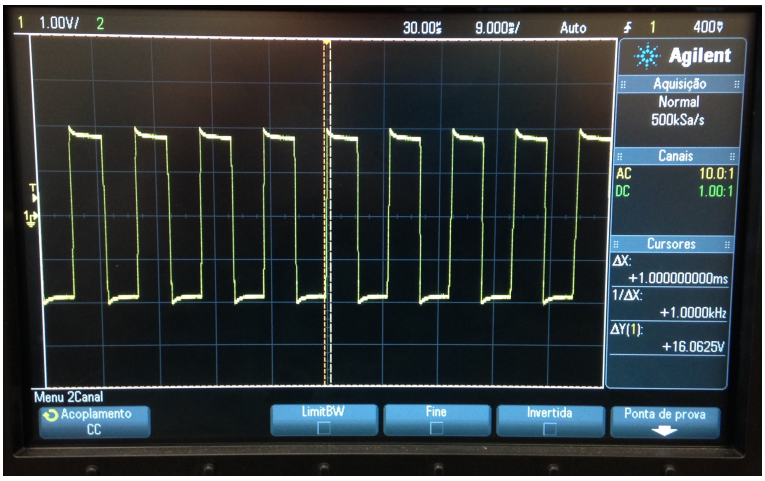


Figura 3-14 Señal generada medida con osciloscopio a la salida final del circuito

Una vez comprobado de esta manera que el circuito funcionaba bien, se procedió a realizar una prueba con el radar. Usando el osciloscopio para tomar medida a la salida final del circuito, moviendo la mano delante del sensor radar, se vio que la señal del radar era correctamente amplificada y recortada. (*figura 3-15)*

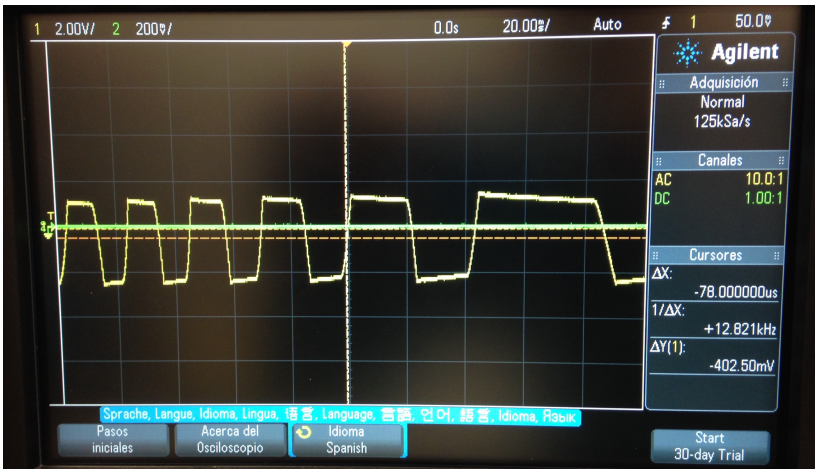


Figura 3-15 Señal del radar medida con osciloscopio a la salida final del circuito

Viendo el correcto funcionamiento del circuito acondicionador, tanto con una señal generada como con la señal del propio radar, se procedió a realizar un montaje más eficiente, que fuera menos aparatoso, utilizando únicamente una *protoboard*. El resultado final se muestra en la *figura 3-16.* Después de realizar este montaje, se comprobó su correcto funcionamiento de la misma manera que se hizo con el circuito montado por etapas.

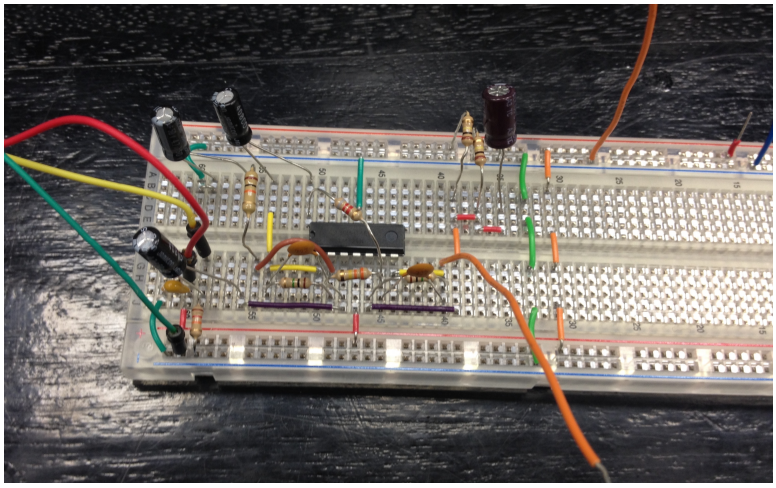


Figura 3-16 Circuito acondicionador montado sobre una única *protoboard*

3.1.5 Conexión del circuito con el microcontrolador Arduino

Hecho esto, ya sólo quedaba mandar la señal de salida del circuito acondicionador al microcontrolador para que la procesara.

Para conseguirlo, por un lado era necesario, en primer lugar, no depender de los equipos de alimentación del laboratorio para alimentar el circuito acondicionador y el módulo radar y, por otro lado, había que enviar la señal de la salida del circuito a la placa de Arduino.

La placa de Arduino Uno que se ha utilizado en este proyecto, como se ha visto anteriormente en el apartado de "*Estado del Arte"*, cuenta con un pin de alimentación de 5V, tres tomas a tierra y varios pines configurables como entradas o salidas según la necesidad.

Por lo tanto esta parte del proceso no tuvo mayor complicación. Se utilizó la salida de 5V y una de las tomas a tierra para alimentar la *protoboard* donde estaba montado el circuito amplificador y donde también estaba conectado el radar HB100. Para mandar la señal del radar amplificada al Arduino, sólo hubo que conectar la salida IF del módulo a la entrada del circuito, y la salida del circuito a uno de los pines de la placa que más adelante se configuraría como entrada digital.

Una vez conseguido que la placa de Arduino esté recibiendo la señal del radar con una amplitud cómoda para ser procesada, quedaba la parte complicada de esta etapa del desarrollo: Procesar la señal para extraer de ella la información necesaria para, por un lado, decidir la detección/no detección de movimiento y, por otro lado, obtener el valor de la velocidad del blanco detectado.

Para hacerlo, se utilizó una librería llamada "*FreqPeriod",* creada por Martin Nawrath, y colgada en la página web *"Laboratory for Experimental Computer Science at the Academy of Media Arts Cologne"* [27] en el año 2012 . Esta librería se creó para medir frecuencias por debajo de los 20 KHz, un margen muy adecuado para el presente proyecto, teniendo en cuenta que ese límite superior de frecuencia se correspondería con una velocidad de 1026.17 Km/h.

Haciendo uso de esta librería, y tomando como base el código recomendado en la página web antes mencionada, se implementaron tres programas diferentes:

- Decisión detección/no detección: En este programa, el sistema lo que hace es ver si hay presencia de movimiento y, en el caso de que la haya durante 0.6 segundos, enciende un LED a modo de aviso. A su vez, envía al ordenador tres valores de velocidad tomados durante los 0.6 segundos. Para establecer el umbral de frecuencia a partir de la cual se considera presencia de movimiento, se realizó un experimento de observación que se explica en el apartado 4 del presente proyecto. La idea de este código es simular un sistema automático para abrir una puerta o encender la luz de un pasillo.

- Obtener velocidad promedio: En este programa, el sistema lo que hace es tomar 10 valores de velocidad a lo largo de un segundo. Tomados estos valores, calcula el valor promedio y lo manda al ordenador a través de un puerto serie, para que se muestre en la pantalla. La intención de este código es conseguir una mayor precisión en la medición de la velocidad de un blanco.

- Velocidad máxima con alarma: En este programa, el sistema toma también 10 valores durante 1 segundo, pero esta vez en lugar de calcular el valor promedio y mandarlo al ordenado, simplemente coge el mayor de ellos y lo manda al ordenador. A su vez, si este valor supera un límite de 15 Km/h (límite modificable en el programa) parpadea un LED a modo de alarma. La idea de este código es simular un radar de policía. El parpadeo del LED puede simular la toma de una fotografía o simplemente una alarma para el policía, un aviso de que alguien ha superado el límite de velocidad permitido.

3.1.6 Independizar el montaje del PC

Llegados a este punto, los objetivos básicos de este TFG estaban cubiertos. Sin embargo, el sistema dependía del PC, tanto para recibir alimentación como para mostrar la información de velocidad.

Siendo así, iba a ser complicado realizar ciertas pruebas, tales como medir la velocidad de un coche en exteriores dado que el sistema era aparatoso. Por eso se decidió independizar el sistema y unirlo todo en un bloque manejable.

Para independizar el sistema del PC había que solucionar dos problemas:

- Alimentación: La alimentación del sistema era proporcionada por el PC a través de un puerto USB. Este problema fue fácil de solucionar al contar con una batería que utiliza el mismo tipo de conexión que el PC. Simplemente se conectó a la entrada USB por donde el PC alimenta el Arduino, una batería fácilmente transportable.



Figura 3-17 Batería con entradas USB



Figura 3-18 Cable USB para conexión de la placa Arduino

- Presentación de la información: Para solucionar este problema se empleó un *display* LCD. Para ello, hacían falta dos cosas: por un lado conectarlo de manera adecuada a la placa de Arduino, y por otro lado implementar el código necesario para enviar la información del Arduino al *display*. La plataforma Arduino cuenta con un tutorial de uso del LCD, el cual fue un buen apoyo para ver las conexiones necesarias y para aprender el mecanismo que utiliza Arduino para mandar la información al *display*.

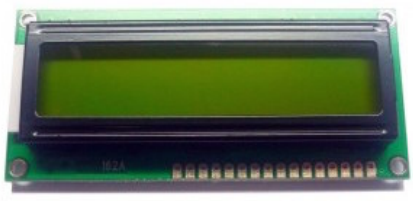


Figura 3-19 *Display* LCD

Una vez independizado del ordenador, el sistema estaba formado por 5 partes: El sensor radar, el circuito acondicionador, el microcontrolador Arduino, la pantalla LCD y por último la batería. Estos elementos estaban únicamente unidos por medio de cables, lo que hacía el sistema completamente inestable cuando se movía.

Para convertirlo en un sistema estable, portátil y fácilmente manejable se decidió ensamblar cada uno de los elementos. Para ello se utilizó una placa donde se ancló el Arduino y el *display* LCD, y se pegó la *protoboard* donde estaba montado el circuito acondicionador y el sensor HB100.

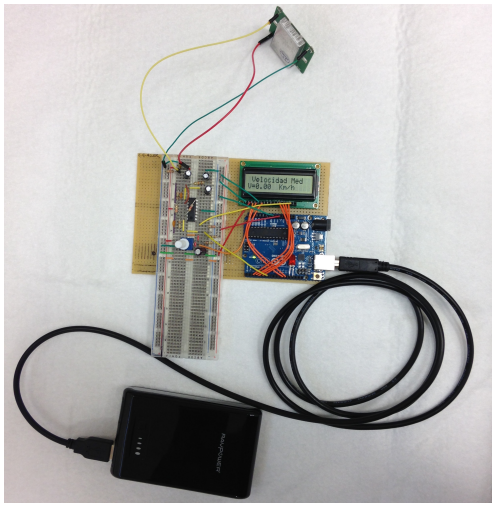


Figura 3-20 Sistema independiente del PC

Para conseguir que el *display* mostrara la información de velocidad procedente de la señal del radar, el Arduino tenía que alimentar la pantalla y a la vez mandarle la información. Como se ha mencionado antes, se hizo uso de los tutoriales con los que cuenta Arduino para aprender cuáles eran las conexiones necesarias y cual era el código que había que incluir en el programa para mandar la información.

Al escribir el nuevo código no sólo se configuró de manera que mandara la información al *display*, sino que también se implementaron pequeñas mejoras en cada uno de los programas.

- Decisión detección/no detección: La mejora de este código (véase Anexo I) consistió en añadir en el *display* LCD un mensaje de bienvenida (“BIENVENIDO A LA E.N.M.) cada vez que se detectaba movimiento, manteniendo el encendido el LED. De esta manera, si el sistema fuera utilizado para abrir una puerta automática, al llegar a la misma se abriría y a su vez daría la bienvenida.

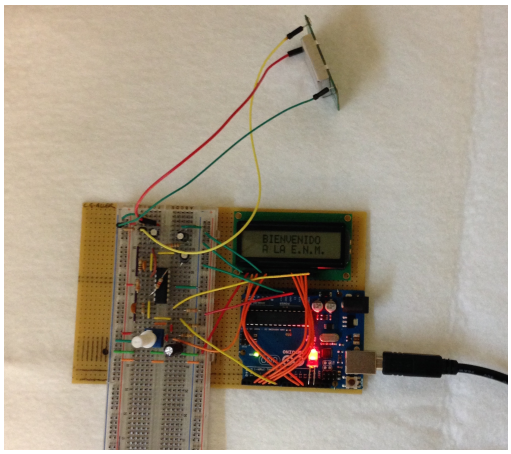


Figura 3-21 Sistema detectando presencia de movimiento

- Obtener velocidad promedio: La modificación de este código (véase Anexo II) consiste en mostrar en el *display* LCD el valor medio y si se desea, se podría conectar el sistema al PC para, a la vez mostrar en la pantalla del ordenador cada una de los diez valores tomados cada segundo. De esta manera se puede comprobar que el valor mostrado en el LCD es realmente el valor promedio de los valores tomados.

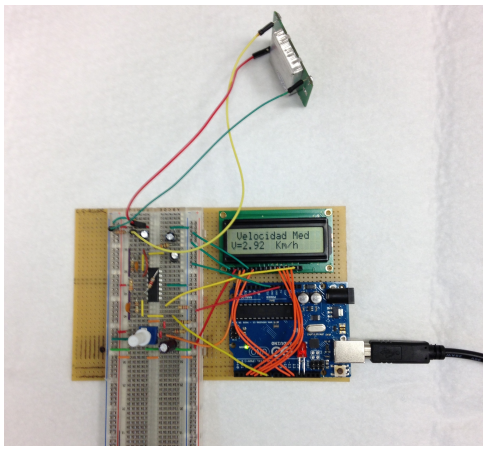


Figura 3-22 Sistema mostrando velocidad media calculada

- Obtener la velocidad máxima: La modificación de este código (recogido en el Anexo III) consiste en mostrar en el *display* LCD el valor de velocidad máximo de los diez tomados en cada segundo y, si se desea, se podría conectar el sistema al PC, para al mismo tiempo mostrar en la pantalla del ordenador todos los valores tomados. El LED sigue parpadeando al pasar el límite de velocidad establecido.

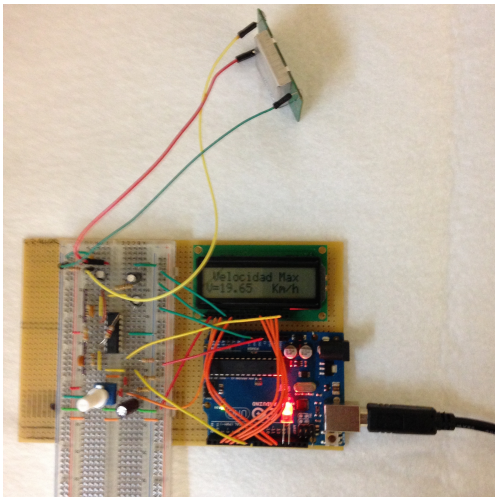


Figura 3-23 Sistema mostrando velocidad máxima superior al límite

4 VALIDACIÓN

4.1 Comprobación del funcionamiento del programa

Esta prueba tenía como objetivo comprobar el correcto funcionamiento de la librería "*FreqPeriod"* utilizada para medir la frecuencia. Para ello había que introducir en la placa de Arduino una señal conocida para poder compararla con el resultado impreso en el *display* por el Arduino.

Por lo tanto lo que se hizo fue, utilizar los equipos del laboratorio de física para sustituir la señal IF del sensor radar del sistema por una señal generada de valores conocidos. Como se puede ver en la *figura 4-1*, a la entrada del circuito acondicionador se introdujo una señal sinusoidal de amplitud 0.05 V y frecuencia 100 Hz; y la salida del circuito quedó conectada a su lugar habitual en el puerto de Arduino definido como entrada digital.

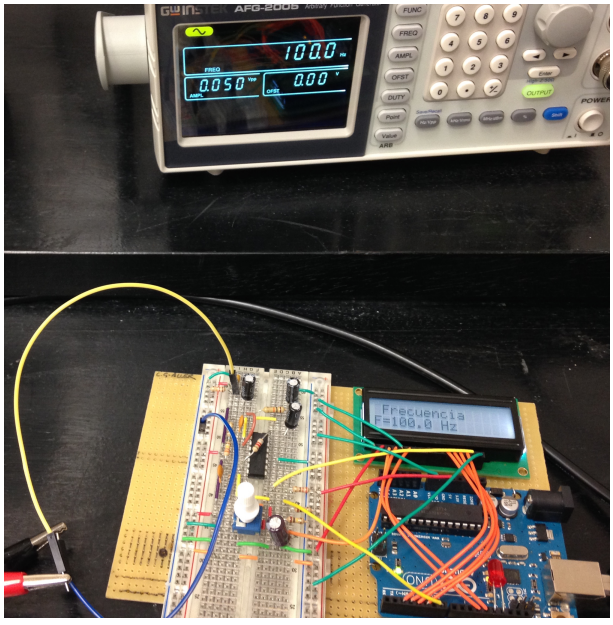


Figura 4-1 Sistema mostrando resultados de procesamiento de señal generada

El microcontrolador mandó la señal procesada al *display* mostrando el valor exacto de frecuencia (100.00 Hz).

Esta prueba se realizó con diferentes valores de frecuencia (entre 10 y 2500 Hz, que corresponde con un rango de velocidades de entre 0,5 y 130 Km/h), obteniendo siempre resultados positivos. Por lo tanto se comprobó que la librería utilizada extraía perfectamente el valor de frecuencia de la señal de salida del circuito acondicionador.

4.2 Estimación del umbral de detección

El objetivo de esta prueba era determinar el umbral de detección para el programa de "*detección/no detección"* ya que, debido a posibles interferencias, se observó que el sistema ocasionalmente daba valores de velocidad sin haber presencia de movimiento delante del mismo.

Como se ha explicado anteriormente, el programa de "*detección/no detección"* no solo enciende un LED cuando detecta movimiento, sino que cuando estaba conectado al PC, en la pantalla del ordenador se muestran los valores de velocidad que el sistema detecta cada 0.2 segundos.

Por tanto, lo que se realizó para determinar este umbral de detección fue un análisis basado en la observación de los valores de velocidad enviados al PC durante aproximadamente diez minutos sin presencia de movimiento. Durante este periodo de tiempo, el sistema tomaba un valor de velocidad cada 0,2 segundos. Esto hace un total de 3000 muestras de velocidad durante el periodo de observación. De los cuales, sólo se obtuvieron 58 valores distintos de cero. Los resultados se pueden observar en las *figuras 4-2 y 4-3*.

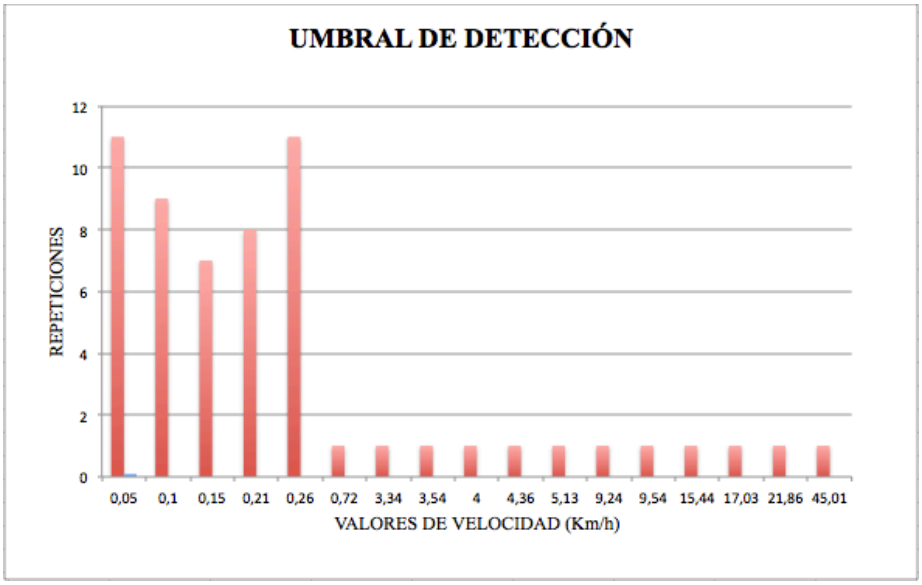


Figura 4-2 Gráfica que muestra resultados de prueba de umbral de detección

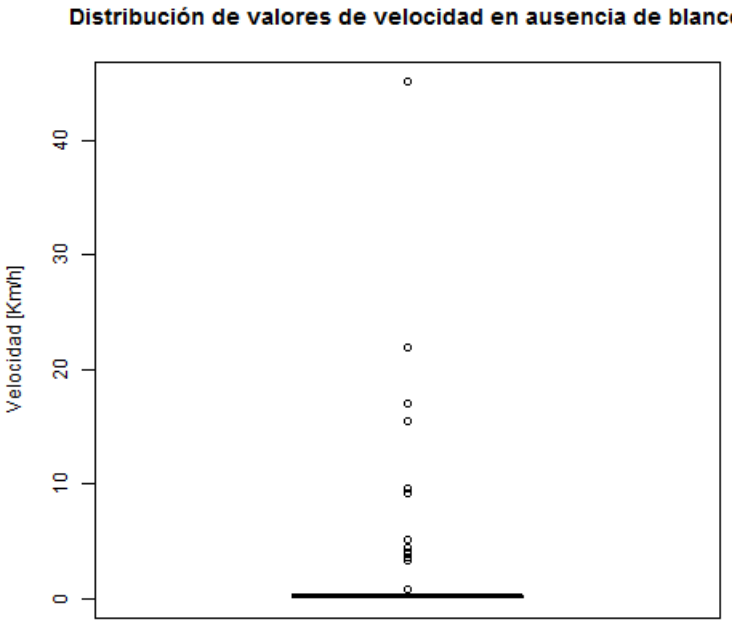


Figura 4-3 *Boxplo*t de la prueba de determinación del umbral de detección

La mayoría de los valores que salieron distintos de cero estaban entre 0,05 Km/h y 0,26 Km/h. El resto de valores únicamente aparecieron una vez durante todo el periodo de observación. Por lo tanto se decidió establecer como valor umbral 0,26 Km/h. Esto significa que si el sistema detecta un valor por encima de 0,26 Km/h, lo interpretaría como movimiento. Si el valor que detecta está por debajo, el sistema lo descarta.

4.3 Pruebas de alcance del sistema

El objetivo de esta prueba era determinar el alcance a partir del cual el sistema detecta presencia de movimiento y es capaz de determinar la velocidad de un blanco.

Para ello se utilizó el código de "*detección/no detección*". La prueba consistió en acercarse al módulo radar desde un punto que estuviera fuera del alcance del sistema. En el momento en el que el sistema encendía el LED, es decir, detectaba tres valores de velocidad mayores que el umbral, la persona se paraba y media la distancia restante hasta el módulo radar con un medidor de distancias de ultrasonidos. Cabe destacar que en todas las pruebas realizadas el sistema fue capaz de detectar movimiento (en todas ellas se encendió el LED) aunque con distancias de detección variables.

Se tomaron treinta muestras y se obtuvieron los siguientes resultados, representados en la gráfica de las *figuras 4-4 y 4-5*.

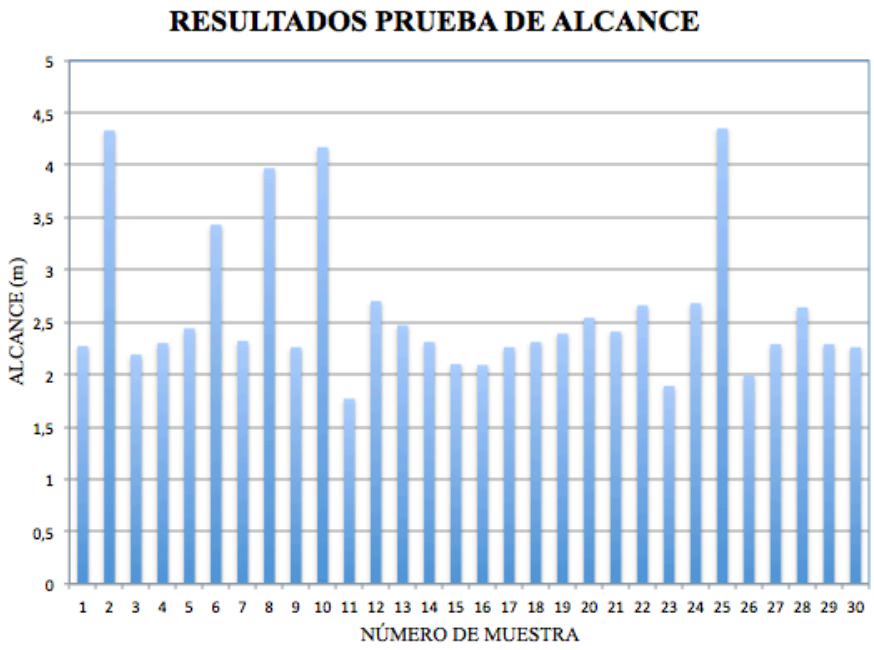


Figura 4-4 Gráfica que muestra resultados de prueba de alcance

- Distancia máxima: 4,35 m.
- Distancia mínima: 1,77 m.
- Distancia media: 2,60 m.
- Mediana: 2,30 m.

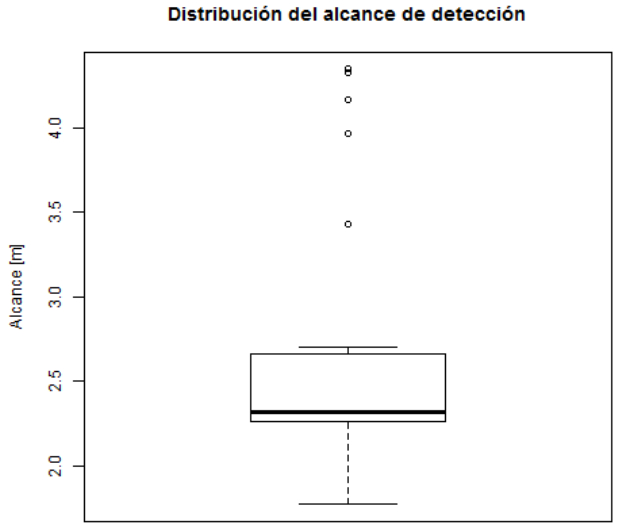


Figura 4-5 *Boxplot* de la prueba de alcance

Vistos los resultados se estima que el sistema tiene una alcance de aproximadamente 2,3 m.

4.4 Pruebas de medida de velocidad

El objetivo de esta prueba era saber si el sistema era capaz de medir la velocidad de un blanco correctamente. Para lo cual había que conocer la velocidad a la que el blanco se movía y, de esta manera, poder comparar el valor real de velocidad con el valor que mostraba el sistema en la pantalla.

Para conseguir un valor de velocidad lo que se hizo fue medir la distancia que el blanco iba a recorrer en la prueba con un medidor de distancias de ultrasonidos. Conociendo el valor de la distancia y moviéndose el blanco a una velocidad aproximadamente constante, por medio de un cronometro se obtenía el valor de velocidad del blanco gracias a la *ecuación (4)*.

$$v = \frac{d}{t} \qquad (4)$$

donde "v" es la velocidad del blanco, "d" la distancia recorrida y "t" el tiempo que tarda el blanco en recorrer la distancia.

Para que el sistema mostrase la velocidad del blanco se utilizó el programa de "*velocidad media*", en el cual, el sistema toma 10 valores cada segundo y muestra en la pantalla el valor medio.

La distancia que el blanco recorría era desde un punto a 6 metros del sensor, hasta un punto donde estaba situado el propio sensor. Como se ha visto en el apartado de *estimación de alcance*, el sistema sólo empezaría a dar valores los dos últimos metros del recorrido, teniendo en cuenta que el sistema sólo imprime un valor cada segundo, en cada muestra solo se obtenía uno o dos valores de velocidad.

Se hicieron dos pruebas: Una a velocidad "lenta" (aproximadamente 3 Km/h) en la que el blanco era una persona andando a ritmo habitual, y otra a velocidad "rápida" (aproximadamente 4 Km/h) en el cual el blanco era una persona andando un poco mas deprisa.

Se tomaron 30 muestras de cada prueba y con los resultados de cada una se calcularon dos gráficas de dispersión y dos rectas de regresión lineal para comparar los valores calculados con la *ecuación (4)* y los valores de velocidad obtenidos por el sistema de detección.

4.4.1 Prueba de velocidad "lenta"

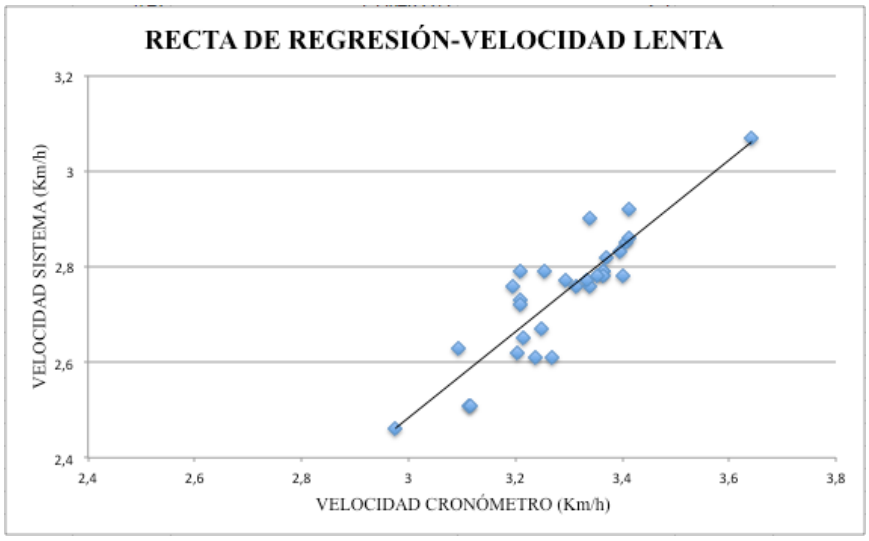


Figura 4-6 Representación de resultados obtenidos en prueba de medida de velocidad "lenta"

En esta prueba se obtuvo un coeficiente de determinación[5] $R^2 = 0{,}789$ (R^2 ideal = 1) y la siguiente relación lineal (*ecuación (5)*):

$$a = 0{,}899 * b - 0{,}213 \qquad (5)$$

Siendo:

- a: Velocidad medida con el sistema radar.
- b: Velocidad obtenida con *ecuación (4)*.

Viendo la gráfica de la *figura 4-6,* que representa la primera prueba de medida de velocidad, y analizando los resultados estadísticos obtenidos se pueden llegar a varias conclusiones:

- En primer lugar se observa cierta dispersión. Esto se puede deber a varios factores como el error humano a la hora de parar el cronómetro o a la hora de andar a una velocidad constante, o simplemente a que al sistema le falta precisión.

- En segundo lugar, se puede observar que cada valor de velocidad obtenido con la *ecuación (4)* (eje horizontal de la gráfica de la *figura 4-6*) corresponde con un valor del sistema (eje vertical de la gráfica de la *figura 4-6*) sumando una corrección de aproximadamente 0,21 Km/h como se puede observar en la *ecuación (5)*. Esto puede ser debido a que el sistema detecta sólo los 2 últimos metros y la persona acaba parándose completamente delante del radar, lo que hace que la persona comience a frenar una vez dentro del alcance del sensor empezando a reducir los valores de velocidad captados por el radar.

- Por último, se puede observar que la recta de regresión tiene una inclinación de aproximadamente 42 grados. Esto es algo positivo, ya que quiere decir que un aumento de la velocidad del blanco obtenida por medio de la *ecuación (4)* corresponde con un aumento prácticamente igual en la velocidad que muestra el sistema de detección. El ángulo de inclinación ideal es de 45 grados que quiere decir que el aumento de velocidad sería el mismo de las dos maneras, es decir, que $a = b$.

[5] En estadística, el coeficiente de determinación, denominado R^2 es un estadístico usado en el contexto de un modelo estadístico cuyo principal propósito es predecir futuros resultados o probar una hipótesis. El coeficiente determina la calidad del modelo para replicar los resultados, y la proporción de variación de los resultados que puede explicarse por el modelo [29].

4.4.2 Prueba de velocidad "rápida"

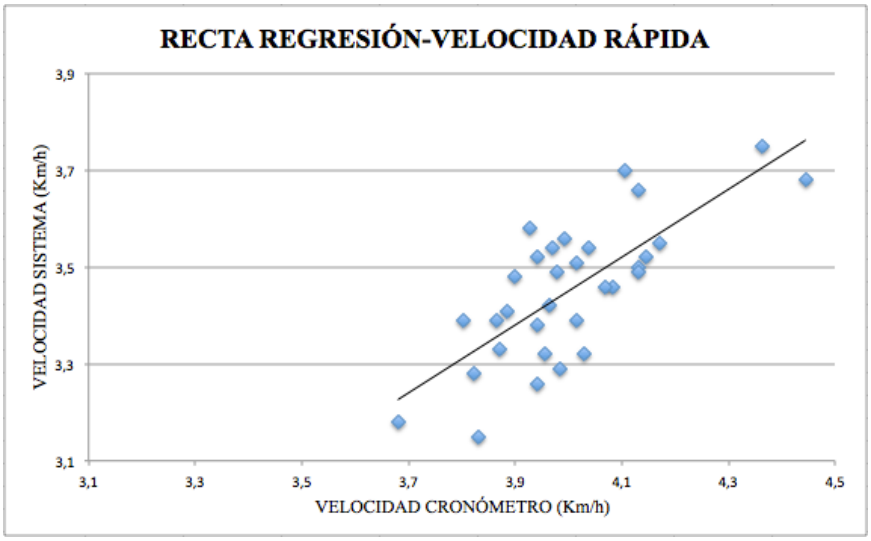


Figura 4-7 Representación de resultados obtenidos en prueba de medida de velocidad "rápida"

En esta prueba se obtuvo un coeficiente de determinación $R^2 = 0{,}511$ (R^2 ideal = 1) y la siguiente relación lineal (*ecuación (6)*):

$$a = 0{,}649 * b - 0{,}845 \qquad (6)$$

Siendo:

- a: Velocidad medida con el sistema radar.
- b: Velocidad obtenida con *ecuación (4)*.

Analizando esta segunda gráfica (*figura 4-7*), que representa los resultados de la segunda prueba de velocidad cuya única diferencia con la primera es un pequeño aumento de la velocidad del blanco, se pueden observar las siguientes diferencias:

- Hay una mayor dispersión. Esto puede ser debido al aumento de velocidad del blanco, que hace que el error humano a la hora de marcar el tiempo y llevar una velocidad constante sea algo mayor.

- Se observa además que cada valor de velocidad obtenido con la *ecuación (4)* (eje horizontal de la gráfica de la *figura 4-7*) corresponde con un valor del sistema (eje vertical de la gráfica de la *figura 4-7*) sumando una corrección de aproximadamente 0,85 Km/h como se puede observar en la *ecuación (6)* frente a los 0,21 Km/h de la prueba a velocidad "lenta". Esta diferencia se puede deber a que al andar más rápido, la deceleración en los dos últimos metros en los que detecta el sistema, es mayor.

- La recta de regresión tiene una inclinación de aproximadamente 33 grados. Este resultado se aleja mucho más del resultado ideal de 45 grados. Lo cual indica que un aumento en la velocidad obtenida con la *ecuación (4)* se corresponde con un aumento menor de la velocidad

detectada por el sistema de detección. La razón por la que ocurre esto puede estar relacionada con la deceleración del blanco dentro del alcance del radar.

4.4.3 Prueba en exteriores

Además, se realizó también una prueba en exteriores en la que se pretendía medir la velocidad de un coche circulando a una velocidad conocida.

En la mayoría de los intentos que se realizaron, el sistema no fue capaz de dar la velocidad del coche, que se acercaba hacia el sensor a una velocidad de 30 Km/h. Únicamente mostró un valor lógico (aproximadamente 27 Km/h) en dos ocasiones cuando el coche pasó muy cerca del sensor. Esto puede ser debido al corto alcance del sistema, o a posibles interferencias al estar en exteriores, ya que la antena tiene muy poca directividad.

Una posible solución sería utilizar una antena que hiciera el sistema más directivo. De esta manera se podría aumentar un poco el alcance y reducir las posibles interferencias.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Llegados a este punto, se puede afirmar que se han cumplido los objetivos planteados al principio del proyecto, al conseguir un sistema que puede usarse, por un lado, para la detección de presencia de movimiento y, por otro lado, para la medida de velocidad.

En cuanto al objetivo académico del proyecto, se puede afirmar que se trata de un proyecto que abarca varios campos estudiados durante la carrera en asignaturas como Sistemas de Control y Sensores Navales, donde se estudian los diferentes tipos de radar y su funcionamiento, Tecnología Electrónica, donde se aprende a analizar el funcionamiento de diferentes circuitos electrónicos, e Informática para la Ingeniería, donde se introduce al alumno en el ámbito de la programación.

Por lo tanto se puede sacar como conclusión que este proyecto es una buena herramienta docente planteada como resumen y aplicación práctica de estas tres materias destacadas del grado.

En cuanto al planteamiento inicial de posible uso del proyecto para la asignatura de Sistemas de Control y Sensores Navales, puede ser interesante utilizarlo como ejemplo de sistema de radar Doppler para obtención de velocidad, en alguna práctica en la que se explique junto a otros tipos de radar, utilizando diferentes proyectos relacionados.

5.2 Líneas Futuras

El presente TFG puede ser utilizado como base para posibles proyectos futuros. Ya que el sistema cuenta con muchas mejoras o modificaciones posibles. Algunas de ellas podrían ser:

- Conseguir un mayor alcance del sistema.
- Conseguir una mayor directividad.
- Mejorar el mecanismo de detección de movimiento utilizando un sensor PIR.
- Caracterización del sensor usando el instrumental necesario del CEMEDEM.
- Cambiar el modo de uso de onda continua a onda pulsada.
- Obtención de la distancia del blanco además de la velocidad.

6 BIBLIOGRAFÍA

[1] «Real Academia Española (R.A.E.),» [En línea]. Available: http://lema.rae.es/drae/?val=radar. [Último acceso: 26 Enero 2015].

[2] ««meteovallirana.com»,» [En línea]. Available: http://www.meteovallirana.es/mediapool/96/966160/data/historia_del_radar.parker.pdf. [Último acceso: 26 Enero 2015].

[3] ««campodocs.com»,» [En línea]. Available: http://campodocs.com/articulos-para-saber-mas/article_45374.html. [Último acceso: 26 Enero 2015].

[4] ««unicrom.com»,» [En línea]. Available: http://www.unicrom.com/art_historia_radar.asp. [Último acceso: 26 Enero 2015].

[5] ««historiade.jimbo.com»,» [En línea]. Available: http://historiade.jimdo.com/inventos/historia-del-radar/. [Último acceso: 26 Enero 2015].

[6] ««windows2universe.org»,» [En línea]. Available: http://www.windows2universe.org/php/tour_test_sqli.php?page=/earth/Atmosphere/tornado/radar_history.html&lang=s. [Último acceso: 26 Enero 2015].

[7] ««tecnoradar.es»,» [En línea]. Available: http://www.tecnoradar.es. [Último acceso: 26 Enero 2015].

[8] ««arantxa.ii.uam.es»,» [En línea]. Available: http://arantxa.ii.uam.es/~taao1/practica/trabajos0203/pdfs/radar0203.pdf. [Último acceso: 26 Enero 2015].

[9] «Real Academia de Ingeniería (RAING),» [En línea]. Available: http://diccionario.raing.es. [Último acceso: 26 Enero 2015].

[10] ««laserlince.com»,» [En línea]. Available: http://www.laserlince.com/datasheets/10.525GHz%20Microwave%20Motion%20Sensor.pdf. [Último acceso: 18 Febrero 2015].

[11] ««limpkin.fr»,» [En línea]. Available: http://www.limpkin.fr/public/HB100/HB100_Microwave_Sensor_Application_Note.pdf. [Último acceso: 30 Ebero 2015].

[12] ««ocw.uc3m.es»,» [En línea]. Available: http://ocw.uc3m.es/teoria-de-la-senal-y-comunicaciones/microondas-y-circuitos-de-alta-frecuencia/temas/osciladores2009.pdf. [Último acceso: 26 Enero 2015].

[13] ««ocw.uc3m.es»,» [En línea]. Available: http://ocw.uc3m.es/teoria-de-la-senal-y-comunicaciones/microondas-y-circuitos-de-alta-frecuencia/temas/mezcladores2009.pdf. [Último acceso: 26 Enero 2015].

[14] ««antenna-theory.com»,» [En línea]. Available: http://www.antenna-theory.com/spanish/antennas/patches/patch.php. [Último acceso: 26 Enero 2015].

[15] ««catarina.udlap.mx»,» [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/hernandez_a_r/capitulo3.pdf.. [Último acceso: 13 Enero 2015].

[16] ««hipertextual.com»,» [En línea]. Available: http://hipertextual.com/archivo/2014/03/hardware-novatos-arduino/. [Último acceso: 26 Enero 2015].

[17] ««Arduino Verkstad Education»,» [En línea]. Available: http://madrid.verkstad.cc/es/course-literature/que-es-arduino/. [Último acceso: 26 Enero 2015].

[18] ««arduinodhtics.weebly.com»,» [En línea]. Available: http://arduinodhtics.weebly.com/historia.html. [Último acceso: 24 Enero 2015].

[19] ««luisllamas.es»,» [En línea]. Available: http://www.luisllamas.es/2013/10/que-es-arduino-que-modelo-comprar/. [Último acceso: 28 Enero 2015].

[20] ««arduino.cc»,» [En línea]. Available: http://arduino.cc/en/Main/arduinoBoardUno. [Último acceso: 26 Enero 2015].

[21] ««didacticaselectronicas.com»,» [En línea]. Available: http://www.didacticaselectronicas.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=621&category_id=39&option=com_virtuemart&Itemid=108&vmcchk=1&Itemid=108. [Último acceso: 23 Enero 2015].

[22] ««uca.es»,» [En línea]. Available: http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANEXOS_2011_2012/22232441_310201212102.pdf. [Último acceso: 29 Enero 2015].

[23] ««cscjprofes.com»,» [En línea]. Available: http://www.cscjprofes.com/wp-content/uploads/2014/02/teoria_arduino2009.pdf. [Último acceso: 27 Enero 2015].

[24] Banzi, Massimo. Getting Started With Arduino. Massimo Banzi. Second Edition., O'Reilly 2nd Edition.

[25] ««playground.arduino.cc»,» [En línea]. Available: http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf. [Último acceso: 26 Enero 2015].

[26] ««arduino.cc»,» [En línea]. Available: http://arduino.cc/en/pmwiki.php?n=Reference/HomePage. [Último acceso: 27 Enero 2015].

[27] «interface.khm,» [En línea]. Available: http://interface.khm.de/index.php/lab/interfaces-advanced/frequency-measurement-library/. [Último acceso: 15 Febrero 2015].

[28] ««upv.es»,» [En línea]. Available: http://www.upv.es/

antenas/Tema_1/diagramas_de_radiacion.htm. [Último acceso: 26 Enero 2015].

[29] «es.wikipedia.org,» [En línea]. Available: http://es.wikipedia.org/wiki/Coeficiente_de_determinación. [Último acceso: 5 Marzo 2015].

[30] «es.wikipedia.org,» [En línea]. Available: http://es.wikipedia.org/wiki/Placa_de_pruebas. [Último acceso: 5 Marzo 2015].

ANEXO I: CÓDIGO DETECCIÓN/NO DETECCIÓN

```
#include <FreqPeriod.h> // utilizamos la libreria de Martin Nawrath para medir la frecuencia de la senal.
#include <LiquidCrystal.h> // utilizamos esta libreria para imprimir en el display LCD.
//definimos las variables necesarias de este código:
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
double lfrq;
long int pp;
const int LED = 13;
int valor_1 = 1;
int valor_2 = 1;
int valor_3 = 1;
void setup() {
  Serial.begin(115200); //iniciamos comunicacion con el PC
  FreqPeriod::begin(); //comenzamos a leer la frecuencia de la senal utilizando la libreria
  Serial.println( "FreqPeriod Library Test" ); //imprimimos FreqPeriod Library Test en el PC al arrancar una prueba.
  pinMode(LED, OUTPUT); //definimos el LED como output.
}
void loop() { //comienza el codigo ciclico.
  delay(200); // es necesario esperar un tiempo antes de la primera lectura ya que si no siempre seria cero.
  pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
      valor_1 = lfrq;
      delay(200);
pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
      valor_2 = lfrq;
      delay(200);
pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
      valor_3 = lfrq;
      delay(200);
    // tomamos tres valores de velocidad en 0.6 segundos.
Serial.print(valor_1 / 19.49);
      Serial.print( "  (1)Km/h\n");
    Serial.print(valor_2 / 19.49);
      Serial.print( "  (2)Km/h\n");
   Serial.print(valor_3 / 19.49);
      Serial.print( "  (3)Km/h\n");
  //imprimimos los tres valores en el PC
  if ((valor_1 > 0.26) && (valor_2 > 0.26) && (valor_3 > 0.26)){ //si los tres valores son mayores que cero:
   digitalWrite(LED, HIGH); //encender el LED.
   lcd.begin(16,2); //comienza presentacion en LCD.
    lcd.setCursor(15,0); //definimos donde queremos que empiece a escribir.
    lcd.print( "BIENVENIDO" );
```

```
 lcd.setCursor(15,1);
 lcd.print( "A LA E.N.M." );
 // escribimos “BIENVENIDO A LA ENM” el LCD
 for (int mover= 0; mover <12; mover++){
    lcd.scrollDisplayLeft();
    delay(50); // movemos el mensaje de bienvenida desde la derecha hasta el centro del LCD.
  }
delay(2000); //esperamos 2 segundos.
digitalWrite(LED, LOW); // apagamos el LED
lcd.clear(); //borramos el display
lcd.setCursor(3,0);
 lcd.print( "DISFRUTE SU" );
 lcd.setCursor(5,1);
 lcd.print( "VISITA" );
 delay(1000);
 lcd.clear();
 //escribimos otro mensaje que dura en la pantalla 1 segundo.
 }
}
```

ANEXO II: CÓDIGO VELOCIDAD PROMEDIO

```
#include <FreqPeriod.h> //utilizamos la libreria de Martin Nawrath para medir la frecuencia de la senal/
#include <LiquidCrystal.h> //utilizamos la libreria para imprimir en el display LCD.
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
double lfrq;
long int pp;
int entrada = 7;
int valor_1 = 1;
int valor_2 = 1;
int valor_3 = 1;
int valor_4 = 1;
int valor_5 = 1;
int valor_6 = 1;
int valor_7 = 1;
int valor_8 = 1;
int valor_9 = 1;
int valor_10 = 1;
int valor_11 = 1;
int lecturas = 0;
int sumatorio = 0;
int promedio = 0;
//definimos todas la variabnles necesarias en este codigo.
void setup() {
  Serial.begin(115200); //iniciamos comunicacion con el PC.
  FreqPeriod::begin(); //comenzamos a leer la frecuencia de la senal utilizanbdo la libreria.
  Serial.println("FreqPeriod Library Test"); // imprimimos en el PC FreqPeriod Library Test al inicio de la prueba.
lcd.begin(16,2); // Definimos la LCD con dimension 2x16 y definimos los caracteres que deben salir en las filas:
lcd.setCursor(1,0); // elegimos la posicion donde queremos que empiece a escribir la informacion en el LCD.
lcd.print(“”Velocidad Med”); //imprimimos velocidad med en la primera linea del display.
lcd.setCursor(0,1);
lcd.print(“V= “); // en la segunda linea imprimimos V= y dejamos hueco para que imprima el valor de la velocidad.
}
void loop() { //comienza el codigo que se repetira ciclicamente.
  delay (100);
 pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
    if (lecturas == 0) {
      valor_1 = lfrq;
      lecturas = 1;
   delay(100);
    }
      pp = FreqPeriod::getPeriod();
```

```
 lfrq = 16000400.0 /pp;
 if (lecturas == 1) {
  valor_2 = lfrq;
  lecturas = 2;
 delay(100);
 }
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 2) {
  valor_3 = lfrq;
  lecturas = 3;
 delay(100);
 }
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 3) {
  valor_4 = lfrq;
  lecturas = 4;
 delay(100);
}
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 4) {
  valor_5 = lfrq;
  lecturas = 5;
 delay(100);
}
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 5) {
  valor_6 = lfrq;
  lecturas = 6;
 delay(100);
 }
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 6) {
  valor_7 = lfrq;
  lecturas = 7;
 delay(100);
}
 pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 7) {
  valor_8 = lfrq;
  lecturas = 8;
```

```
    delay(100);
  }
  pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
  if (lecturas == 8) {
    valor_9 = lfrq;
    lecturas = 9;
   delay(100);
  }
  pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
  if (lecturas == 9) {
    valor_10 = lfrq;
    lecturas = 10;
   delay(100);
  }
  pp = FreqPeriod::getPeriod();
    lfrq = 16000400.0 /pp;
  if (lecturas == 10) {
    valor_11 = lfrq;
    lecturas = 11;
  delay(100);
 }
//tomamos 11 valores de velocidad durante 1.1 segundos
 if ((valor_1 > 15*valor_2) && (valor_1 > 15*valor_3) && (valor_1 > 15*valor_4) && (valor_1 > 15*valor_5) && (valor_1 > 15*valor_6) &&
(valor_1 > 15*valor_7) && (valor_1 > 15*valor_8) && (valor_1 > 15*valor_9) && (valor_1 > 15*valor_10) && (valor_1 > 15*valor_11)){
    sumatorio = valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;
    promedio = sumatorio / 10;
  }
  else {
    sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;
    promedio = sumatorio / 11;
  }
  if ((valor_2 > 15*valor_1) && (valor_2 > 15*valor_3) && (valor_2 > 15*valor_4) && (valor_2 > 15*valor_5) && (valor_2 > 15*valor_6) &&
(valor_2 > 15*valor_7) && (valor_2 > 15*valor_8) && (valor_2 > 15*valor_9) && (valor_2 > 15*valor_10) && (valor_2 > 15*valor_11)){
sumatorio = valor_1 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;
    promedio = sumatorio / 10;
  }
  if ((valor_3 > 15*valor_2) && (valor_3 > 15*valor_1) && (valor_3 > 15*valor_4) && (valor_3 > 15*valor_5) && (valor_3 > 15*valor_6) &&
(valor_3 > 15*valor_7) && (valor_3 > 15*valor_8) && (valor_3 > 15*valor_9) && (valor_3 > 15*valor_10) && (valor_3 > 15*valor_11)){
sumatorio = valor_1 + valor_2 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;
    promedio = sumatorio / 10;
  }
  if ((valor_4 > 15*valor_2) && (valor_4 > 15*valor_3) && (valor_4 > 15*valor_1) && (valor_4 > 15*valor_5) && (valor_4 > 15*valor_6) &&
(valor_4 > 15*valor_7) && (valor_4 > 15*valor_8) && (valor_4 > 15*valor_9) && (valor_4 > 15*valor_10) && (valor_4 > 15*valor_11)){
sumatorio = valor_1 + valor_2 + valor_3+ valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;
    promedio = sumatorio / 10;
  }
```

```
  if ((valor_5 > 15*valor_2) && (valor_5 > 15*valor_3) && (valor_5 > 15*valor_4) && (valor_5 > 15*valor_1) && (valor_5 > 15*valor_6) &&
(valor_5 > 15*valor_7) && (valor_5 > 15*valor_8) && (valor_5 > 15*valor_9) && (valor_5 > 15*valor_10) && (valor_5 > 15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_6 > 15*valor_2) && (valor_6 > 15*valor_3) && (valor_6 > 15*valor_4) && (valor_6 > 15*valor_5) && (valor_6 > 15*valor_1) &&
(valor_6 > 15*valor_7) && (valor_6 > 15*valor_8) && (valor_6 > 15*valor_9) && (valor_6 > 15*valor_10) && (valor_6 > 15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_7 + valor_8 + valor_9 +valor_10 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_7 > 15*valor_2) && (valor_7 > 15*valor_3) && (valor_7 > 15*valor_4) && (valor_7 > 15*valor_5) && (valor_7 > 15*valor_6) &&
(valor_7 > 15*valor_1) && (valor_7 > 15*valor_8) && (valor_7 > 15*valor_9) && (valor_7 > 15*valor_10) && (valor_7 > 15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_8 + valor_9 +valor_10 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_8 > 15*valor_2) && (valor_8 > 15*valor_3) && (valor_8 > 15*valor_4) && (valor_8 > 15*valor_5) && (valor_8 > 15*valor_6) &&
(valor_8 > 15*valor_7) && (valor_8 > 15*valor_1) && (valor_8 > 15*valor_9) && (valor_8 > 15*valor_10) && (valor_8 > 15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_9 +valor_10 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_9 > 15*valor_2) && (valor_9 > 15*valor_3) && (valor_9 > 15*valor_4) && (valor_9 > 15*valor_5) && (valor_9 > 15*valor_6) &&
(valor_9 > 15*valor_7) && (valor_9 > 15*valor_8) && (valor_9 > 15*valor_1) && (valor_9 > 15*valor_10) && (valor_9 > 15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 +valor_10 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_10 > 15*valor_2) && (valor_10 > 15*valor_3) && (valor_10 > 15*valor_4) && (valor_10 > 15*valor_5) && (valor_10 > 15*valor_6)
&& (valor_10 > 15*valor_7) && (valor_10 > 15*valor_8) && (valor_10 > 15*valor_9) && (valor_10 > 15*valor_1) && (valor_10 >
15*valor_11)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 + valor_11;

    promedio = sumatorio / 10;

  }

  if ((valor_11 > 15*valor_2) && (valor_11 > 15*valor_3) && (valor_11 > 15*valor_4) && (valor_11 > 15*valor_5) && (valor_11 > 15*valor_6)
&& (valor_11 > 15*valor_7) && (valor_11 > 15*valor_8) && (valor_11 > 15*valor_9) && (valor_11 > 15*valor_10) && (valor_11 >
15*valor_1)){

sumatorio = valor_1 + valor_2 + valor_3 + valor_4 + valor_5 + valor_6 + valor_7 + valor_8 + valor_9 +valor_10;

    promedio = sumatorio / 10;

  }

    printDouble(valor_1 / 19.49,2);

     Serial.print( "  (1)Km/h\n" );

    printDouble(valor_2 / 19.49,2);

     Serial.print( "  (2)Km/h\n" );

    printDouble (valor_3 / 19.49,2);

     Serial.print( "  (3)Km/h\n" );

    printDouble (valor_4 / 19.49,2);

     Serial.print( "  (4)Km/h\n" );

    printDouble (valor_5 / 19.49,2);

     Serial.print( "  (5)Km/h\n" );

    printDouble (valor_6 / 19.49,2);

     Serial.print( "  (6)Km/h\n" );

    printDouble (valor_7 / 19.49,2);
```

```
    Serial.print( "  (7)Km/h\n" );
   printDouble (valor_8 / 19.49,2);
   Serial.print( "  (8)Km/h\n" );
   printDouble (valor_9 / 19.49,2);
    Serial.print ( "  (9)Km/h\n" );
   printDouble (valor_10 / 19.49,2);
    Serial.print ( "  (10)Km/h\n" );
   printDouble (valor_11 / 19.49,2);
    Serial.print( "  (11)Km/h\n" );
      printDouble (promedio / 19.49,2);
   Serial.print ( "  (MEDIA)Km/h\n" );
  //imprimimos en el PC cada uno de los valores tomados y el valor medio/
lcd.setCursor(2,1);
  lcd.print("               ");
  lcd.setCursor(2,1);
  lcdprintDouble(promedio / 19.49, 2); //imprimimos en el LCD el valor promedio.
  lcd.setCursor(7,1);
  lcd.print(" Km/h"); //a continuacion del valor de la velocidad apareceran la unidades.
  sumatorio = 0;
  promedio = 0;
  lecturas = 0;
 //ponemos de nuevo las variables a cero.
 }
```

ANEXO III: CÓDIGO VELOCIDAD MÁXIMA

```
#include <FreqPeriod.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
double lfrq;
long int pp;

int entrada = 7;
const int LED = 13;
int valor_1 = 1;
int valor_2 = 1;
int valor_3 = 1;
int valor_4 = 1;
int valor_5 = 1;
int valor_6 = 1;
int valor_7 = 1;
int valor_8 = 1;
int valor_9 = 1;
int valor_10 = 1;
int valor_11 = 1;
int lecturas = 0;
void setup() {
  Serial.begin(115200);
  FreqPeriod::begin();
  Serial.println("FreqPeriod Library Test");
  pinMode(LED, OUTPUT);
    // Definimos la LCD con dimension 2x16 y definimos los caracteres que deben salir en las filas:
lcd.begin(16,2);
lcd.setCursor(1,0);
lcd.print("Velocidad Max");
lcd.setCursor(0,1);
lcd.print("V=")
}
void loop() {
  pp=FreqPeriod::getPeriod();
  if (pp ){
    Serial.print("period: ");
    Serial.print(pp);
    Serial.print(" 1/16us  /  frequency: ");
  }
  delay (100);
 pp = FreqPeriod::getPeriod();
```

```
 lfrq = 16000400.0 /pp;
 if (lecturas == 0) {
   valor_1 = lfrq;
   lecturas = 1;
delay(100);
 }
   pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 1) {
   valor_2 = lfrq;
   lecturas = 2;
  delay(100);
 }
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 2) {
   valor_3 = lfrq;
   lecturas = 3;
  delay(100);
 }
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 3) {
   valor_4 = lfrq;
   lecturas = 4;
  delay(100);
}
 pp = FreqPeriod::getPeriod();
 lfrq = 16000400.0 /pp;
 if (lecturas == 4) {
   valor_5 = lfrq;
   lecturas = 5;
  delay(100);
}
pp = FreqPeriod::getPeriod();
   lfrq = 16000400.0 /pp;
 if (lecturas == 5) {
   valor_6 = lfrq;
   lecturas = 6;
 delay(100);
 }
pp = FreqPeriod::getPeriod();
   lfrq = 16000400.0 /pp;
 if (lecturas == 6) {
   valor_7 = lfrq;
   lecturas = 7;
  delay(100);
```

```
}
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 7) {
  valor_8 = lfrq;
  lecturas = 8;
  delay(100);
}
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 8) {
  valor_9 = lfrq;
  lecturas = 9;
  delay(100);
}
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 9) {
  valor_10 = lfrq;
  lecturas = 10;
  delay(100);
}
pp = FreqPeriod::getPeriod();
  lfrq = 16000400.0 /pp;
 if (lecturas == 10) {
  valor_11 = lfrq;
  lecturas = 11;
 delay(100);
}
 Serial.print(valor_1 / 19.49);
  Serial.print( "  (1)Km/h\n");
 Serial.print(valor_2 / 19.49);
  Serial.print( "  (2)Km/h\n");
 Serial.print(valor_3 / 19.49);
  Serial.print( "  (3)Km/h\n");
 Serial.print(valor_4 / 19.49);
  Serial.print( "  (4)Km/h\n");
 Serial.print(valor_5 / 19.49);
  Serial.print( "  (5)Km/h\n");
 Serial.print(valor_6 / 19.49);
  Serial.print ( "  (6)Km/h\n");
 Serial.print (valor_7 / 19.49);
  Serial.print ( "  (7)Km/h\n");
 Serial.print (valor_8 / 19.49);
 Serial.print ( "  (8)Km/h\n");
 Serial.print (valor_9 / 19.49);
```

```
    Serial.print( "  (9)Km/h\n");
    Serial.print(valor_10 / 19.49);
    Serial.print( "  (10)Km/h\n");
    Serial.print(valor_11 / 19.49);
    Serial.print( "  (11)Km/h\n");
  if ((valor_1 > valor_2) && (valor_1 > valor_3) && (valor_1 > valor_4) && (valor_1 > valor_5) && (valor_1 > valor_6) && (valor_1 > valor_7)
&& (valor_1 > valor_8) && (valor_1 > valor_9) && (valor_1 > valor_10) && (valor_1 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("               ");
  lcd.setCursor(2,1);
  lcd.print(valor_1 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_1 / 19.49);
  }
   if ((valor_2 > valor_1) && (valor_2 > valor_3) && (valor_2 > valor_4) && (valor_2 > valor_5) && (valor_2 > valor_6) && (valor_2 > valor_7)
&& (valor_2 > valor_8) && (valor_2 > valor_9) && (valor_2 > valor_10) && (valor_2 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("               ");
  lcd.setCursor(2,1);
  lcd.print(valor_2 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_2 / 19.49);
  }
   if ((valor_3 > valor_2) && (valor_3 > valor_1) && (valor_3 > valor_4) && (valor_3 > valor_5) && (valor_3 > valor_6) && (valor_3 > valor_7)
&& (valor_3 > valor_8) && (valor_3 > valor_9) && (valor_3 > valor_10) && (valor_3 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("               ");
  lcd.setCursor(2,1);
  lcd.print(valor_3 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_3 / 19.49);
}
   if ((valor_4 > valor_2) && (valor_4 > valor_3) && (valor_4 > valor_1) && (valor_4 > valor_5) && (valor_4 > valor_6) && (valor_4 > valor_7)
&& (valor_4 > valor_8) && (valor_4 > valor_9) && (valor_4 > valor_10) && (valor_4 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("               ");
  lcd.setCursor(2,1);
  lcd.print(valor_4 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_4 / 19.49);
  }
```

```
  if ((valor_5 > valor_2) && (valor_5 > valor_3) && (valor_5 > valor_4) && (valor_5 > valor_1) && (valor_5 > valor_6) && (valor_5 > valor_7) && (valor_5 > valor_8) && (valor_5 > valor_9) && (valor_5 > valor_10) && (valor_5 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_5 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print(  "MAX: " );
  Serial.print(valor_5 / 19.49);
  }
  if ((valor_6 > valor_2) && (valor_6 > valor_3) && (valor_6 > valor_4) && (valor_6 > valor_5) && (valor_6 > valor_1) && (valor_6 > valor_7) && (valor_6 > valor_8) && (valor_6 > valor_9) && (valor_6 > valor_10) && (valor_6 > valor_11)){
lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_6 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print(  "MAX: " );
  Serial.print(valor_6 / 19.49);
  }
  if ((valor_7 > valor_2) && (valor_7 > valor_3) && (valor_7 > valor_4) && (valor_7 > valor_5) && (valor_7 > valor_6) && (valor_7 > valor_1) && (valor_7 > valor_8) && (valor_7 > valor_9) && (valor_7 > valor_10) && (valor_7 > valor_11)){
 lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_7 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print(  "MAX: " );
  Serial.print(valor_7 / 19.49);
  }
  if ((valor_8 > valor_2) && (valor_8 > valor_3) && (valor_8 > valor_4) && (valor_8 > valor_5) && (valor_8 > valor_6) && (valor_8 > valor_7) && (valor_8 > valor_1) && (valor_8 > valor_9) && (valor_8 > valor_10) && (valor_8 > valor_11)){
lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_8 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print(  "MAX: " );
  Serial.print(valor_8 / 19.49);
  }
  if ((valor_9 > valor_2) && (valor_9 > valor_3) && (valor_9 > valor_4) && (valor_9 > valor_5) && (valor_9 > valor_6) && (valor_9 > valor_7) && (valor_9 > valor_8) && (valor_9 > valor_10) && (valor_9 > valor_10) && (valor_9 > valor_11)){
  lcd.setCursor(2,1);
  lcd.print("                ");
```

```
  lcd.setCursor(2,1);
  lcd.print(valor_9 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_9 / 19.49);
  }
   if ((valor_10 > valor_2) && (valor_10 > valor_3) && (valor_10 > valor_4) && (valor_10 > valor_5) && (valor_10 > valor_6) && (valor_10 > valor_7) && (valor_10> valor_8) && (valor_10 > valor_9) && (valor_10 > valor_1) && (valor_10 > valor_11)){
lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_10 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_10 / 19.49);

  }
   if ((valor_11 > valor_2) && (valor_11> valor_3) && (valor_11 > valor_4) && (valor_11 > valor_5) && (valor_11 > valor_6) && (valor_11 > valor_7) && (valor_11 > valor_8) && (valor_11 > valor_9) && (valor_11 > valor_10) && (valor_11 > valor_1)){
  lcd.setCursor(2,1);
  lcd.print("                ");
  lcd.setCursor(2,1);
  lcd.print(valor_11 / 19.49);
  lcd.setCursor(10,1);
  lcd.print("Km/h");
  Serial.print( "MAX: " );
  Serial.print(valor_11 / 19.49);
}
  delay(400);

  if ((valor_1>200)||(valor_2>200)||(valor_3>200)||(valor_4>200)||(valor_5>200)||(valor_6>200)||(valor_7>200)||(valor_8>200)||(valor_9>200)||(valor_10>200)||(valor_11>200)){
    digitalWrite (LED,HIGH);
    delay (100);
    digitalWrite (LED, LOW);
    delay (70);
    digitalWrite (LED,HIGH);
    delay (100);
    digitalWrite (LED, LOW);
    delay (70);
    digitalWrite (LED,HIGH);
    delay (100);
    digitalWrite (LED, LOW);
    delay (70);
    digitalWrite (LED,HIGH);
    delay (100);
```

```
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);
  digitalWrite (LED,HIGH);
  delay (100);
  digitalWrite (LED, LOW);
  delay (70);

 }
valor_1 = 1;
valor_2 = 1;
valor_3 = 1;
valor_4 = 1;
valor_5 = 1;
valor_6 = 1;
valor_7 = 1;
valor_8 = 1;
valor_9 = 1;
valor_10 = 1;
valor_11 = 1;
lecturas = 0;
}
```