



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECHANICO

# **DISEÑO DE UN SENSOR DE DISTANCIA Y VELOCIDAD BASADO EN UN SENSOR LASER LIDAR-LITE**

Autor: Jaime Manuel Cantos Sánchez  
Director: Nagarajan Kandasamy

Madrid  
Julio 2016

## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO**

### ***1º. Declaración de la autoría y acreditación de la misma.***

El autor D. JAIME MANUEL CANTOS SÁNCHEZ DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Diseño de un sensor de distancia y velocidad basado en un sensor laser Lidar-Lite, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### ***2º. Objeto y fines de la cesión.***

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### ***3º. Condiciones de la cesión y acceso***

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### ***4º. Derechos del autor.***

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

### ***5º. Deberes del autor.***

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

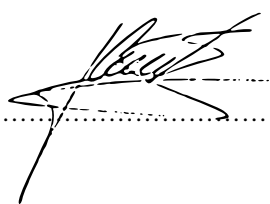
**6º. Fines y funcionamiento del Repositorio Institucional.**

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a ...17... de .....julio..... de ...2016....

**ACEPTA**

Fdo.....  


Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
DISEÑO DE UN SENSOR DE DISTANCIA Y VELOCIDAD BASADO EN  
UN SENSOR LASER LIDAR-LITE

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2016/2017 es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es  
plagio de otro, ni total ni parcialmente y la información que ha sido tomada  
de otros documentos está debidamente referenciada.

Fdo.:



Fecha: 17 / 07 / 2016

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.:

Fecha: ..... / ..... / .....

Vº Bº del Coordinador de Proyectos

Fdo.:

Fecha: ..... / ..... / .....

# DISEÑO DE UN SENSOR DE DISTANCIA Y VELOCIDAD BASADO EN UN SENSOR LASER LIDAR-LITE.

**Autor: Cantos Sánchez, Jaime Manuel**

Director: Kandasamy, Nagarajan.

Entidad Colaboradora: Drexel University.

## RESUMEN DEL PROYECTO

### *1.- INTRODUCCIÓN*

---

El presente trabajo tiene como objetivo desarrollar un dispositivo capaz asistir a conductores de todo tipo de vehículos. El dispositivo es capaz de medir distancias mediante un sensor laser “LIDAR-LITE”, con un rango de hasta 40 metros. A partir de estos valores se calcula la velocidad a la que el obstáculo o vehículo se aproxima y toda esta información es representada en una pantalla LCD, incorporada en el salpicadero, para lectura de los usuarios.

La intención es que, con una mínima interacción del usuario con el sistema, este consiga gran cantidad de información de los alrededores del vehículo sin apenas tener que desviar la atención de la carretera, evitando distracciones y haciendo más seguras las carreteras.

### *2.- ESTADO DEL ARTE*

---

El uso de tecnología capaz de detectar objetos ha ido aumentando considerablemente a lo largo del tiempo, esto se debe, en gran medida, a la multitud de aplicaciones que pueden ofrecer. En este proyecto, la tecnología específica se denomina LIDAR (un acrónimo del inglés *Light Detection and Ranging* o *Laser Imaging Detection and Ranging*), que haciendo uso de un haz laser pulsado permite determinar la distancia desde un emisor a un objeto.

El campo automovilístico ya usa multitud de sensores. Quizás el uso más conocido es el de los sistemas de aparcamiento asistido, en el que sensores colocados alrededor del vehículo ayudan a los conductores en la maniobra de aparcamiento, llegando los más modernos a realizar la maniobra de manera totalmente automatizada. En esta aplicación la mayor parte de los sensores son sistemas ultrasónicos, de rango mucho menor a los sistemas láser.

La tecnología lidar ya se está usando como sistemas de calculo de velocidades en las conocidas como “pistolas laser” de las fuerzas del estado, que las usan como alternativa a los radares dispuestos a lo largo de las calles y carreteras. La diferencia del sistema estudiado en este proyecto frente a este uso, es que en el caso de las “pistolas laser” el sensor debe de estar inmóvil, o la lectura que hará será errónea, hasta ahora no se había implementado desde el punto de vista de un emisor móvil, calculando la velocidad absoluta a partir de las velocidades relativas calculadas por el sensor.

### **3.- MOTIVACIÓN Y OBJETIVOS**

---

Una parte fundamental de la investigación en el campo automovilístico consiste en encontrar diferentes métodos para reducir accidentes y aumentar la seguridad, tanto de los conductores como de los peatones. Este campo está en continua evolución, consiguiendo que en los últimos años el número de accidentes se haya reducido de manera considerable. No obstante, el número de siniestros sigue siendo elevado, es por ello que todo avance, por pequeño que sea, puede suponer una gran diferencia.

El objetivo de este proyecto es doble: por una parte programar el sensor para que mida las distancias de manera continua, cada medio segundo realiza una nueva medición, calculando también la velocidad y, en segundo lugar, el diseño de un sistema de interacción con el usuario, constando de la pantalla donde se lee la información y de un sistema de alerta que avise al conductor de la proximidad de objetos una vez estos estén dentro de unos rangos de proximidad peligrosos.

### **4.- METODOLOGÍA**

---

El sistema básico se compone por dos elementos fundamentales: por un lado se tiene el sensor lidar y por otro la pantalla LCD, teniendo entre ambos un microprocesador Arduino, que se encargará de regir la comunicación entre los elementos del sistema, así como de realizar los cálculos necesarios para el correcto funcionamiento del mismo.

Una vez finalizada la programación y configuración de los elementos básicos se le incluirán los elementos secundarios. Estos son: un pulsador para cambiar de función y un altavoz piezoeléctrico para alarma en caso de peligro.

#### **4.1 Sensor Lidar-Lite**

El sensor utilizado, como mencionado anteriormente, hace uso de un haz laser pulsado para medir distancias. Para ello, el sensor mide el tiempo que pasa desde que emite el láser y detecta el haz reflejado. Una vez medido el tiempo, conociendo la velocidad a la que se propaga el haz, la velocidad de la luz, es capaz de determinar la distancia al objeto de manera muy precisa, el margen de error es de 1 cm. Tal y como se ha programado en este proyecto, el sensor mide la distancia cada medio segundo. La conexión entre el sensor y el microprocesador se realiza mediante el uso del bus I<sup>2</sup>C. De esta manera, tanto la conexión como la programación son mucho más sencillas a las otras alternativas.

#### **4.2 Microprocesador Arduino UNO**

El microprocesador utilizado es el Arduino UNO, debido a la facilidad de acceso a él y a su simplicidad para programar. Las funciones principales del microprocesador en este proyecto son tres: recepción de las lecturas del sensor lidar, realizar los cálculos necesarios para obtener la velocidad y la configuración de la pantalla LCD.

Debido a lo completo del sensor lidar utilizado, lo único necesario para su programación es la inclusión de la librería I2C, disponible online, al directorio del proyecto, indicar la dirección del sensor para empezar a recibir la información y la configuración de las variables para recoger las mediciones.

Los cálculos necesarios para el proyecto son muy pocos y sencillos, se basan en cambios de unidades, para hacer más intuitiva la lectura para los usuarios, pues el sensor mide en cm. Otro cálculo que debe realizar es basándose en mediciones de distancias consecutivas y el tiempo entre mediciones conocer la velocidad de aproximación, considerando tanto el sensor en movimiento como estático.

Una vez recogida la información del sensor y realizados todos los cálculos previos, se configura la pantalla LCD para que muestre la información deseada por el usuario. Para ello se incluye un pulsador que indica al microprocesador qué información mostrar, si distancia o velocidad. Además se incluye un altavoz que se configura para empezar a emitir un tono cuando la distancia medida por el sensor es menor a una distancia determinada previamente como distancia peligrosa.

## ***5.- RESULTADOS***

---

Para comprobar el correcto funcionamiento de todos los elementos se fueron realizando varias pruebas a lo largo del desarrollo del proyecto. Se configuró elemento a elemento y se comprobó que, independientemente, todos cumplían su función.

Para las mediciones de distancia se hizo uso del monitor serie, una cinta métrica y varios objetos, que se iban moviendo a lo largo del eje del sensor, comprobando que la distancia mostrada por el monitor y la cinta métrica coincidían. De esta manera se determinó que el sensor funcionaba correctamente y que el margen de error (1 cm) era el esperado y aceptable para la aplicación a estudiar.

La segunda prueba fue comprobar que la pantalla respondiese correctamente a la programación del microprocesador. Para ello se empezó escribiendo la frase “Hola mundo”, para más adelante configurarla para que mostrase la distancia medida por el sensor, comprobando que los datos del monitor serie y la pantalla coincidían.

Una vez configurada la pantalla se procedió a configurar el cálculo de las velocidades, para ello se instaló el pulsador que regulaba el cambio de la función distancia a la función velocidad.

El último elemento a instalar fue la alarma sonora. Para ello se configuró una distancia de seguridad de 1 metro. Se comprobó que siempre que el sensor detectaba una distancia inferior a esta la alarma empezaba a sonar.

Una vez finalizados los experimentos, se montó un prototipo con todos los elementos interconectados y se realizó la prueba final, en la que se comprobó que se cumplían los objetivos y que el prototipo funcionaba perfectamente.

## ***6.- CONCLUSIONES***

---

Tras todos los experimentos y pruebas, se determinó que los requisitos propuestos para este proyecto se han completado y el prototipo funcionaba perfectamente, no obstante, a pesar de haberse cubierto todos los objetivos propuestos para el mismo, este prototipo no está todavía preparado para la función para la cual fue ideado.

Para ello haría falta el uso de más de un sensor, configurando el microprocesador para regular la información entre todos y el uso de una pantalla LCD más compleja, que permitiera realizar el diseño de los alrededores del vehículo con total precisión.

Con todo se puede considerar que el resultado es un éxito, pues con un poco más de trabajo y de pruebas en entornos reales, este dispositivo podría ser de gran utilidad aumentando la seguridad vial.

# **DESIGN OF A SENSOR CAPABLE OF MEASURING DISTANCE AND SPEED BASED ON A LIDAR-LITE LASER SENSOR.**

**Author: Cantos Sánchez, Jaime Manuel.**

Director: Kandasamy, Nagarajan.

Collaborating entity: Drexel University.

## **ABSTRACT**

### ***1.- INTRODUCTION***

---

The aim of this project is to develop a device capable of assisting drivers. The device is able to measure distances with a range of up to 40 meters, by the use of the laser sensor "LIDAR-LITE". It will also be able to calculate the speed at which an object or another vehicle is approaching based on those measurements. All this information will be displayed on a LCD display, incorporated in the vehicle.

The reason for this project is so that, with a minimum action from the user, he will obtain a great deal of information about the surroundings without the need to deviate his attention from the road, avoiding distractions and making roads safer.

### ***2.- STATE OF THE ART***

---

The use of technology capable of detecting objects has experimented a considerable raise over the last years. One of the main reasons for this would be the great amount of applications such technology posses. In this project we will focus on a specific one called LIDAR (Laser Imaging Detection and Ranging), which, by the use of a laser pulse is capable of measuring distances to an object.

Technology similar to this one is already applied for motoring purposes. Perhaps the most common one of them will be the assisted parking system, which, by means of sensors located all around the vehicle, aid the driver in the parking maneuver, with the most advanced ones allowing the vehicle to park in a completely automated way. The most common of the sensors used today for this purpose are ultrasonic based, the downside of these is the very low range they posses, compared to laser ones.

In the same field, there are already systems using lidar technology. This can be found in the portable radars policemen use to determine the speed of the vehicles while driving. The main difference with the system studied and developed in this project is, in order for those systems to work, the policemen need to be still, for they are not programmed to calculate absolute speed based on relative speed.

### ***3.- MOTIVATION Y OBJECTIVES***

---

A big part of the research done nowadays by the car companies is to find ways to make motorized vehicles safer thus reducing the number of victims and making them more secure both for the drivers and the pedestrians. This is a field in constant change and evolution, and thanks to that research, the number of accidents is decreasing every year. The number of accidents is still very high, though. Therefore, any advance to improve safety, no matter how small might it be, can make a huge difference.



This project has a double intention: first, programming the sensor so that it can be continually measuring, every half a second will make a new measure, calculating also the speed. And secondly the user interface, composed by the LCD display where to read the information and a warning system, so that the driver is alerted when an obstacle or vehicle is too close.

#### ***4.- METHODOLOGY***

---

The basic system is composed by two fundamental elements: on one hand the lidar sensor and, on the other hand, the LCD display. Connecting these will be a microprocessor Arduino UNO which will be the core, controlling the information and making all the necessary calculations for the system to work correctly.

Once the configuration of these elements is done some secondary elements will be included. Such as: a button so that the user can decide what information to display and a piezo, which will work as a speaker, so that the alarm can be heard by the driver.

##### **4.1 Lidar-Lite sensor**

As previously stated, the sensor used in this project will measure distance by the use of a laser pulse. In order to do so, it will emit the pulse and detects the reflection of it, measuring the time between these two actions. Once the time has been measured, and knowing that a laser moves at the speed of light, it is capable of calculating the distances between the sensor and the object that made the laser bounce, with a very low error (around 1 cm). In this project the sensor has been programmed to measure the distance every half a second. For communication between the lidar and the microprocessor the use of the I<sup>2</sup>C bus is used, making the wiring and programming a lot simpler than any other possibility.

##### **4.2 Arduino UNO microprocessor**

The microprocessor chosen for this project was Arduino UNO, given the easy access to it, and the simplicity of its programming. The microprocessor has three main functions in this project: receiving the measurements from the lidar, making all the needed calculations for determining the speed and the programming of the LCD display.

Given how complete the lidar sensor is, the only necessity for programming it is to include the I2C library to the Arduino project directory, this can be found online, determine the address of the sensor so that the communication starts and determining the variables to collect and save the information given by the lidar.

This project has only a few and very simple calculations, most of them are basically making the readings user-friendlier. This is due to the fact that the sensor gives the data in centimeters, which is not useful for the application; so we have to change it to meters and the speed give it in kilometers per hour. Another calculation to be done is, based on two consecutive distance measures and the delay between them, determine the speed, considering both the sensor to be static or moving.

Once the information from the sensor is collected and all the math done, the LCD is programmed to show the information desired by the user. For this, the system includes a button, which will indicate the microprocessor what to show in the display. Also a speaker is programmed to start an alarm when the distance between the sensor and the object is considered as dangerous.

## **5.- RESULTS**

---

Throughout the development of the project experiments were realized to make sure every element in it worked correctly. In order to do so, every element was first proved, wired and programmed independently.

The first experiment was to check the lidar sensor. For which, the use of a ruler, various objects and the series monitor in the Arduino software were used. Objects were put on various distances from the sensor and then the information displayed in the monitor and the measurements with the ruler were compared. After this experiment the sensor was proved to work perfectly with an error margin of 1 centimeter, which was both expected and acceptable for the use of the sensor.

The second experiment was to make sure the screen was correctly wired and it was programmed correctly. To do so, the first thing to be done was wiring the screen and printing "Hola mundo". Once this was done, it was programmed to print the distance calculated by the sensor, and then checked that the data in the monitor and screen were the same.

The third experiment was the math necessary for the speed calculation. First the button was installed so that we could change between displaying the speed and distance and then checked math was correct.

The last element to configure was the alarm, to do so a dangerous distance of 1 meter was used, so that when the sensor detected anything less than 1 meter the alarm would go off.

Once all the experiments were done, the whole system was wired, so that the last proof could be done, checking that every element was correctly intercommunicated and the prototype was correctly working.

## **6.- CONCLUSIONS**

---

After all the experiments were conducted, the conclusion was that all the requisites imposed for the projects were completed and the prototype worked perfectly. Nevertheless, even after completing all the objectives, the prototype is not yet ready to be implemented on a real system.

In order to do so, more than one sensor must be added, reconfiguration of the microprocessor must be done to accept information of as many as needed to be able to, in a more complex LCD screen, draw a virtual map of the surroundings of the vehicle with minimum error.

All in all the result of the project can be considered successful, for with only a little more work and the ability to test it in real systems, this device could be of great help in reducing the accidents and making roads safer.



*Índice de la memoria*

<b>Parte I Memoria</b> .....	<b>1</b>
<b>Capítulo 1 Introducción</b> .....	<b>3</b>
<b>1.1 Estudio de los trabajos existentes / tecnologías existentes</b> .....	<b>3</b>
<b>1.2 Motivación del proyecto</b> .....	<b>5</b>
<b>1.3 Objetivos</b> .....	<b>7</b>
<b>1.4 Metodología / Solución desarrollada</b> .....	<b>8</b>
<b>1.5 Recursos / herramientas empleadas</b> .....	<b>8</b>
<b>Capítulo 2 Solución Desarrollada</b> .....	<b>11</b>
<b>2.1 Hardware</b> .....	<b>11</b>
2.1.1 Sensor LIDAR-LITE .....	11
2.1.1.1 Funcionamiento .....	12
2.1.1.2 Esquema de conexión: .....	13
2.1.1.3 Diagrama de bloques .....	13
2.1.1.4 Datos técnicos e información detallada .....	14
2.1.2 Pantalla LCD LCM1602C .....	14
2.1.2.1 Funcionamiento .....	14
2.1.2.2 Esquema de conexión: .....	15
2.1.2.3 Datos técnicos e información detallada .....	16
2.1.3 Microcontrolador Arduino UNO .....	16
2.1.3.1 Funcionamiento .....	17
2.1.3.2 Datos técnicos e información detallada .....	17
2.1.4 Esquema de conexión .....	17
2.1.5 Resultado final.....	18
<b>2.2 Software</b> .....	<b>19</b>
2.2.1 Control del sensor LIDAR-Lite.....	19
2.2.2 Control de la pantalla LCD .....	21



---

2.2.3 Programación de la alarma por proximidad .....	21
2.2.4 Operaciones realizadas .....	22
<b>Capítulo 3 Resultados/Experimentos.....</b>	<b>25</b>
3.1 Mediciones de distancia.....	25
3.2 Pruebas de pantalla .....	26
3.3 Cálculo de velocidad .....	27
3.4 Sistema de alerta.....	28
<b>Capítulo 4 Conclusiones .....</b>	<b>29</b>
<b>Capítulo 5 Futuros desarrollos .....</b>	<b>31</b>
<b>Bibliografía</b>	<b>33</b>
<b>Parte II Estudio económico.....</b>	<b>35</b>
<b>Capítulo 1 Mediciones.....</b>	<b>37</b>
1.1 Componentes.....	37
1.2 Software.....	37
1.3 Mano de obra directa .....	38
<b>Capítulo 2 Precios unitarios .....</b>	<b>39</b>
2.1 Componentes.....	39
2.2 Software.....	39
2.3 Mano de obra directa .....	40
<b>Capítulo 3 Sumas parciales.....</b>	<b>41</b>
3.1 Componentes.....	41
3.2 Software.....	42
3.3 Mano de obra directa .....	42
<b>Capítulo 4 Presupuesto general.....</b>	<b>43</b>
<b>Parte III Manual de usuario.....</b>	<b>45</b>

---



<i>Capítulo 1</i>	<i>Manual de usuario 1</i> .....	<i>47</i>
<i>Parte IV</i>	<i>Código fuente</i> .....	<i>49</i>
<i>Capítulo 1</i>	<i>Código fuente</i> .....	<i>51</i>
<i>Parte V</i>	<i>Datasheets</i> .....	<i>55</i>
<i>Capítulo 1</i>	<i>Datasheet LIDAR-LITE</i> .....	<i>57</i>
<i>Capítulo 2</i>	<i>Datasheet pantalla LCD</i> .....	<i>59</i>
<i>Capítulo 3</i>	<i>Datasheet Arduino</i> .....	<i>61</i>



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

*ÍNDICE DE FIGURAS*

---



---

## Índice de figuras

Figura 1. Fuerzas del estado usando "pistola laser" .....	4
Figura 2. Ejemplo de vehículo con sensores ultrasónicos .....	5
Figura 3. Gráfico evolución numero de accidentes por año (INE).....	6
Figura 4. Sensor LIDAR-LITE .....	11
Figura 5. Esquema de conexión LIDAR-Lite y Arduino UNO (Manual de usuario del lidar).....	13
Figura 6. Diagrama de bloques del sensor (Manual de Usuario del lidar).....	13
Figura 7. Pantalla LCD LCM1602C [Parte Frontal] .....	14
Figura 8. Esquema de conexión LCD (Pagina web oficial Arduino) .....	15
Figura 9. Microcontrolador Arduino.....	16
Figura 10. Esquema montaje completo.....	18
Figura 11. Prototipo finalizado .....	19
Figura 12. Ejemplo esquemático de un bus I <sup>2</sup> C con un maestro y varios esclavos .....	20
Figura 13. Pantalla del monitor serie durante experimento .....	26
Figura 14. Pantalla LCD mostrando la distancia .....	27
Figura 15. Pantalla LCD mostrando la velocidad como elemento fijo.....	28



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

*ÍNDICE DE TABLAS*

---





## *Índice de tablas*

Tabla 1. Numero de accidentes en 2013 (DGT) .....	6
Tabla 2. Medición de componentes .....	37
Tabla 3. Medición de software.....	37
Tabla 4. Medición mano de obra directa .....	38
Tabla 5. Precio unitario de los componentes .....	39
Tabla 6. Precio unitario de software .....	39
Tabla 7. Precio unitario de mano de obra .....	40
Tabla 8. Sumas parciales de componentes.....	41
Tabla 9. Sumas parciales de software .....	42
Tabla 10. Sumas parciales mano de obra directa.....	42
Tabla 11. Presupuesto general en dólares .....	43
Tabla 12. Presupuesto general en euros .....	43



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

*ÍNDICE DE TABLAS*

---



# *Parte I MEMORIA*



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Introducción

---



## Capítulo 1 INTRODUCCIÓN

El presente trabajo tiene como objetivo desarrollar un dispositivo capaz de asistir a conductores de todo tipo de vehículos, con la intención de añadir medidas de seguridad para los mismos y así intentar prevenir accidentes, reduciendo la mortalidad de dichos vehículos.

El dispositivo es capaz de medir distancias mediante un sensor láser “LIDAR-LITE”. A partir de los valores obtenidos se obtiene la velocidad de un obstáculo o vehículo en un rango de 40 metros. Posteriormente dichos valores son representados en una pantalla LCD para lectura de los conductores.

### ***1.1 ESTUDIO DE LOS TRABAJOS EXISTENTES / TECNOLOGÍAS EXISTENTES***

---

El uso de componentes electrónicos capaces de detectar objetos ha aumentado considerablemente desde que se empezó a hacer uso de este tipo de tecnologías. La tecnología específica usada en este proyecto se denomina LIDAR (un acrónimo del inglés *Light Detection and Ranging* o *Laser Imaging Detection and Ranging*) que permite determinar la distancia desde un emisor láser a un objeto utilizando un haz láser pulsado.

En general, la tecnología lidar tiene aplicaciones en geología, sismología y física de la atmósfera. Aunque en este trabajo nos centraremos en una posible aplicación como ayuda a conductores, pues hasta ahora el uso que se le ha dado en este campo consiste en las “pistolas laser” de los cuerpos de seguridad para determinar la velocidad de los vehículos en tráfico rodado, como alternativa a los sistemas radares.

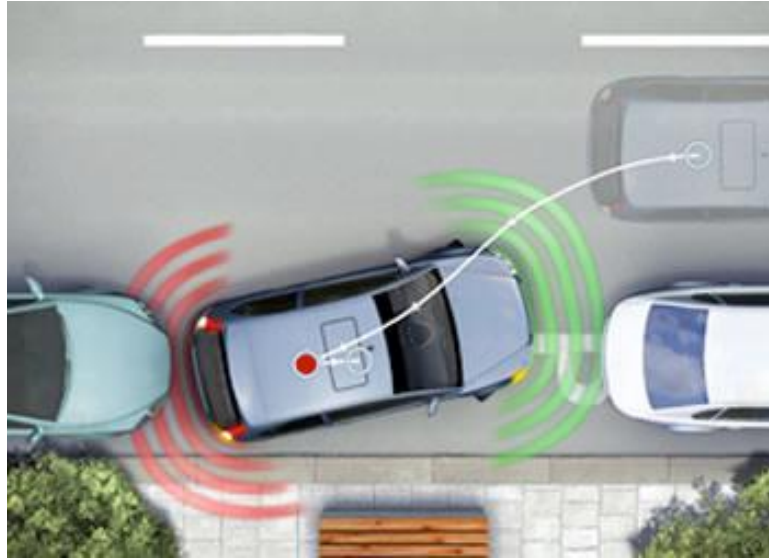


*Figura 1. Fuerzas del estado usando "pistola laser"*

Las ventajas del LIDAR con respecto al radar son bastantes, de las cuales las más importantes serían:

- Es más rápido. Es capaz de obtener la velocidad de los vehículos en 0.3 segundos.
- Tamaño muy conveniente. Es fácil de manejar, transportar y mantener.
- Económico. Más barato que un radar.

Otro ejemplo de este tipo de utilidad de los detectores de objetos, relacionado con el tema del trabajo, son los sistemas de aparcamiento asistido. Estos se basan en el uso de radares, generalmente por ultrasonidos, para hacer más fácil la maniobra de aparcamiento. El primero de su tipo se basaba en una alarma sonora, que, variando su intensidad, indicaba la proximidad de los obstáculos. A medida que se fue perfeccionando ya hay vehículos que proyectan una imagen virtual en el salpicadero de los alrededores del vehículo.



*Figura 2. Ejemplo de vehículo con sensores ultrasónicos*

Las ventajas del lidar frente a estos sistemas ultrasónicos son:

- Velocidad. El lidar al ser un haz laser se propaga a la velocidad de la luz, mucho mas alta que la velocidad del sonido.
- Rango. Los sistemas ultrasónicos tienen un rango mucho más limitados, debido a la dispersión del sonido frente a la luz.

## ***1.2 MOTIVACIÓN DEL PROYECTO***

---

Una parte muy grande de la investigación automovilística consiste en I+D para reducir los accidentes de vehículos motorizados y de esta manera reducir la mortalidad de este tipo de medio de transporte. A lo largo de los últimos años, gracias precisamente a este tipo de investigación, los accidentes se han reducido de manera considerable, pero todavía no estamos en una situación de seguridad absoluta, por lo que todo avance en este campo, por pequeño que sea, puede suponer una gran diferencia.

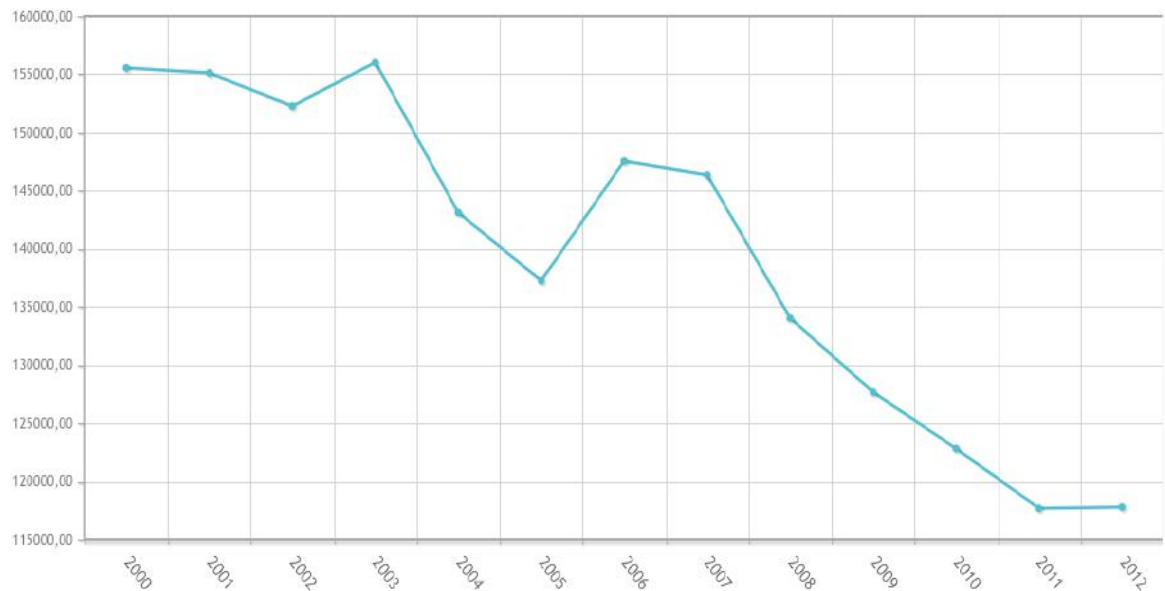


Figura 3. Gráfico evolución número de accidentes por año (INE)

Según datos de la DGT, la mayor parte de los accidentes se deben a errores humanos ocasionando que el vehículo colisione con otros vehículos y, en casos mas reducidos, contra obstáculos debido a distracciones o falta de atención en la carretera. Es por esto que los vehículos a tracción tienen una de las mortalidades más altas de todos los medios de transporte.

	Accidentes con víctimas	Fallecidos	Heridos graves	Heridos leves
<b>Colisión frontal</b>	2.661	222	787	4.050
<b>Colisión lateral y frontolateral</b>	25.002	246	2.316	32.889
<b>Colisión trasera y múltiple</b>	21.617	153	1.181	34.562
<b>Salidas de vía</b>	16.388	508	2.170	19.276
<b>Vuelco</b>	3.334	30	434	3.317
<b>Atropello a peatón</b>	11.026	349	1.893	10.207
<b>Otro tipo</b>	9.491	172	1.305	10.333

Tabla 1. Numero de accidentes en 2013 (DGT)





La intención de este proyecto es intentar reducir el numero de accidentes mejorando la seguridad en carretera, ya que avisando al conductor con una antelación de 40 metros hay suficiente tiempo de reacción para evitar muchas de estas colisiones.

Mediante la instalación de pequeños sensores alrededor del vehículo seriamos capaces de, en una pantalla LCD, dibujar una imagen virtual de los alrededores del vehículo e, incluso, determinar a que velocidad están variando esas distancias entre vehículos, reduciendo el numero de veces que el conductor debiera desviar la vista de la carretera para hacerse una idea de su situación momentánea.

### ***1.3 OBJETIVOS***

---

---

El proyecto busca cumplir una serie de objetivos:

- Montar un sistema electrónico conectando un microprocesador Arduino UNO, un sensor laser LIDAR-LITE y una serie de componentes para complementar funciones como botones, resistencias y un altavoz para alarma sonora.
- Realizar un programa que procese la información del sensor y la imprima en una pantalla LCD, además de, cuando solicitado, calcule la velocidad basándose en las lecturas reiteradas y envíe un pulso a un altavoz para notificar la proximidad de un vehículo.



## ***1.4 METODOLOGÍA / SOLUCIÓN DESARROLLADA***

---

Para cumplir los objetivos se llevarán a cabo las siguientes tareas:

- Montaje y conexión del cableado.
- Comprobar, con el monitor serie incorporado en el programa Arduino, que el sensor lidar funciona correctamente.
- Conectar la pantalla LCD y programarla para empezar a imprimir la distancia.
- Conectar el pulsador y reprogramar el arduino para indicarle al microprocesador que cambie de función y empiece a imprimir la velocidad.
- Conectar la alarma sonora para que, una vez sobrepasado un limite determinado, empiece a sonar.

## ***1.5 RECURSOS / HERRAMIENTAS EMPLEADAS***

---

En este trabajo ha sido necesario el uso de recursos de hardware y de software. Se listan a continuación:

- HARDWARE:
  - Microprocesador Arduino UNO, equipado con un microcontrolador ATmega328P.
  - Sensor laser LIDAR-Lite.
  - Placa base para conexión.
  - Resistencias. Una de 220  $\Omega$  y otra de 10 k $\Omega$ .
  - Potenciómetro de 10 k $\Omega$ .



- Pantalla LCD “LCM1602C” de 16x2 caracteres (Columnas x Filas).
  - Pulsador.
  - Fuente de alimentación, tanto una batería como a través del cable USB.
- SOFTWARE:
- Software Arduino.
  - Librería I<sup>2</sup>C.

El sensor láser LIDAR-Lite está desarrollado por la empresa PulsedLight, la versión utilizada es la primera, pues ahora el que se encuentra en el mercado es la nueva versión “Blue Label”, de similares características técnicas e idénticas funciones. Asimismo tanto el microprocesador como el software utilizado para programarlo están desarrollados por ARDUINO (USA) / GENUINO (Resto del mundo) y son una plataforma de prototipo open-source de fácil uso, incluso para gente sin apenas experiencia en el mundo de la electrónica.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Introducción

---



## Capítulo 2 SOLUCIÓN DESARROLLADA

### 2.1 *HARDWARE*

---

Como se menciona en la Sección 1.5 el hardware de este proyecto está compuesto por tres componentes principalmente: el sensor lidar, la pantalla LCD y el microprocesador que controla el funcionamiento del dispositivo.

El sensor se encargará de calcular la distancia a los objetos de manera continua, actualizándola cada medio segundo y la pantalla LCD será la interfaz incorporada al salpicadero/ordenador de a bordo del vehículo para la interacción dispositivo/conductor.

#### 2.1.1 SENSOR LIDAR-LITE

---



*Figura 4. Sensor LIDAR-LITE*



Como se mencionó ligeramente en la Sección 1.1 LIDAR es una tecnología que permite determinar distancias desde un emisor a un objeto haciendo uso de un haz láser pulsado.

### ***2.1.1.1 Funcionamiento***

La forma de determinar la distancia es midiendo de manera precisa el tiempo entre la emisión del pulso lumínico y la detección de la señal reflejada. El sensor tiene implementada una técnica que permite una precisión de hasta 1 cm mediante la digitalización y estudio de dos señales, una señal de referencia detectada directamente desde el emisor antes de calcular la distancia y de recibir la señal reflejada desde el objeto a estudiar.

El sistema estima el retraso entre las dos señales mediante una técnica de estudio de señales conocida como correlación, que consigue determinar si efectivamente la señal recibida es la reflejada de la emitida y no una interferencia de otras posibles fuentes lumínicas. Una vez analizadas las dos señales y medido el tiempo entre ambas, conociendo la velocidad de la luz es fácil calcular la distancia.

El principal motivo por el cual se decidió usar este sensor es el hecho de que esta adaptado para conectarse al Arduino directamente y funcionar inmediatamente. Esta diseñado para todo tipo de proyectos de electrónica, robótica y automática, con innumerables posibilidades. Es un dispositivo de muy fácil acceso, económico y del que se puede obtener mucha información en su pagina web oficial (<https://www.pulsedlight3d.com/products/lidar-lite-v2-blue-label.html>) , así como ejemplos y modelos de uso en una pagina web que esta conectada con la anterior (<https://github.com/pulsedlight3d>). En esta última se encuentra todos los ejemplos para empezar a familiarizarse con el sensor.

Por la conexión decidida, para poder utilizar este sensor necesitaremos añadir la biblioteca I<sup>2</sup>C a nuestro programa Arduino. El link para encontrarla es: (<http://www.dsscircuits.com/index.php/articles/66-arduino-i2c-master-library>)

### 2.1.1.2 Esquema de conexión:

El esquema de conexión es el siguiente:

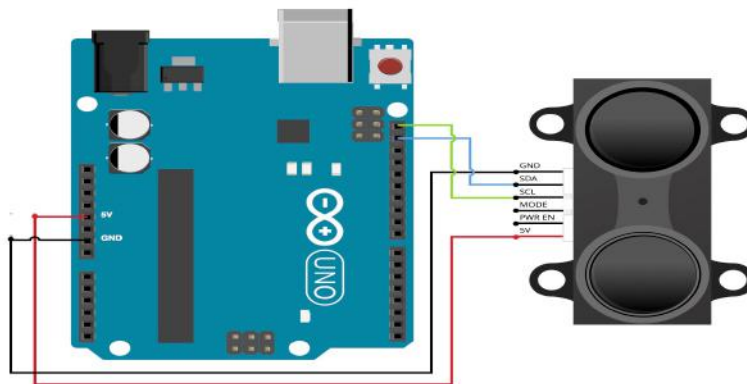


Figura 5. Esquema de conexión LIDAR-Lite y Arduino UNO (Manual de usuario del lidar)

### 2.1.1.3 Diagrama de bloques

El diagrama de funcionamiento del sensor, mediante el uso de los bloques que lo componen, es:

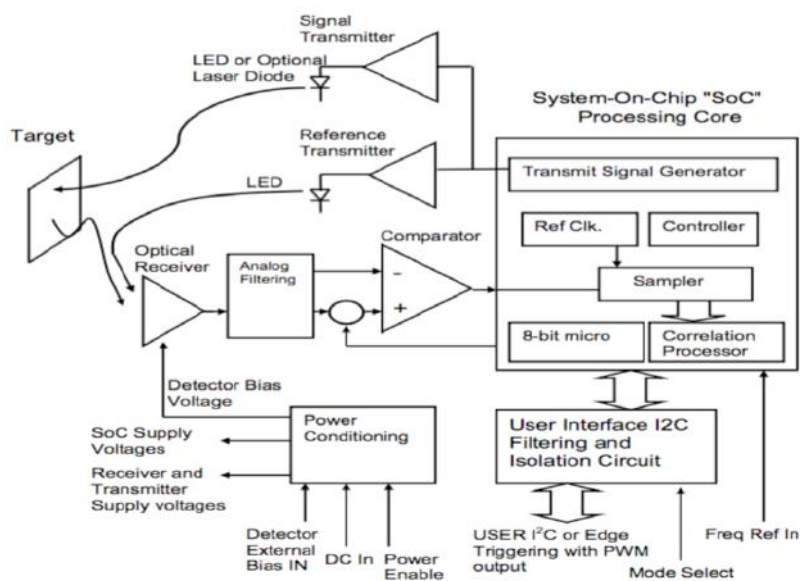


Figura 6. Diagrama de bloques del sensor (Manual de Usuario del lidar)



### 2.1.1.4 Datos técnicos e información detallada

En la Parte VCapítulo 1 se encuentra la Datasheet del sensor para cualquier duda.

### 2.1.2 PANTALLA LCD LCM1602C

---

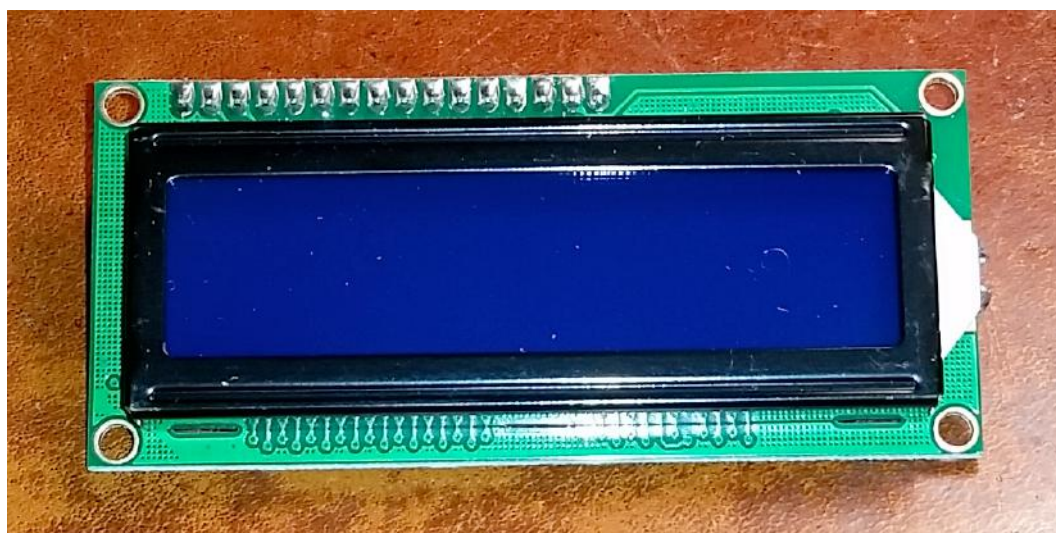


Figura 7. Pantalla LCD LCM1602C [Parte Frontal]

La pantalla de cristal líquido (LCD por sus siglas en inglés, *liquid crystal display*) es donde la información calculada será proyectada. El motivo para usar una pantalla de este tipo es que utilizan cantidades de energía eléctrica mínimas, esto la hace muy utilizada en dispositivos electrónicos de pilas.

#### 2.1.2.1 Funcionamiento

Las pantallas LCD funcionan cambiando las propiedades eléctricas de un líquido contenido entre cristales polarizados. Estos cristales solo dejan pasar ciertos tipos de luz. Cuando el líquido entre los cristales se carga eléctricamente empieza a pasar a un estado de semi-solidez. Este nuevo desvía la luz en una dirección diferente a la del cristal polarizado, bloqueando el paso de luz a través



de determinados sitios. De esta forma, se consigue crear los caracteres que se ven en la pantalla.

La pantalla LCD en cuestión es de un funcionamiento muy básico, permitiéndonos mostrar texto y variables de manera sencilla pero eficaz. Para empezar a programarla, necesitamos hacer uso de la biblioteca, ya incluida en la plataforma Arduino, llamada “LiquidCrystal” (escrita por Limor Fried) que nos permite controlar pantallas LCD compatibles con el driver Hitachi HD44780.

### 2.1.2.2 Esquema de conexión:

La pantalla LCD se puede conectar casi directamente al microprocesador, los únicos elementos extra que tenemos que añadir consisten en un potenciómetro de 10 k $\Omega$ , que nos permitirá regular el contraste y una resistencia de 220  $\Omega$  para evitar dañar la pantalla.

El esquema de conexión final para la pantalla LCD es:

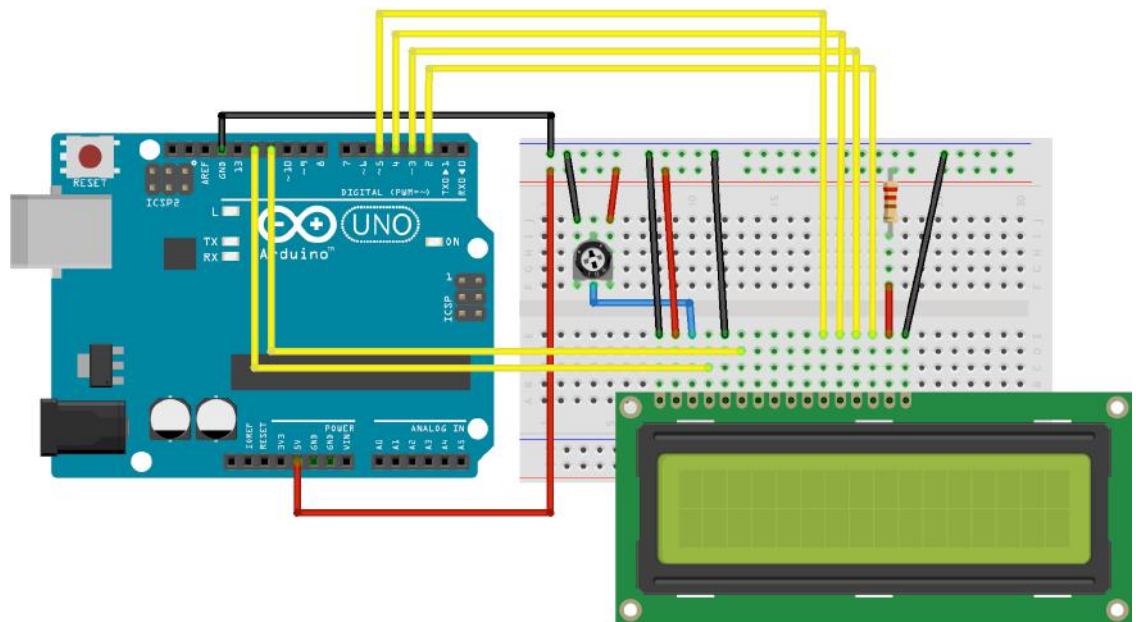


Figura 8. Esquema de conexión LCD (Página web oficial Arduino)

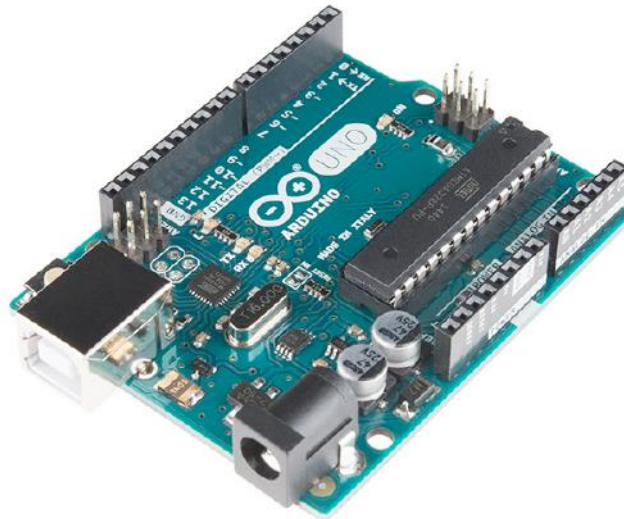


### 2.1.2.3 Datos técnicos e información detallada

En la Parte VCapítulo 2 se encuentra la Datasheet de la pantalla para cualquier duda.

### 2.1.3 MICROCONTROLADOR ARDUINO UNO

---



*Figura 9. Microcontrolador Arduino*

El microcontrolador se es el dispositivo que conecta todos los demás elementos entre si, una vez cargado el código, se encargará de que nuestro sistema finalizado cumpla las funciones que hemos diseñado para el.

Hay muchos microcontroladores en el mercado, el motivo por el cual hemos elegido precisamente Arduino UNO es su facilidad de uso, la versatilidad del software para trabajar con él y la buena compatibilidad entre el lidar y específicamente las placas Arduino. Además de todo esto es una opción económica y robusta, convirtiéndolo en el candidato ideal para un proyecto como en el que se está trabajando.



### ***2.1.3.1 Funcionamiento***

Arduino/Genuino Uno es un microprocesador basado en un microcontrolador ATmega328P. Tiene 14 pines para entrada/salida digitales, de los cuales 6 pueden ser utilizados como salidas PWM (modulación por ancho de pulsos, por las siglas del inglés *Pulse Width Modulation* y 6 entradas analógicas.

Para su programación solo hace falta descargar el software gratuito que se puede encontrar en el siguiente link (<https://www.arduino.cc/en/Main/Software>).

### ***2.1.3.2 Datos técnicos e información detallada***

En la Parte VCapítulo 3 se encuentra la Datasheet del microprocesador para cualquier duda.

## **2.1.4 ESQUEMA DE CONEXIÓN**

---

En los apartados anteriores se han descrito las conexiones de cada aparato de forma individual, a continuación se muestra el esquema de todos los componentes interconectados. Cabe destacar que, por sencillez, se ha modificado el orden de los pines de la pantalla LCD con respecto a la realidad, de manera que el esquema tenga un orden más lógico, haciéndolo más entendible.

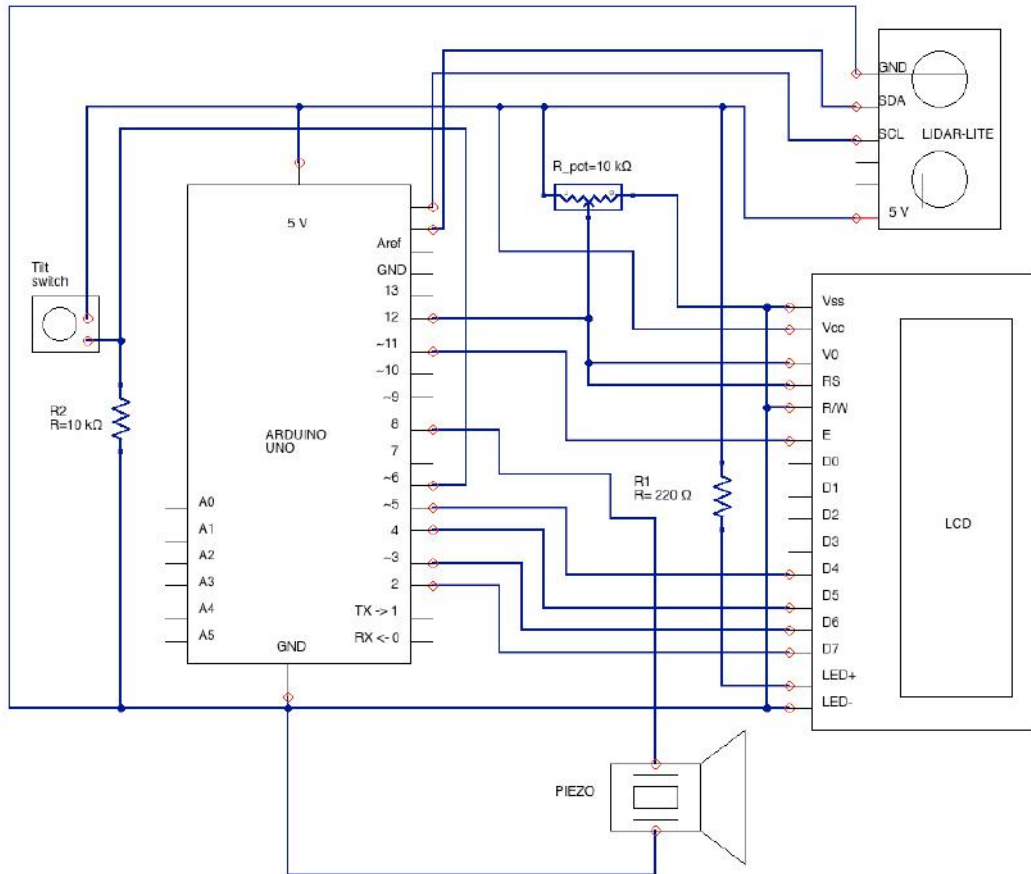


Figura 10. Esquema montaje completo

## 2.1.5 RESULTADO FINAL

A continuación se muestra una figura mostrando el prototipo finalizado, con todos los componentes conectados y listo para la implementación en el medio deseado.

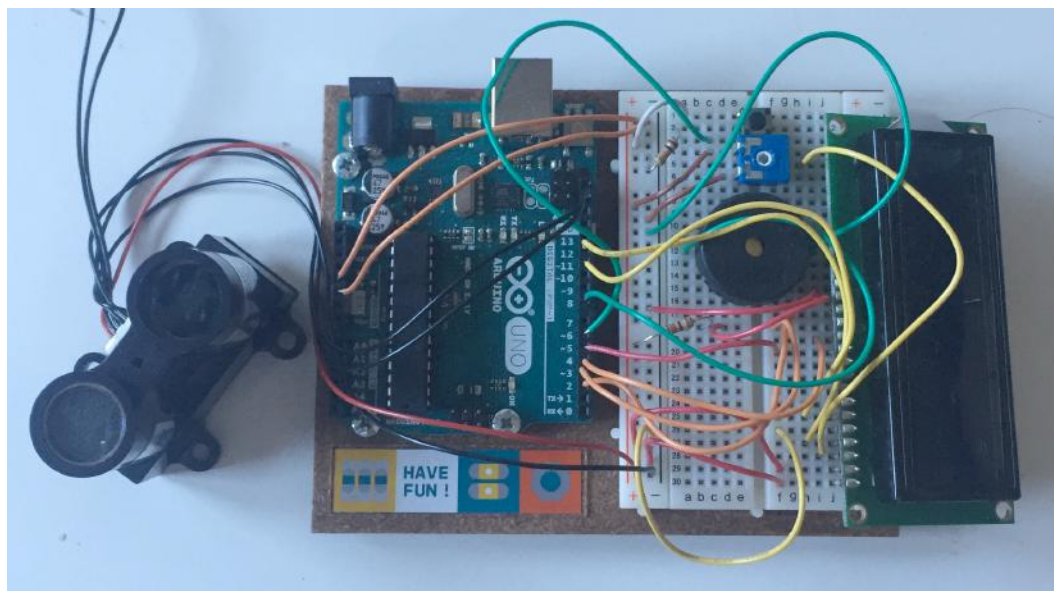


Figura 11. Prototipo finalizado

## 2.2 SOFTWARE

En esta parte del trabajo se explicará el software programado en el Arduino que regula el funcionamiento del prototipo, incluyendo notas acerca de cómo proceder para soliviantar las asunciones que se han tenido que hacer por motivos demostraciones y deben ser modificadas para implementación en sistemas reales.

Se harán continuas referencias al código programado, que puede ser encontrado en la Parte IV Capítulo 1.

### 2.2.1 CONTROL DEL SENSOR LIDAR-LITE

Como se mencionó en el apartado 2.1.1.1 para la conexión entre el lidar y el Arduino hemos hecho uso de la tecnología I<sup>2</sup>C (en inglés *Inter-Integrated Circuit*), que consiste en un bus de datos serial desarrollado en 1982 por *Philips Semiconductors* (hoy *NXP Semiconductors*).

Se utiliza normalmente para la comunicación interna entre diferentes partes de un circuito, en nuestro caso entre el Arduino y el sensor lidar. El I<sup>2</sup>C consiste en un bus maestro-esclavo. El maestro es siempre quien inicializa la transferencia de datos y el esclavo reacciona a ella.

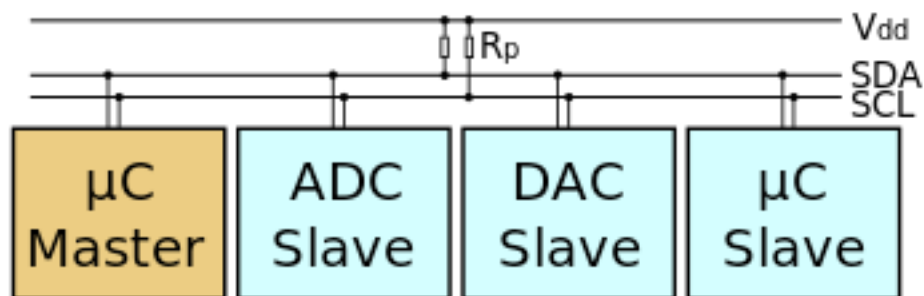


Figura 12. Ejemplo esquemático de un bus I<sup>2</sup>C con un maestro y varios esclavos

La propiedad fundamental del I<sup>2</sup>C, como se puede apreciar en la Figura 12. Ejemplo esquemático de un bus I<sup>2</sup>C con un maestro y varios esclavos es que un microcontrolador puede controlar toda una red de circuitos integrados con solo dos pines I/O (*Input/Output*) y un software muy simple. Solo se precisan dos líneas de señal: el reloj (SCL por las siglas en inglés de *Serial Clock*) y la línea de datos (SDA por las siglas en inglés de *Serial Data*).

Inicialmente la dirección del I<sup>2</sup>C estándar era el primer Byte que se enviaba, específicamente los primeros 7 bits representaban la dirección y el octavo era el R/W-Bit (Bit de lectura o escritura). El hecho de que la dirección era determinada por 7 bits significaba que solo se permitían hasta 112 nodos por bus. Después se añadió a la dirección 3 bits más, lo que permitía que la dirección quedase determinada por 10 bits, permitiendo la conexión de hasta 1136 nodos por bus.

El inicio de la transmisión viene indicado por una señal emitida por el maestro, acompañada de la dirección del esclavo que se desea activar. Después el ACK-Bit del esclavo correspondiente confirma que la dirección es correcta. En función del R/W-Bit se envían datos hacia el esclavo o se reciben datos del mismo.



Por tanto en la primera parte del código activamos el bus I<sup>2</sup>C que en el caso del Arduino. Para ello hacemos uso de las funciones incluidas en la biblioteca mencionada anteriormente. Las que hemos usado son:

- `I2c.begin()`; Abre y se une al bus como maestro.
- `I2c.timeOut()`; Establece un tiempo de espera por si acaso falla la comunicación .
- `I2c.write()`; Escribe datos en el nodo esclavo.
- `I2c.read()`; Lee datos desde el nodo esclavo.

### 2.2.2 CONTROL DE LA PANTALLA LCD

---

Para poder usar y controlar la pantalla LCD tenemos que incluir la biblioteca LiquidCrystal, incluida en el software Arduino. Para la programación de la pantalla LCD solo es necesario el uso de algunas funciones incluidas en la librería previamente mencionada, las que hemos usado son:

- `LiquidCrystal lcd()`; Establece un nombre para el uso de las funciones de la librería e indica que pines del microprocesador se corresponden con ella.
- `Lcd.begin()`; Empieza la comunicación entre el microprocesador y la pantalla.
- `Lcd.clear()`; Deja la pantalla en blanco.
- `Lcd.setCursor()`; Determina las coordenadas donde empezar a actuar en la pantalla LCD.
- `Lcd.print()`; Escribe en la pantalla la información deseada.

### 2.2.3 PROGRAMACIÓN DE LA ALARMA POR PROXIMIDAD

---

Como medida extra de seguridad se ha implementado una alarma sonora que empieza a sonar en el momento en que el sensor detecta que la distancia al

---





vehículo es menor que una distancia de seguridad previamente configurada en el código. Para propósitos demostrativos esta se ha configurado en 1 metro.

Para ello simplemente hemos conectado un altavoz piezoeléctrico al microprocesador y mediante la comparación entre la variable “*distance*” y la “*dangerousDistance*” se envía un tono para que empiece a sonar. Las funciones utilizadas para ello son:

- `tone()`; Envía un tono al altavoz
- `notone()`; Desactiva el tono hacia el altavoz

## 2.2.4 OPERACIONES REALIZADAS

---

Dada la potencia del sensor lidar utilizado una vez programado el sensor las operaciones necesarias para el desarrollo del programa se ven verdaderamente simplificadas. A continuación están explicadas todas las operaciones que hemos realizado, utilizando los mismos nombres que en el código adjunto para mayor facilidad en la comprensión:

- Para la comodidad del usuario lo más útil es que la distancia representada esté en metros, como el sensor da la información en centímetros, la primera operación es un cambio de variable.

$$E. 1 \quad \text{distance}(m) = \frac{\text{distance}(cm)}{100}$$

- Para calcular la velocidad del objeto lo primero es calcular, basado en la velocidad del usuario, cuanto ha avanzado el sensor en el tiempo en que se mide las distancias. Para ello tenemos que realizar un cambio de variable y después multiplicarlo por el tiempo entre mediciones, que hemos estipulado como medio segundo.

$$E. 2 \quad \text{myadvance}(cm) = \text{myspeed}(km/h) * \frac{100000}{3600} * 0.5(s)$$





- Una vez se ha calculado el avance del sensor, calcular la velocidad del objeto aproximándose es usar la ecuación que relaciona espacio, velocidad y tiempo. Después un nuevo cambio de variable para que el resultado este en metros por segundo.

$$E. 3 \quad \text{hisspeed}(m/s) = \frac{(\text{prevdistance}(cm) + \text{myadvance}(cm) - \text{distance}(cm))}{0.5} * \frac{1}{100}$$



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Solución Desarrollada

---



## Capítulo 3 RESULTADOS/EXPERIMENTOS

Para comprobar el correcto funcionamiento de todos los elementos necesarios para el desarrollo del proyecto se han ido haciendo pruebas a lo largo de todo el montaje, configurando elemento a elemento y comprobando que, independientemente, todos cumplieran su función. Finalmente se ha realizado un último experimento con el prototipo ya finalizado y con todos los elementos conectados entre sí.

### *3.1 MEDICIONES DE DISTANCIA*

---

La primera prueba realizada, nada más empezar el proyecto, fue comprobar que el código utilizado para programar el sensor lidar funcionaba correctamente. Para ello, se utilizó el diagrama de conexión expuesto en la Figura 5. Esquema de conexión LIDAR-Lite y Arduino UNO (Manual de usuario del lidar) Una vez instalado y programado el sistema, se comprobó a través del monitor serie incorporado en el programa Arduino que el sensor midiese correctamente las distancias.

Para ello, se empezó midiendo la distancia del sensor al techo de la habitación y después colocando objetos en el rango y desplazándolos de manera vertical. Ayudándose de una cinta métrica se fueron contrastando los datos mostrados.

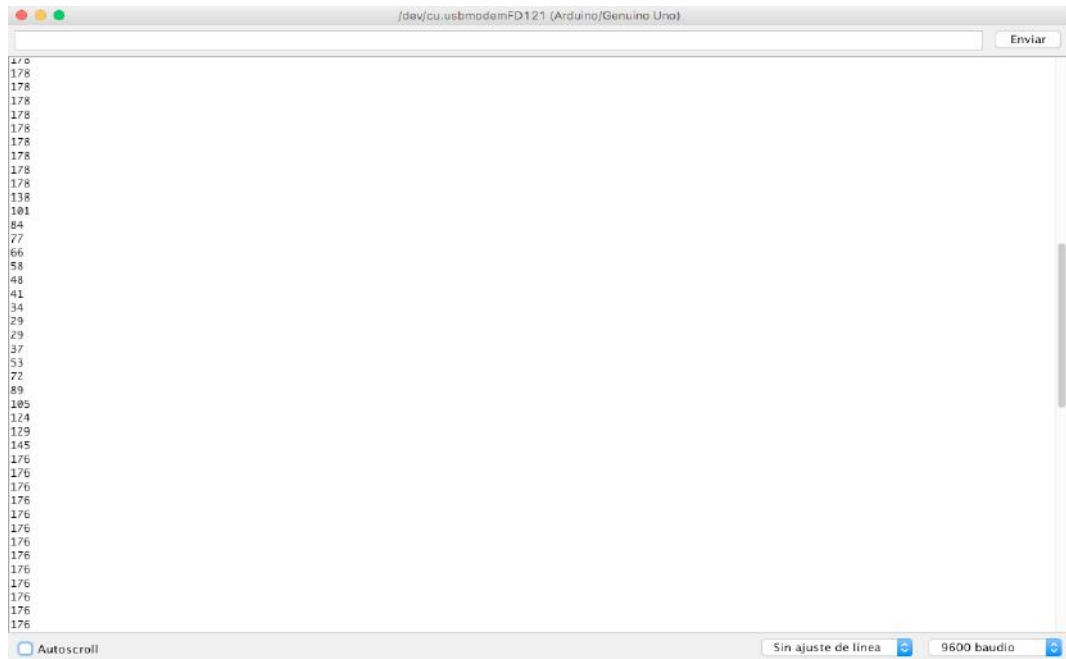


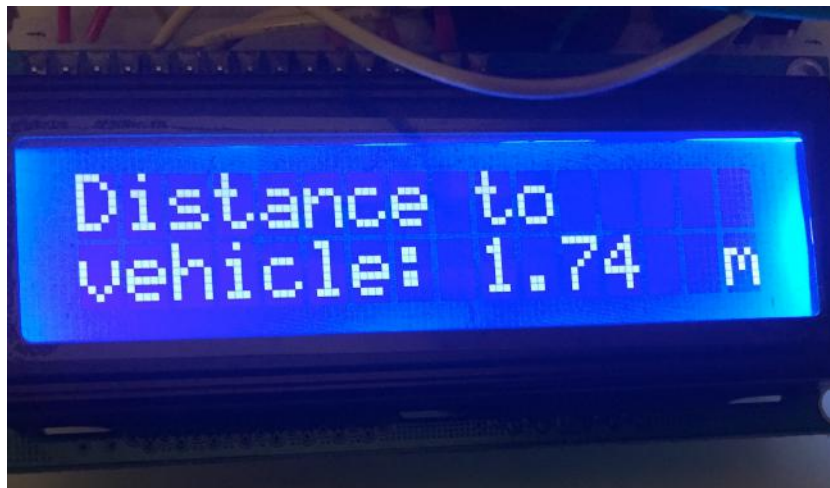
Figura 13. Pantalla del monitor serie durante experimento

Una vez finalizado el primer experimento se comprobó que el sensor funcionaba correctamente y que el margen de error (1 cm) era el esperado y aceptable para la aplicación a estudiar.

### 3.2 PRUEBAS DE PANTALLA

El segundo experimento fue comprobar el buen funcionamiento de la pantalla LCD. Para ello se instaló siguiendo el esquema expuesto en la Figura 8. Esquema de conexión LCD (Página web oficial Arduino) Una vez conectada se programó para que se leyese la frase “Hola mundo”.

Una vez comprobado que la conexión y la programación eran correctas se programó para que proyectase la distancia calculada previamente por el sensor, y se comparó con los datos expuestos en el monitor serie.



*Figura 14. Pantalla LCD mostrando la distancia*

Finalizado el experimento se comprobó que la pantalla mostraba la misma cifra que el monitor serie, actualizándose la información cada vez que variaba la distancia detectada por el sensor.

### **3.3 CÁLCULO DE VELOCIDAD**

---

El siguiente paso dado fue comprobar que la segunda función programada en nuestro código, el cálculo de la velocidad, funcionase.

Para ello primero se instaló el pulsador que regula qué función se desea realizar en cada momento. Una vez realizadas todas las conexiones y programado el microprocesador se realizó el experimento moviendo objetos de forma vertical y viendo que la velocidad mostrada en la pantalla era acorde con la realidad. En el prototipo, y por motivos meramente demostrativos, la velocidad no se muestra en kilómetros por hora, la opción mas conveniente para el usuario, sino en metros por segundo. A la hora de una posible implementación en un sistema real, haría falta modificar el cambio de variable en la ecuación E. 3.

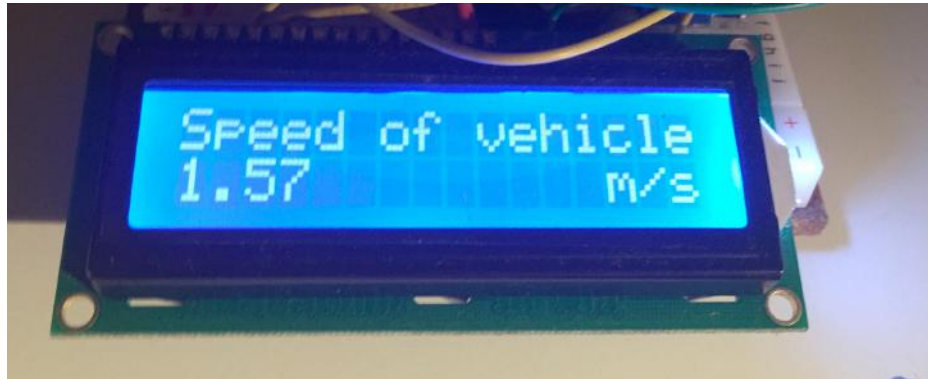


Figura 15. Pantalla LCD mostrando la velocidad como elemento fijo

Posteriormente se repitió el experimento asumiendo que el sensor se mueve a una velocidad de 100 kilómetros por hora, para comprobar si a partir de velocidades relativas era capaz de calcular la velocidad absoluta del vehículo. Para ello simplemente se modificó el valor de la variable “*myspeed*” a 100. Y se observó lo que se leía en la pantalla. En una futura aplicación del dispositivo habría que conectar la variable en cuestión con el velocímetro del vehículo, para asegurar el correcto funcionamiento del sistema.

Al final este experimento se comprobó que el cálculo de velocidades era acertado en ambos casos, tanto si el lidar estaba fijo como si se consideraba en movimiento.

### 3.4 SISTEMA DE ALERTA

El último experimento realizado fue el de comprobar el estado del sistema de alarma. Para ello y, de nuevo únicamente por motivos demostrativos, se programó la distancia de seguridad de 1 metro. Esto es fácilmente modificable, tan solo es cambiar el valor de la variable “*dangerousDistance*” al valor deseado.

Se comprobó que, efectivamente, cuando la pantalla mostraba un valor de distancia igual o menor a 1 metro la alarma empezaba a sonar independientemente de en qué función estuviera.



## **Capítulo 4 CONCLUSIONES**

Se ha diseñado, construido y programado un sistema capaz de medir distancias en tiempo real y a partir de estas calcular la velocidad instantánea de los objetos o vehículos alrededor del usuario.

Sin embargo, a pesar de haberse cumplido todos los objetivos impuestos en este proyecto, y técnicamente estando el prototipo preparado para la implementación en sistemas reales, para poder aprovechar las posibilidades generadas por este dispositivo todavía hace falta trabajar un poco más. Un único sensor, como diseñado para el prototipo, no es suficiente para resultar en el aumento en la seguridad intencionado. Sería necesaria la implementación de sensores todo alrededor del vehículo. Además sería necesario ajustar un poco el código para que el microprocesador fuese capaz de controlar la información de múltiples sensores al mismo tiempo. Por último realizar pruebas en un entorno más real, implementándolo en vehículos reales, objetivo que escapa al alcance de este trabajo.

A pesar de que el prototipo construido para el proyecto tiene funcionalidad limitada, se puede considerar un éxito, pues la parte más compleja funciona correctamente y con no muchos desarrollos futuros la implementación puede llegar a ser tal que se cumpla la función con la cual fue empezado este trabajo.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Conclusiones

---





## **Capítulo 5 FUTUROS DESARROLLOS**

A pesar de que en este proyecto se han cubierto todos los objetivos impuestos originalmente, en el campo de la seguridad vial siempre hay avances que hacer.

Desarrollando un programa más complejo y usando componentes más capacitadas que las del prototipo utilizado se pueden hacer mediciones más precisas o con un rango todavía mayor que los 40 metros de limitación del sensor utilizado.

Para el prototipo no se ha utilizado más que un sensor, pero una vez implementado en vehículos se puede hacer uso de muchos sensores todos comunicados de tal manera que en una pantalla LCD incorporada en el salpicadero del vehículo se pueda obtener, en tiempo real, una imagen virtual de los alrededores del vehículo. Las posibilidades de esto serían ilimitadas, desde perfección de sistemas que ya existen, como pueden ser los de frenado de emergencia en caso de posible colisión o aparcamiento asistido, hasta estar un paso más cerca de vehículos capaces de conducirse automáticamente sin intervención humana.

Todo esto en un intento de acercarnos a un futuro con cada vez menos accidentes viales, aumentando la seguridad de las calles y carreteras, tanto para los conductores como los peatones.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Futuros desarrollos

---



## BIBLIOGRAFÍA

- [1] Dirección General de Tráfico (2013) *Principales cifras de la siniestralidad vial*. Recuperado de [http://www.dgt.es/Galerias/seguridad-vial/estadisticas-e-indicadores/publicaciones/principales-cifras-siniestralidad/Siniestralidad\\_Vial\\_2013.pdf](http://www.dgt.es/Galerias/seguridad-vial/estadisticas-e-indicadores/publicaciones/principales-cifras-siniestralidad/Siniestralidad_Vial_2013.pdf)
- [2] Instituto Nacional de Estadística (INE). <http://www.ine.es/jaxi/Tabla.htm?path=/t10/a109/a04/10/&file=r70062.px&L=0>
- [3] Pulsed Light, fabricantes del Lidar-Lite. <https://www.pulsedlight3d.com/products/lidar-lite-v2-blue-label.html>
- [4] Arduino/Genuino. <https://www.arduino.cc>



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Bibliografía

---



***Parte II ESTUDIO***

***ECONÓMICO***



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Mediciones

---



## Capítulo 1 MEDICIONES

### 1.1 COMPONENTES

Componente	Cantidad
Sensor LIDAR-Lite	1
Microprocesador Arduino	1
Pantalla LCD LCM1602C	1
Potenciómetro 10 k $\Omega$	1
Pulsador	1
Resistencias ohmicas	2
Altavoz piezoeléctrico	1

Tabla 2. Medición de componentes

### 1.2 SOFTWARE

Nombre	Horas de uso
Entorno de programación Arduino	180

Tabla 3. Medición de software



---

### ***1.3 MANO DE OBRA DIRECTA***

---

<b>Actividad</b>	<b>Horas</b>
Estado del arte	20
Búsqueda y elección de equipos	40
Búsqueda y elección de software	1
Montaje	10
Programación	50
Pruebas y solución de problemas	130
Documentación del proyecto	40

*Tabla 4. Medición mano de obra directa*





## Capítulo 2 PRECIOS UNITARIOS

### 2.1 COMPONENTES

Componente	Precio/Unidad (\$)
Sensor LIDAR-Lite	105.85
Microprocesador Arduino	79.99
Pantalla LCD LCM1602C	Incluido con Arduino
Potenciómetro 10 k $\Omega$	Incluido con Arduino
Pulsador	Incluido con Arduino
Resistencias ohmicas	Incluido con Arduino
Altavoz piezoeléctrico	Incluido con Arduino

Tabla 5. Precio unitario de los componentes

### 2.2 SOFTWARE

Nombre	Precio/hora (\$)
Entorno de programación Arduino	Gratuito

Tabla 6. Precio unitario de software



---

### **2.3 MANO DE OBRA DIRECTA**

---

<b>Actividad</b>	<b>Precio/hora (\$)</b>
Estado del arte	20
Búsqueda y elección de equipos	20
Búsqueda y elección de software	20
Montaje	20
Programación	20
Pruebas y solución de problemas	20
Documentación del proyecto	20

*Tabla 7. Precio unitario de mano de obra*



## Capítulo 3 SUMAS PARCIALES

### 3.1 COMPONENTES

Componente	Cantidad	Precio/Unidad (\$)	Coste total (\$)
Sensor LIDAR-Lite	1	105.85	105.85
Microprocesador Arduino	1	79.99	79.99
Pantalla LCD LCM1602C	1	0	0
Potenciómetro 10 k $\Omega$	1	0	0
Pulsador	1	0	0
Resistencias ohmicas	2	0	0
Altavoz piezoeléctrico	1	0	0
TOTAL Componentes			185.84

Tabla 8. Sumas parciales de componentes



### 3.2 SOFTWARE

Nombre	Horas de uso	Precio/hora	Coste total (\$)
Entorno de programación Arduino	20	0	0
TOTAL Software			0

Tabla 9. Sumas parciales de software

### 3.3 MANO DE OBRA DIRECTA

Actividad	Horas	Precio/Hora (\$)	Coste total (\$)
Estado del arte	20	20	400
Búsqueda y elección de equipos	40	20	800
Búsqueda y elección de software	1	20	20
Montaje	10	20	200
Programación	50	20	1000
Pruebas y solución de problemas	130	20	2600
Documentación del proyecto	40	20	800
TOTAL Mano de obra			5820

Tabla 10. Sumas parciales mano de obra directa



## Capítulo 4 PRESUPUESTO GENERAL

Concepto	Coste (\$)
Componentes	185.84
Software	0
Mano de obra	5820
<b>Total</b>	<b>6005.84</b>

*Tabla 11. Presupuesto general en dólares*

Tasa de conversión	1.1035 \$ / €
<b>Total en euros</b>	<b>5442.49</b>

*Tabla 12. Presupuesto general en euros*



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Presupuesto general

---



***Parte III MANUAL DE  
USUARIO***



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Presupuesto general

---





## Capítulo 1 MANUAL DE USUARIO 1

El presente documento está dirigido a entregar las pautas de operación del Sensor de Distancia y Velocidad para Vehículos. Este sistema permite obtener información en tiempo real de los alrededores del vehículo sin necesidad de desviar la vista de la carretera de manera reiterada.

Una vez iniciado el sistema se leerá en la pantalla LCD “*Press button to start*”. A pesar de que no se lea información en la pantalla el sistema ya está calculando la distancia de forma continua en caso de que sea necesario activar la alarma por proximidad.

Para empezar a ver información en la pantalla hace falta presionar el pulsador. La primera vez que se pulse ya aparecerá en la pantalla el mensaje “*Distance to vehicle:* ” seguido de la distancia en metros.

Si se vuelve a presionar el pulsador cambiará a la segunda función, cálculo de velocidad de aproximación. En la pantalla ahora se leerá el mensaje “*Speed of vehicle*” seguido de la velocidad a la que el sensor detecta un objeto acercándose o el mensaje “*Not approaching*” que significa que el objeto no está aproximándose, sino alejándose.

A lo largo de todas las opciones la alarma seguirá activa, ya que esta toma preferencia sobre la función en la que estemos. Por tanto, en la pantalla se seguirá viendo la función elegida, pero la alarma empezará a sonar en cuanto la distancia sea inferior a la considerada distancia de seguridad.



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Manual de usuario 1

---



## *Parte IV CÓDIGO FUENTE*



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Código fuente

---



## Capítulo 1 CÓDIGO FUENTE

```
#include <I2C.h>
#include <LiquidCrystal.h>

#define LIDARLite_ADDRESS 0x62 // Default I2C Address of LIDAR-Lite.
#define RegisterMeasure 0x00 // Register to write to initiate ranging.
#define MeasureValue 0x04 // Value to initiate ranging.
#define RegisterHighLowB 0x8f // Register to get both High and Low
bytes in 1 call.

LiquidCrystal lcd(12,11,5,4,3,2);
float myspeed=0; //Variable in km/h
int prevdistance=0;
const int switchPin = 6;
int switchState = 0;
int prevSwitchState = 0;
int option = 0;
int dangerousDistance = 100;
void setup(){
    Serial.begin(9600); //Opens serial connection at 9600bps.
    I2c.begin(); // Opens & joins the irc bus as master
    delay(100); // Waits to make sure everything is powered up before
sending or receiving data
    I2c.timeOut(50); // Sets a timeout to ensure no locking up of sketch if
I2C communication fails
    lcd.begin(16,2); // Starts connection with the LCD display
    pinMode(switchPin,INPUT);
}

void loop(){
    // Write 0x04 to register 0x00
    uint8_t nackack = 100; // Setup variable to hold ACK/NACK responses
    while (nackack != 0){ // While NACK keep going (i.e. continue polling
until sucess message (ACK) is received )
```



```
        nackack = I2c.write(LIDARLite_ADDRESS,RegisterMeasure, MeasureValue);
// Write 0x04 to 0x00
        delay(1); // Wait 1 ms to prevent overpolling
    }

    byte distanceArray[2]; // array to store distance bytes from read
function

    // Read 2byte distance from register 0x8f
    nackack = 100; // Setup variable to hold ACK/NACK responses
    while (nackack != 0){ // While NACK keep going (i.e. continue polling
until sucess message (ACK) is received )
        nackack      =      I2c.read(LIDARLite_ADDRESS,RegisterHighLowB,      2,
distanceArray); // Read 2 Bytes from LIDAR-Lite Address and store in
array
        delay(1); // Wait 1 ms to prevent overpolling
    }
    int distance = (distanceArray[0] << 8) + distanceArray[1]; // Shift
high byte [0] 8 to the left and add low byte [1] to create 16-bit int

    if (distance <= dangerousDistance){
        tone(8,440);
    }else{
        noTone(8);
    }
    if(abs(distance - prevdistance) <= 1){
        distance = prevdistance;
    }
    float distancem = (float)distance/(float)100; //Calculate the distance
in meters for printing
    Serial.println(distance); //Print in the serial monitor

    // Set the function of the button:

    switchState = digitalRead(switchPin);
    if (switchState != prevSwitchState) {
```

---



```
if (switchState == 0) {
    if (option == 2){
        option = 1;
    } else {
        option = option + 1;
    }
}
}

switch(option){
    // Print intro:
    case 0:
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Press button");
        lcd.setCursor(0,1);
        lcd.print("to start");
        break;
    // Print Distance
    case 1:
        lcd.clear();
        lcd.setCursor(0,0);

        lcd.print("Distance to");
        lcd.setCursor(0,1);
        lcd.print("vehicle:");
        lcd.setCursor(9,1);
        lcd.print(distancem);
        lcd.setCursor(15,1);
        lcd.print("m");
        break;
    // print speed
    case 2:
        float myadvance = (((float)myspeed * (float)100000)/(float)3600) *
(float)0.5;
```



```
float  hisspeed = (((float)prevdistance + (float)myadvance -
(float)distance)/(float)100)/(float)0.5;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Speed of vehicle");

lcd.setCursor(0,1);

//lcd.print("vehicle:");

//lcd.setCursor(9,1);

if (hisspeed < 0){

    lcd.print("Not approaching");

}

else{

    lcd.print(hisspeed);

    lcd.setCursor(13,1);

    lcd.print("m/s");

}

}

prevdistance=distance;

prevSwitchState = switchState;

delay(500);

}
```





# *Parte V DATASHEETS*



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Datasheet LIDAR-LITE

---



# Capítulo 1 DATASHEET LIDAR-LITE



**PulsedLight, Inc.**  
 A NEW BENCHMARK IN OPTICAL  
 SENSOR TECHNOLOGY



**LIDAR-LITE  
 SPECIFICATIONS**

**Dimensions**  
 21 X 48.3 X 35.5 mm  
 PCB 44.5mm X 16.5mm

**Performance**  
 Range: 0-40m Laser Emitter  
 Accuracy: +/- 0.025m  
 Power: 5vdc, <100ma  
 Acquisition Time: < 0.02 sec  
 Rep Rate: 1-100Hz

**Configurations**

- Laser/PIN Diode 14mm Optics (Class I Laser Product)

**Interface**

- I2C
- PWM

US Patent: 8,125,620  
 Additional Patents Pending

## Overview

PulsedLight targets the need for high performance, very compact optical distance measurement sensors for applications such as robotics and UAV's where a very small, low-power, high performance, reduced cost optical ranging sensor is desired.

Our single chip processing solution in combination with minimal supporting hardware enables a new class of optical distance measurement sensors.

PulsedLight's goal is to make our technology available in an easily configurable sensor module that can be used as the basic building block for sensor applications in robotics, UAV and Maker projects.

## Technology

Our single board implementation uses an edge emitting, 905nm, single stripe laser. This Laser Product is designated as Class I during all procedures of operation, however operating the sensor without it's optics or housing or making modifications to the housing can result in direct exposure to laser radiation and the risk of permanent eye damage.

The standard detector is based on a Si PIN diode, but optionally, could support a Si Avalanche Photo-Diode (APD) for greater sensitivity and range. Use of an APD would require external power and temperature compensation circuitry and provisions have been made to allow for access to the detector bias input circuit.

## Technology innovations

- The use of a signature matching technique (known as signal correlation) that estimates time delay by electronically sliding a stored transmit reference over the received signal in order to find the best match.
- Operation of the laser in short bursts allowing a 100:1 advantage in peak output power over measurement systems using a continuous beam.
- A novel current driver technology with nanosecond signal transition times at high peak currents to produce high power transmit burst sequences.
- A signal processing approach implementable in a single programmable logic chip.

## Other Innovations

While not implemented in LIDAR-Lite, other innovations to be released in future products include;

- Detector switching technology allowing multiple detectors to be processed by a single signal-processing channel. Enabling compact multichannel systems.
- Multiple digital processing cores implementable in a single programmable logic chip enable use of our technology in high resolution machine vision or scanning systems.





Signal/Power Interfaces	Specifications
Power	4.7 - 5.5V DC Nominal, Maximum 6V DC
Weight	PCB 4.5 grams, Module 16 grams with optics and housing
Size	PCB 44.5 X 16.5mm, Housing 21 X 48.3 X 35.5mm
Current Consumption	<100ma continuous operation, <2ma @ 1Hz (power off between acquisitions)
Max Operating Temp.	70° C
External Trigger	3.3V logic, high-low edge triggered
PWM Range Output	PWM Signal proportional to range, 1msec/meter, 10µsec step size
I2C Machine Interface	100Kb - Fixed, 0xC4 slave address. Internal register access & control
Supported I2C Commands	Single Distance Measurement, Velocity, Signal Strength
Mode Control	Busy status using I2C, External trigger input PWM Outputs

System Parameters	Laser/Pin <sup>(1)</sup> Class I Laser Product
Transmitter	905nm, 75µm, 1watt, 8mrad, 14mm optic
Receiver	Surface mount PIN, 3° FOV wi 14mm optics
Detector Gain	1X
Max Range @ 1Hz 30% Target	30 Meters
Max Range @ 1Hz 90% target	40 Meters
Accuracy	+/- 0.025 meter
Acquisition Time	<0.02 sec
Max Rep Rate	100Hz <sup>(2)</sup>

NOTES:

1. CLASS I LASER PRODUCT CLASSIFIED EN/IEC 60825-1 2007. Complies with US FDA performance standards for laser products except for deviations pursuant to Laser Notice No. 50, dated June 24, 2007. System contains no user serviceable components. Repair or service of the system is only to be handled by factory-trained technicians. No Service by the user is allowed.
2. Higher Rep Rates have an impact on maximum range. 1Hz to 10Hz there is no change, from 10Hz to 100Hz max range will decrease until it is approximately 50% at 100Hz. Rep Rate can be dynamically configured.
3. All Operating Specifications are Preliminary.

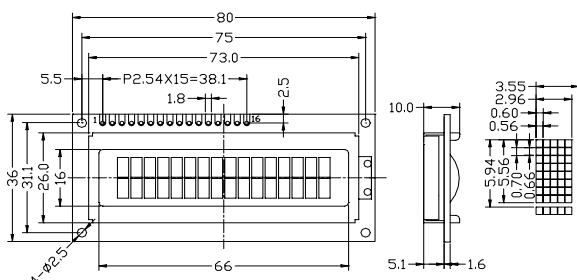


## Capítulo 2 DATASHEET PANTALLA LCD

BEIJING QINGYUN HI-TECH DEVELOPMENT CO.,LTD STANDARD CHARACTER LCM TEL/FAX: 86-10-82125078/79

### LCM1602C

#### 外形尺寸/OUTLINE DIMENSIONS



#### 机械规格和特征/MECHANICAL SPECIFICATIONS & FEATURES

Item 项目	Normal Dimensions 常规尺寸(mm)	Feature 特征
Module Size (W*H*T) 模块尺寸	80x36x10/15	LCD Type & Colour 液晶屏类型及颜色
View Area (W*H) 视域尺寸	66x16	View Angle 视角
Character Size (W*H) 字符尺寸	2.96x5.56	Display Type 显示类型
Character Pitch (W*H) 字符间距	3.55x5.94	Control IC & Package 控制芯片及封装
Character Font 字符字体	5x8	Operating Temperature 工作温度
Dot Size (W*H) 点大小	0.56x0.66	Storage Temperature 储存温度
Dot Pitch (W*H) 点间距	0.60x0.70	Backlight 背光
		STN: Yellow-Green, Gray, FSTN STN 黄绿模式、灰模式、FSTN 黑白模式
		6 O'clock、12 O'clock 6:00 或 12:00 视角
		Positive Type 正向显示
		KS0066, COB
		-20°C ~ 70°C
		-30°C ~ 80°C
		LED、EL

#### 电参数/ELECTRICAL CHARACTERISTICS (工作电流为不含背光的液晶显示模块工作电流。/Supply current doesn't contain backlight.)

Item/项目	Symbol/符号	Test Condition/测试条件	Typ./典型值	Unit/单位
Operating Voltage/工作电压	Vdd	Ta=25°C	5.0	V
Supply Current/工作电流	Idd	Ta=25°C, Vdd=5.0V	1.25	mA

#### 管脚说明/INTERFACE PIN CONNECTIONS

Pin	Symbol	Description (管脚描述)
1	Vss	Ground for Logic (地)
2	VDD	Power supply for Logic (逻辑电压)
3	VO	Power supply for LCD drive (对比度调节)
4	RS	Register selection (指令、数据寄存器选择)
5	R/W	Read/Write selection (读写选择信号)
6	E	Enable signal for LCM (使能信号)
7-14	DB0~DB7	Data Bus lines (数据线 0~7)
15	LEDA	Power supply for Backlight (+) (背光正极)
16	LEDK	Power supply for Backlight (-) (背光负极)

#### 功能特点:

- LCM1602C 显示内容 5x8 点字符点阵式, 16x2 行, 点大小 0.56x0.66 mm<sup>2</sup>, 点间距 0.04mm, 字符间距 0.59mm、0.38mm;
- 显示类型: STN 黄绿模式, 6:00 或 12:00 视角, 正向显示; STN 灰模式, 6:00 或 12:00 视角, 正向显示; FSTN 黑白模式, 6:00 视角, 正向显示;
- LED 背光或 EL 背光;
- 工作电压: 5V, 不含背光工作电流: 1.25mA (典型值);
- 工作温度: -20°C~70°C, 储存温度: -30°C~80°C; 控制器 KS0066, 芯片封装 COB。



**UNIVERSIDAD PONTIFICIA COMILLAS**  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)  
INGENIERO ELECTROMECÁNICO

Datasheet pantalla LCD

---

## Capítulo 3 DATASHEET ARDUINO

### Technical Specification

EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

#### Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

#### the board

