



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO FIN DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

**DESARROLLO DE UNA UNIDAD DE ADQUISICIÓN Y
MONITORIZACIÓN DE DATOS PARA UN VEHÍCULO**



Volumen I

Memoria técnica y Memoria económica

Autor: Oriol Bel Carbó

Director: Manuel Manzanares Brotons

Convocatòria: Mayo 2018



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

INDICE

| | |
|---|-----------|
| RESUM | 4 |
| RESUMEN | 5 |
| ABSTRACT | 6 |
| 1. PREFACIO | 7 |
| 1.1. Origen del proyecto | 7 |
| 1.2. Motivación | 8 |
| 1.3. Requerimientos previos | 9 |
| 2. INTRODUCCIÓN Y OBJETIVO | 11 |
| 2.1. Introducción | 11 |
| 2.2. Objetivo del proyecto | 13 |
| 3. ESPECIFICACIONES BÁSICAS | 14 |
| 3.1. Especificaciones técnicas | 14 |
| 3.2. Especificaciones conceptuales | 15 |
| 4. POSIBLES SOLUCIONES Y SOLUCIÓN ESCOGIDA | 16 |
| 4.1. Procesamiento de datos | 17 |
| 4.2. Obtención de datos | 19 |
| 4.3. Visualización | 21 |
| 5. DISEÑO DEL EQUIPO | 22 |
| 5.1. Diseño del Hardware | 22 |
| 5.1.1. Microcontrolador | 22 |
| 5.1.2. Sensores y acondicionamiento de señales | 28 |
| 5.1.3. Visualización | 43 |
| 5.2. Diagrama de bloques general | 45 |
| 5.3. Diseño del prototipo | 46 |
| 5.4. Conexionado en el vehículo | 47 |
| 6. DISEÑO DEL SOFTWARE | 51 |
| 6.1. Diagrama de flujo | 51 |
| 6.2. Programación App Inventor | 53 |

| | | |
|------------|---|------------|
| 7. | DISEÑO DEL PROTOTIPO DE PRUEBAS | 54 |
| 7.1. | Diseño del Hardware | 55 |
| 7.1.1. | Microcontrolador | 55 |
| 7.1.2. | Sensores y acondicionamiento | 57 |
| 7.1.3. | Visualización | 66 |
| 7.2. | Diseño de la Placa | 68 |
| 7.2.1. | Inconvenientes de diseño con software | 69 |
| 7.2.2. | Procedimiento del diseño del circuito | 70 |
| 7.2.3. | Procedimiento del diseño de la PCB | 71 |
| 8. | SIMULACIONES | 73 |
| 8.1. | Simulación del menú principal | 74 |
| 8.2. | Simulación y lectura de la Temperatura | 75 |
| 8.3. | Simulación del circuito de iluminación Luz | 76 |
| 8.4. | Simulación del reloj a tiempo real | 77 |
| 8.5. | Simulación y lectura de la distancia recorrida | 78 |
| 8.6. | Simulación del control de la carga de la batería | 79 |
| 8.7. | Simulación del registro del ciclo lectura de parámetros | 80 |
| 8.8. | Simulación de guardado en la memoria externa | 81 |
| 8.9. | Simulación del botón de retroceso | 82 |
| 8.10. | Simulación del dispositivo HC-06 | 82 |
| 9. | IMPLEMENTACIÓN | 83 |
| 9.1. | Grabado en la placa | 83 |
| 9.1.1. | Fotolito | 83 |
| 9.1.2. | Grabado | 84 |
| 9.2. | Preparación de la placa | 85 |
| 9.3. | Implementación del hardware | 88 |
| 9.3.1. | Microcontrolador | 90 |
| 9.3.2. | Implementación de los componentes | 91 |
| 10. | COMUNICACIÓN A TRAVÉS DEL HC-06 | 98 |
| 11. | NORMATIVA | 100 |
| 12. | POSIBLES MEJORAS | 101 |
| 13. | MARKETING Y VENTA DEL PRODUCTO | 102 |

| | |
|--|------------|
| 14. PLANIFICACIÓN Y DIAGRAMA GRANTT | 104 |
| 15. CONCLUSIONES | 105 |
| 16. MEMORIA ECONÓMICA | 106 |
| 16.1. Coste de los componentes | 106 |
| 16.2. Coste del prototipo | 107 |
| 16.3. Coste de una producción seriada | 108 |
| BIBLIOGRAFÍA | 109 |

Resum

El següent projecte exposa de forma detallada i precisa, el disseny, simulació i implementació de un dispositiu electrònic capaç d'adquirir una sèrie de paràmetres d'una bicicleta elèctrica i del seu entorn que pot ser del interès del usuari. Aquests paràmetres poden ser visualitzats a temps real o guardats en una memòria externa pel seu posterior tractament. El projecte presenta tots els coneixements necessaris de hardware, des de la selecció dels components fins la realització de la placa amb la descripció del procediment a seguir i els problemes a solucionar. Es presenten també coneixements de software relatius a la programació de microcontroladors i dispositius mòbils. En general es mostra tots els passos d'aprenentatge, disseny, construcció, i comercialització del producte que satisfà les necessitats de informar al usuari per que prengui les seves pròpies decisions o la monitorització d'un sistema com el control lumínic del vehicle.

Resumen

El siguiente proyecto expone de forma detallada y precisa, el diseño, simulación e implementación de un dispositivo electrónico capaz de adquirir una serie de parámetros de una bicicleta eléctrica y de su entorno que pueden ser del interés del usuario. Dichos parámetros pueden ser visualizados en tiempo real o guardados en una memoria para su posterior tratamiento. El proyecto presenta todos los conocimientos necesarios de hardware, desde la elección de los componentes hasta la realización de la placa con la descripción del procedimiento a seguir y los problemas a solucionar. Se presentan también conocimientos de software relativos a la programación de microcontroladores y de dispositivos móviles. En general se muestran todos los pasos para el aprendizaje, diseño, construcción y comercialización del producto que satisface la necesidad de informar al usuario para que tome sus propias decisiones o la monitorización de algún sistema como el control lumínico del vehículo.

Abstract

The following project exposes in a detailed and precise way, the design, simulation and implementation of an electronic device capable of acquiring a series of parameters of an electric bicycle and its environment that may be of interest to the user. These parameters can be visualized in real time or saved in a memory for further processing. The project presents all the necessary knowledge about hardware, from the choice of components to the realization of the board with the description of the procedure to be followed and the problems to be solved. Software knowledge related to the programming of microcontrollers and mobile devices is also presented. In general, all the steps for learning, design, construction and commercialization of the product that satisfies the need to inform the user to make their own decisions or the monitoring of a system such as light control of the vehicle are shown.

1. Prefacio

1.1. Origen del proyecto

El siguiente proyecto es llevado a cabo para demostrar que se han adquirido los conocimientos de la universidad suficientes para diseñar e implementar un dispositivo electrónico, así como la capacidad de aprender cualquier conocimiento relativo a la electrónica u otro campo relacionado o no con la ingeniería. Se destacan sobretodo valores y actitudes de organización, eficiencia, eficacia, responsabilidad y compaginación.



1.2. Motivación

El proyecto en cuestión engloba evidentemente una serie de conocimientos que principalmente son alrededor de la electrónica, rama de la física que se especializa en la ingeniería. Como ya es sabido para el lector, el desarrollo de la electrónica en el último siglo es exponencial y seguirá siendo la ingeniería con mayor proyección de futuro. Se expone así en este proyecto la capacidad del estudiante de, partiendo de los conocimientos adquiridos en la universidad, demostrar la capacidad de comprender y aprender nuevos retos relativos no solo a la ingeniería electrónica sino también a la informática, *marketing*, derecho, y otros ámbitos necesarios para llevar a cabo el diseño de un producto.

Se hace ímpetu también en la adquisición y práctica de valores como la organización, actitud de trabajo, responsabilidad, eficiencia, eficacia y optimización. La realización de un proyecto serio de forma consciente respecto un objetivo final.

1.3. Requerimientos previos

Es necesario advertir y destacar que los conocimientos previos aprendidos en la universidad son evidentemente de gran ayuda, y cuando más conocimiento se tiene sobre el tema o temas parecidos, más facilidad se tiene al aprender y más soltura se tiene a la hora de trabajar. En tal caso conocimientos adquiridos en la universidad relativos a dispositivos analógicos y digitales, señales, sensores y microprocesadores han sido de gran utilidad, ya sea para poner en práctica o para aprender comportamientos de nuevos dispositivos.

Por otro lado los conocimientos adquiridos en la universidad son y se han considerado una base para adquirir otros, o para no limitar la capacidad de aprender conocimientos considerados más complicados. Se considera de todas maneras que siempre se puede empezar a aprender desde las raíces de cualquier temática por complicada que sea o parezca.

2. Introducción y objetivo

2.1. Introducción

El gran avance tecnológico esta hoy en día permitiendo un mayor control de los procesos industriales, cada vez los microprocesadores y microcontroladores son más rápidos y los componentes electrónicos más precisos. En consecuencia se desarrollan productos cada vez más sofisticados como los teléfonos móviles que ya hoy en día están permitiendo la monitorización física de cualquier componente. La dirección tecnológica es clara, con menos dispositivos se accederá a más información. Son así estos sistemas de monitorización y control aquellos que permiten obtener datos e información del entorno en el que vivimos y mostrarlos adecuadamente para su análisis, o su respuesta automática a través de un sistema de control reactivo.

Una de las mayores virtudes de estos sistemas es la capacidad de dar información directa al instante al usuario, dejando la opción libre de actuar. Otra sin ninguna duda es la respuesta automática que puede dar un software informáticos que por sí solos realizan tareas más banales que por ejemplo evitarían un consumo innecesario de una luz encendida de día, una planta mal regada, la regulación de temperatura ambiente. Tanto en el ámbito personal como empresarial el objetivo de esta tecnología es ganar tiempo cometiendo menores errores, y en consecuencia una mayor gestión de los recursos económicos.

Desde la revolución industrial, la implantación de procesos de producción en serie ha permitido un gran crecimiento del mercado, ofreciendo una mayor variedad de productos a un precio asequible. Hoy en día los sistemas de producción automatizados permiten abastecer una mayor demanda y ofrecer una amplia variedad de productos más elaborados y complicados de producir. Se incluyen los componentes electrónicos que de esta forma están al alcance de todo el mundo, permitiendo la implementación de este mismo proyecto.

2.2. Objetivo del proyecto

El siguiente proyecto se basa en la creación y comercialización de un producto electrónico con el objetivo de dar un soporte informativo al usuario, de esta forma se obtendrán datos de una bicicleta eléctrica que informarán a este y le ayudara a tomar decisiones respecto al ahorro energético y la salud personal.

El proyecto ha sido encarado de una forma autodidacta con la intención de aprender conocimientos considerados de importancia tanto en el mundo de la electrónica como en el de la comercialización de un producto. Además, el proyecto trata temas tan actuales como el ahorro energético y la salud personal que son temas de actualidad.

La organización ha sido de gran importancia, estructurando el proyecto en un orden lógico, adquiriendo primero los conocimientos necesarios, diseñando y practicando después y finalmente implementando. Esta estructura contribuye a concentrarse en un tema durante ciertos periodos de tiempo lo que permite su práctica y mejora de eficiencia, realizando las tareas en un menor tiempo. Se han tenido en cuenta variantes de organización como la selección de componentes, que sean accesibles y de un coste asequible, estructurando la memoria desde el principio, comentando el código de programación, escribiendo recordatorios de funcionamiento en papel, guardando copias de contenido. Poco a poco se evitan así errores fatales, como la dependencia en un componente o la pérdida de información, que conlleva a la frustración, malestar a la hora de trabajar y sobretodo pérdidas de tiempo.

3. Especificaciones básicas

3.1. Especificaciones técnicas

El producto está destinado a un usuario que tiene la necesidad de poder visualizar de forma fácil y en todo momento información de la bicicleta eléctrica, interactuando con la interfaz del producto dato a dato o almacenándolos para su posterior visualización.

Para la inicialización del proyecto, se ha realizado una lluvia de ideas en las que se concluye que la finalidad de la información a visualizar es mayoritariamente relativa al ahorro energético del vehículo y a la salud del usuario.

Evidentemente el procesado de la información no es tan simple como parece, se ha llevado a cabo un procesado de esta información que empieza en la adquisición de una señal física del mundo real a través de los sensores, que la convertirán en una señal eléctrica. La señal eléctrica se tendrá que acondicionar mediante los componentes electrónicos que sean necesarios para obtener la señal de salida necesaria. Y finalmente el voltaje acondicionado se convertirá en una señal digital a través de un convertidor analógico digital (DAC) [15]. Se muestra el proceso en la **(Figura 1)**.

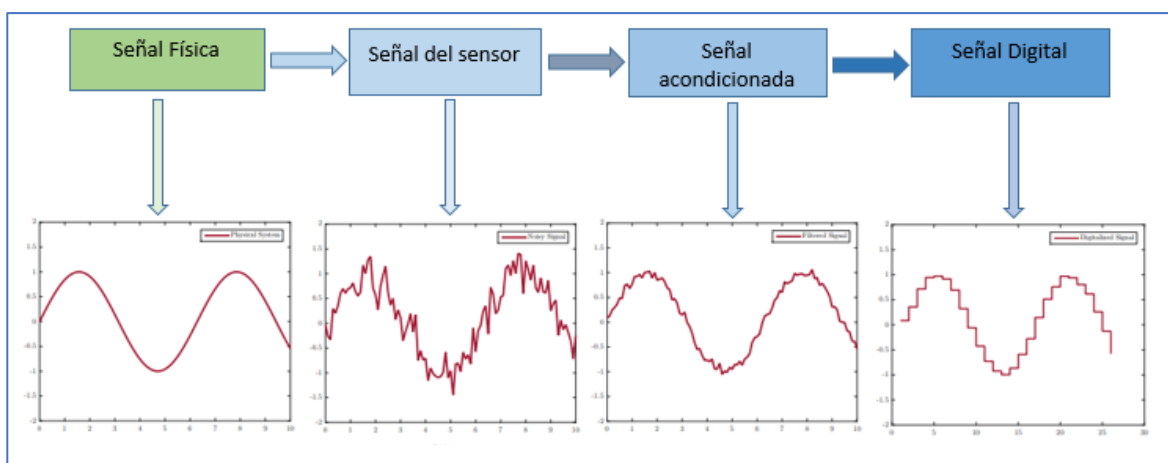


Figura 1. Muestra el proceso de conversión de la señal de física a digital.

Una vez se han obtenido los datos digitales estos son controlados por programas informáticos, *software* que da la posibilidad de interactuar con estos datos pudiendo dar una respuesta automática, almacenándolos y pudiendo ser visualizados.

3.2. Especificaciones conceptuales

Se ha mostrado una idea clara del contexto en el que se quiere crear el producto. La idea es idear su diseño y posterior implementación pensando que el producto en cuestión cumpla las especificaciones básicas y satisfaga las necesidades de autoaprendizaje del estudiante durante el proceso. Para eso se ha dividido el proyecto en cuatro partes que simbolizaran las temáticas que el alumno ha considerado de alta importancia durante el proceso:

- Conocimientos relacionados con la informática concretamente con la programación en C, se han adquirido gran variedad de conceptos de programación, herramientas matemáticas y todo lo que conlleva el manejo de programas como el CCS. También programación de la aplicación para móviles APP INVENTOR.
- Diseño electrónico: Se han adquirido conocimientos sobre el programa de diseño PROTEUS y el acondicionamiento de los componentes del circuito necesarios para su correcto funcionamiento.
- Implementación: Se han aprendido y consolidado conocimientos a la hora de implementar el diseño, especialmente con programas como el EAGLE y optimizando la organización de los componentes.
- Finalmente y no menos importantes se ha buscado la manera de comercializar el producto, la manera más eficiente y económica de producirlo en serie. Y para acabar la forma de vender el producto con campañas de *marketing*, utilizando redes sociales.

4. Posibles soluciones y solución escogida

La solución al problema se lleva a cabo de la forma descrita en la **(Figura 2)**. Esta resume los tres pasos que se llevan a cabo para determinar las posibles soluciones que plantea el proyecto, la obtención de los datos a través de sensores acondicionados, su procesado con un microcontrolador programado con un software informático y su final visualización a través de los posibles canales como un ordenador o un dispositivo móvil .

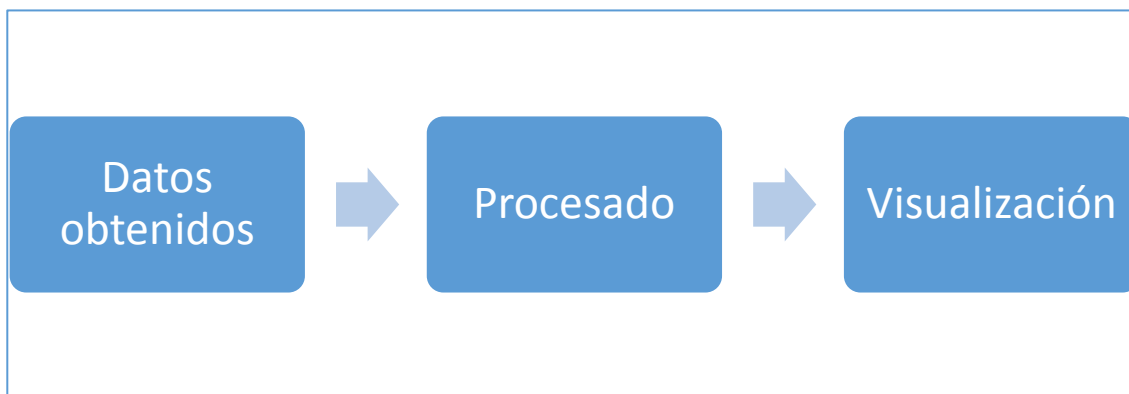


Figura 2. Estructura de los pasos seguidos para determinar las posibles soluciones.

4.1. Procesamiento de datos

En primer lugar se ha decidido la fuente a partir de la cual los datos serán tratados, es necesario comentar que hay varias maneras de procesar los datos de manera industrial, instrumentos tan clásicos como el PLC, o *data loggers*. Pero para el propósito de este proyecto no se buscan específicamente el control de datos conocidos como PID ni mucho menos el uso de instrumentos tan caros. Se ha buscado la versatilidad y libertad de creación que proporcionan los microcontroladores, estos circuitos integrados programables, permiten procesar y controlar a nivel de software los datos obtenidos.

Los microcontroladores son diseñados para aplicaciones de sistemas integrados, en contraste a los microcontroladores destinados a ordenadores personales. Comúnmente con las siguientes características:

- Centro de procesamiento de la señal: Con un rango tan simple como 4-bits a los más complejos 32-bits.
- ROM, EPROM, EEPROM o memoria *flash*: Para programar el almacenamiento de datos.
- Input y outputs: Bits que permiten el control o detección de estados lógicos en los pines.
- Comunicaciones serie: Área de control *network*, Bus *Serial Peripheral Interface* o el canal serie I2C.
- Sistemas periféricos como los *timers*, contadores, generadores PWM o *watchdogs*.
- Generador *clock*: Osciladores de cristal de cuarzo o circuitos RC.
- Convertidores analógico-digitales y convertidores digital-analógicos.

Respecto al tipo de microcontrolador, se ha elegido el modelo PIC por su coste económico y su libertad a la hora de diseñar comparado con otros dispositivos programables como el ARDUINO que parten de una base física más sólida con menos posibilidades de modificaciones. En la **(Figura 3)** se puede apreciar la comparativa.

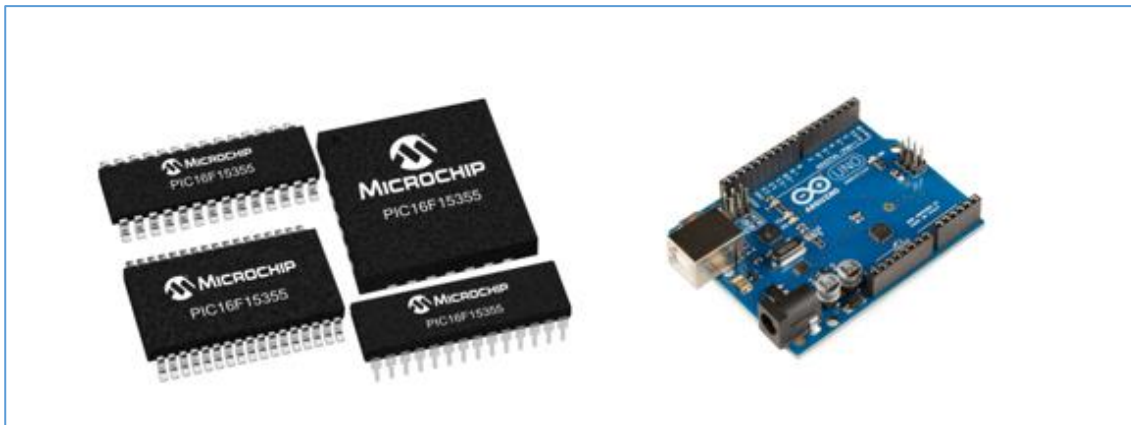


Figura 3: Microcontroladores MICROCHIP y muestra de ARDUINO (Fuente: Google).

4.2. Obtención de datos

Una de las partes importantes del proyecto es la esencia que se le da al producto, como se ha comentado anteriormente envuelve la temática del ahorro energético y a la salud del usuario. Para ello los datos obtenidos son referentes a estos temas comentados. Las funciones escogidas que dan utilidad al producto que son las siguientes:

- **Luminosidad del ambiente**

Se han obtenido los datos correspondientes a la luminosidad que hay en el ambiente constantemente, pudiendo controlar las luces del vehículo eléctrico de forma automática. De esta forma el consumo de luz queda controlado y no a la merced del error humano medida que permite optimizar el consumo. Se ha instalado de esta forma un sensor que permita evaluar la intensidad luminosa del ambiente y transmitir el valor en datos digitales para ser procesado.

- **Carga almacenada de la batería**

Respecto a la batería, se han obtenido los datos en tanto por ciento, informando en todo momento al usuario la batería restante y avisando en caso que esté en batería baja.

- **Temperatura**

Se mostraran también en todo momento datos relativos a la temperatura ambiente como dato informativo, instalando un sensor térmico en la bicicleta.

- **Distancia recorrida**

Se obtendrán datos del recorrido realizado permitiendo al usuario consultar la distancia recorrida en todo momento. Esta es una de las opciones del producto que informa sobre la salud del usuario ya que este tendrá la posibilidad de consultar los datos y marcarse objetivos. Se instala de esta forma un sensor magnético con un imán en la rueda de la bicicleta que permita a nivel de software llevar el conteo de vueltas y determinar la distancia por el vehículo.

- **Fecha y hora**

Se ha utilizado un dispositivo externo con su propia alimentación, la idea es experimentar con nuevos componentes y en este caso se ha programado de tal manera que informe al usuario de la hora, minutos y segundos. Además los valores de tiempo permitirán ser utilizadas como variables de resto de programación.

- **Almacenamiento**

Todo y disponer de un almacenamiento interno en el mismo PIC se ha instalado un almacenamiento externo EEPROM con la idea de poder guardar los datos obtenidos durante el recorrido, con la posibilidad de poderlos descargar posteriormente. Se experimenta así de igual forma con nuevos dispositivos electrónicos.

- **Teclado**

Para la selección de cada sensor se ha incorporado un teclado que permite la elección individual de los sensores además de modos personalizados como pueden ser los de almacenamiento de datos en la EEPROM externa.

4.3. Visualización

El último paso es la visualización, que se ha dividido en tres posibilidades que se muestran en la (Tabla 1). Se han valorado las posibles ventajas que puede tener cada una divididas en familiaridad, precio, consumo, calidad de visualización, dificultad de simulación y dificultad de implementación.

| Dispositivo | LCD | PC | Dispositivo Móvil |
|------------------------------|-------|-------|-------------------|
| Familiaridad | Alta | Alta | Baja |
| Precio | Bajo | Bajo | Bajo |
| Consumo | Bajo | Alto | Medio |
| Calidad de visualización | Media | Buena | Buena |
| Dificultad de simulación | Buena | Media | Media |
| Dificultad de implementación | Baja | Alta | Alta |
| Interés | Bajo | Bajo | Alto |
| Puntuación | 6 | 4 | 5 |

Tabla 1. Ventajas y desventajas de los posibles visualizadores.

Respecto a la visualización se ha tenido una idea clara y es la posibilidad de poder informar al usuario al momento, ventaja que aportan la visualización con LCD y con el dispositivo móvil. Además el LCD permite una increíble disponibilidad de simulación con el programa PROTEUS que ha facilitado la programación. Se ha escogido así el LCD y el móvil como visualizadores, siendo este último el que mayor interés recibe a la hora de aprender.

5. Diseño del equipo

En el siguiente apartado se nombrarán los componentes utilizados con los que se cumplen las necesidades del producto, su funcionamiento conjunto y la estructura que se ha llevado a cabo para el correcto funcionamiento de estos junto al diseño de la placa para su posible producción en serie.

Es necesario remarcar que el diseño previsto del prototipo es el definitivo para su producción en serie, no el prototipo implementado definitivo debido a por ejemplo la imposibilidad de adquirir algún componente, o la limitación de creación del alguno de los softwares informáticos como se comenta más adelante.

5.1. Diseño del Hardware

5.1.1. Microcontrolador

Concretamente se ha elegido el modelo PIC 18f4550 adjuntado en la **(Figura 4)** de la familia de los MICROCHIP. Desde un punto de vista inicial del proyecto se ha pensado evitar la falta de complementos o sistemas del microcontrolador, por eso se ha elegido un microcontrolador que tiene más puertos que los otros modelos PIC y posibilidades de utilizar el bus serie I2C, bus serie USART o los convertidores analógico-digitales.

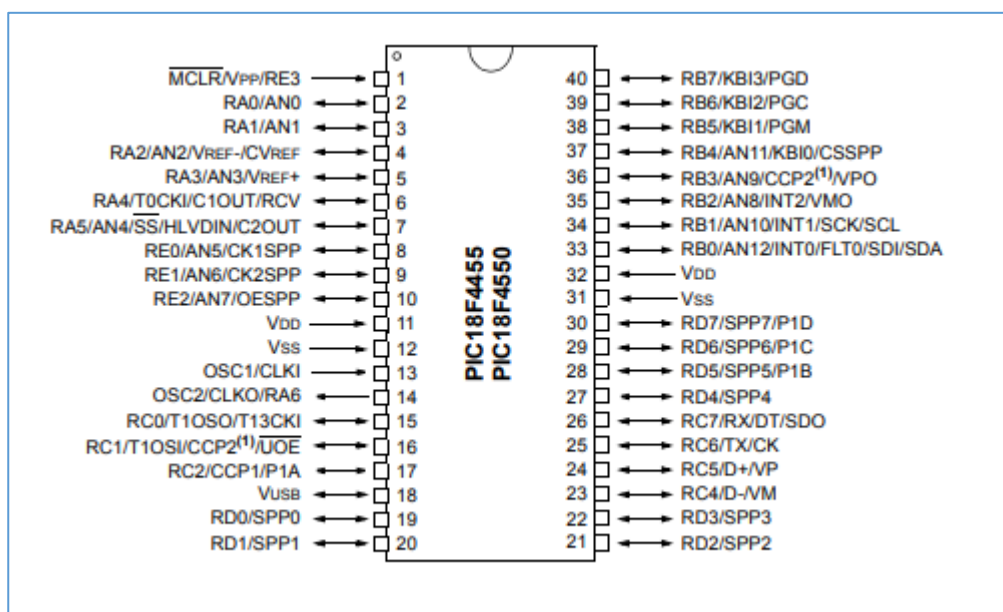


Figura 4. Microcontrolador PIC18f4550 (Fuente: 18f4550 data sheet [6])

5.1.1.1. Bus serie I2C

El bus serie I2C es un síncrono, maestro-esclavo y de transmisión serie diseñado para adjuntar múltiples dispositivos periféricos de baja frecuencia al microcontrolador. Es un bus bidireccional lo que significa que permite tanto enviar como recibir información a través de una línea de datos serie (SDA) y una línea de reloj serie (SCL) controlado por el maestro que van conectadas a una resistencia pull-up. El bus tiene una alimentación típica de 3.3 V- 5V, una velocidad de transmisión de entre 100KHz las más pequeña hasta 400KHz en datos de 7 bits. El esquema típico de conexión es el mostrado en la siguiente figura:

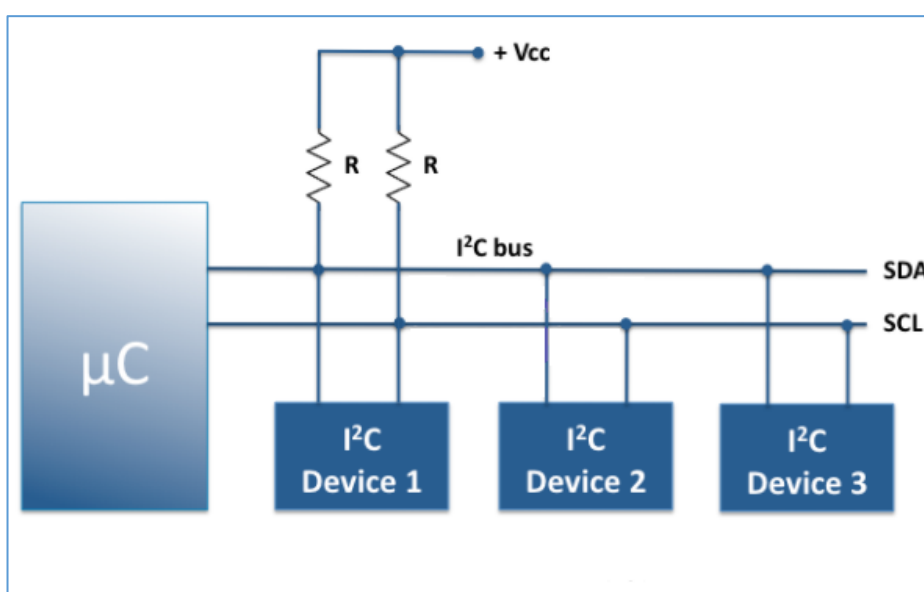


Figura 5: Esquema típico del bus I2C (Fuente: Google)

Con este sistema de transmisión de datos en serie se han incorporado otros componentes integrados como la EEPROM 24LC256 y el reloj a tiempo real DS1307. Se han tenido en cuenta que los dos puertos que ocupa el bus I2C corresponden también a dos pines del puerto B que principalmente ocupa un teclado, se han llevado a cabo pequeñas modificaciones del circuito pensado inicialmente que permiten la optimización de los puertos y del espacio del diseño general.

Es necesario comentar que con cada dispositivo el bus I2C tiene un protocolo de actuación que sigue el formato de la **(Figura 6)**. En este caso el PIC se ha programado con el programa CCS y el lenguaje de programación C y sigue los pasos descritos en la figura.

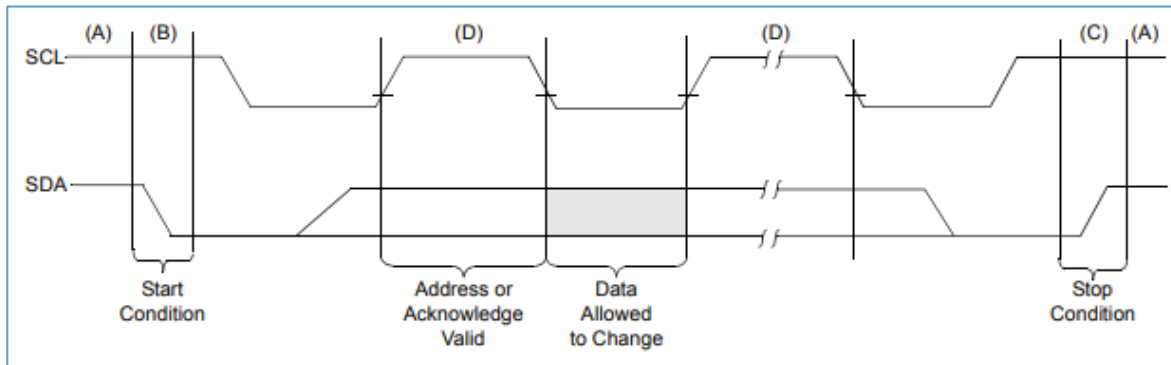


Figura 6: Formato de mensaje que utiliza el bus I2C (Fuente: 18f4550 data sheet [6]).

5.1.1.2. Comunicación USART

El USART es una comunicación en serie síncrona del microcontrolador que permite con un circuito integrado interno, transmitir o recibir información a través de dos canales Tx y Rx. El circuito (UART) contiene un *shift register* que es el método principal de convertir datos de paralelo a serie y al revés. El funcionamiento completo consiste en dos UART, una transmite la información secuencial de bits únicos y un segundo UART los empaqueta en bytes. Toda información transmitida contiene un bit de comienzo que indica la llegada de información y uno de finalización o paridad que es el bit de seguridad, confirma que se ha enviado correctamente la información.

Se ha instalado en este sistema de comunicación serie el componente HC-06, que ha permitido transmitir datos a un dispositivo móvil. Como en el caso del bus de transmisión en serie I2C, la comunicación a través del USART también se ha programado en C con sus correspondientes protocolos de programación. Por otro lado La transmisión y recepción de datos a través del móvil ha sido programada a través de la aplicación para Android App Inventor del MIT.

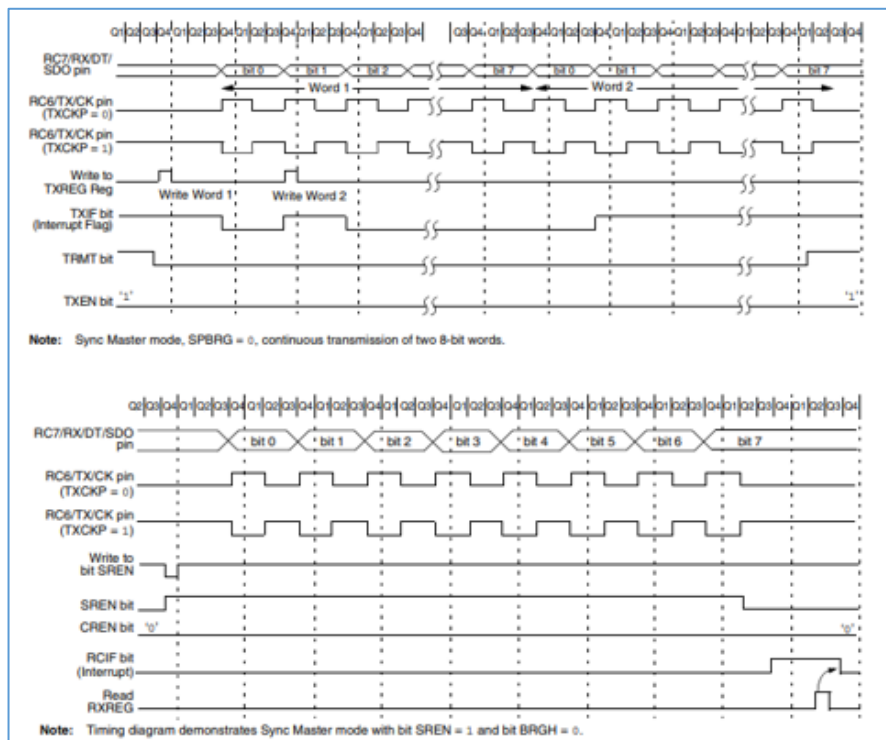


Figura 7: Protocolo de transmisión y recepción de datos (Fuente: 18f4550 fata sheet^[6]).

5.1.1.3. ADC

El convertidor analógico digital es un sistema que convierte una entrada de señal analógica del mundo real a una salida de señal digital. Esta también proporciona una medida que convierte un voltaje o corriente en un valor numérico que representa la magnitud de la señal permitiendo su posterior procesado a través del microcontrolador. Los convertidores se ubican en los puertos A y C del microcontrolador y han sido activados con su correspondiente programación en C. De la misma forma se han programado para obtener una resolución de 10 bits de señal de entrada lo que significa que la señal tendrá entre 0 y 1023 niveles. Siendo la resolución en voltaje del ADC V_{res} y N los niveles:

$$V_{res} = \frac{V}{N} = \frac{5}{1023} = 4,89mV \tag{Ec. 1}$$

El funcionamiento pasa por cuatro etapas:

- Muestreo: Consiste en tomar valores discretos del voltaje de entrada en una cierta velocidad de muestreo que es la frecuencia de muestreo.
- Retención: Es el tiempo que se toma para poder cuantificar las muestras obtenidas.
- Cuantificación: Se cuantifica a un valor un rango de voltaje.
- Codificación: Los valores obtenidos en la cuantificación se traducen al sistema binario.

En la siguiente figura se observa la conversión que realiza la señal al pasar de ser un voltaje analógico a un voltaje digital:

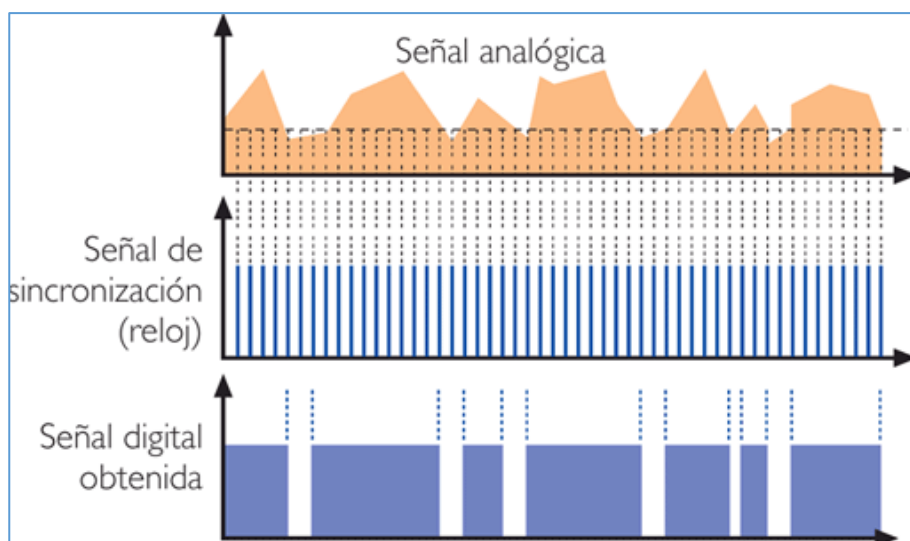


Figura 8: Conversión de una señal analógica a una digital (Fuente: Google)

5.1.1.4. Acondicionamiento

El microcontrolador se ha acondicionado sobre un soporte del mismo número de pines que está fijado a la placa del producto, el objetivo es que el PIC18f4550 pueda quitarse y ponerse para ser reprogramado de forma cómoda. A parte del soporte el microcontrolador se ha alimentado y puesto a tierra por los pines Vdd y Vss. El puerto Vpp del microcontrolador se ha acondicionado de manera que tiene la función de *reset* como se muestra en la **(Figura 9)**, al pulsar el pulsador se conecta el circuito a tierra.

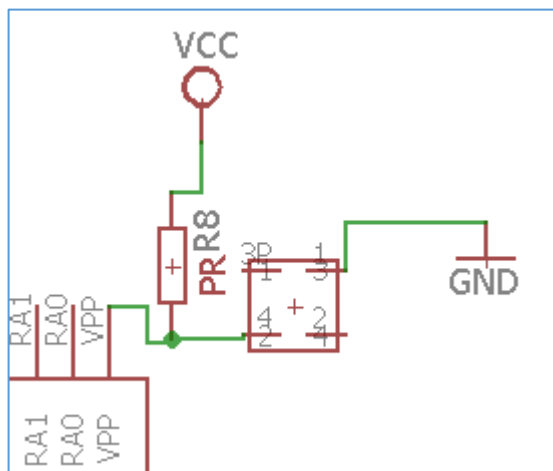


Figura 9: Función del puerto Vpp como reset del microcontrolador

5.1.2. Sensores y acondicionamiento de señales

5.1.2.1. Diseño del circuito de Iluminación

Para el control de la iluminación ha sido necesario la implementación de un sensor lumínico LDR (Resistor dependiente de la luz) concretamente el modelo GL7516. La resistencia interna del foto resistor decrece con la incidencia de la intensidad de la luz medida en lux como muestra el gráfico de la siguiente figura:

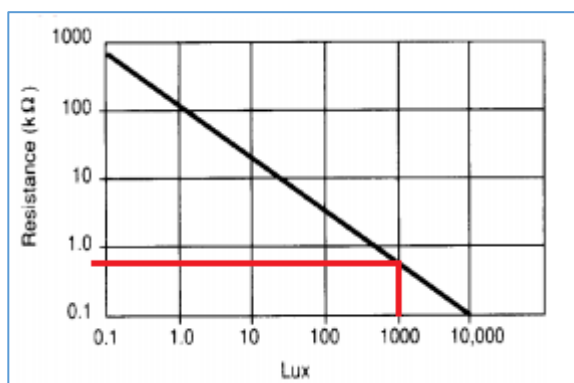


Figura 10: Grafico resistencia e iluminación del foto resistor (fuente: GL7516 data sheet [7]).

El circuito a implementar es fácil, se ha diseñado un divisor de tensión como muestra la **(Figura 11)** en el que hay una resistencia superior R1 de 10kΩ y el LDR actúa como resistencia inferior de la que se obtiene el voltaje de salida. Se ha limitado a 1000 lux la iluminación a partir de la cual se podrían tener problemas de visibilidad, que como se muestra en color rojo en la **(Figura 10)**, representan aproximadamente 800Ω en el grafico que relaciona la variación en la resistencia respecto la incidencia de la luz. Siendo el voltaje de alimentación de 5 voltios el cálculo se describe en la **(ec.3)**, donde se deduce que por debajo 370mV la luz de la bicicleta eléctrica debe permanecer encendida. Los cálculos se han traducido en bits en la misma ecuación donde con el resultado de la **(ec.1)** se ha llegado a la conclusión, que por debajo aproximadamente del valor 76 se activara el circuito que enciende la luz como muestra la **(ec.4)**. Obviamente este resultado se ha utilizado para poder realizar la parte de la programación en C para controlar el valor necesario del sensor.

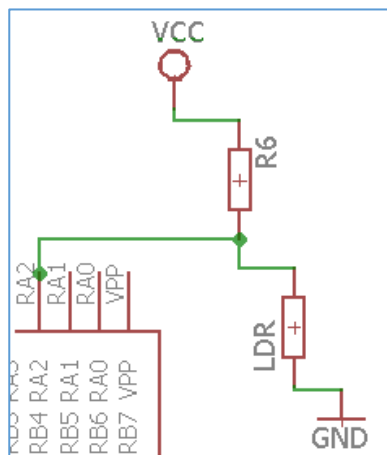


Figura 11: Circuito del LDR (Fuente: PROTEUS)

$$V_{out} = V_{in} \frac{R_{LDR}}{(R1 + R_{LDR})} \tag{Ec. 2}$$

$$V_{out} = 5V \frac{800\Omega}{(10k\Omega + 800\Omega)} = 370mV \tag{Ec. 3}$$

$$Valor = \frac{370mV}{4,89mV} = 76 \tag{Ec. 4}$$

Una vez controlada la intensidad lumínica mínima se ha diseñado un segundo circuito (**Figura 1**) que permita encender y apagar la luz. Este está controlado por un transistor que cuando el microcontrolador le dé la señal, permitirá que actúe el relé que cierra el circuito de iluminación alimentado con una dinamo. Se ha escogido una resistencia que asegure la intensidad permitida por el transistor modelo 2N2222 y un diodo 1N4002 en paralelo con la bobina y polarizado en inversa que asegura que el pico de tensión de apertura generado por la bobina no dañe el transistor.

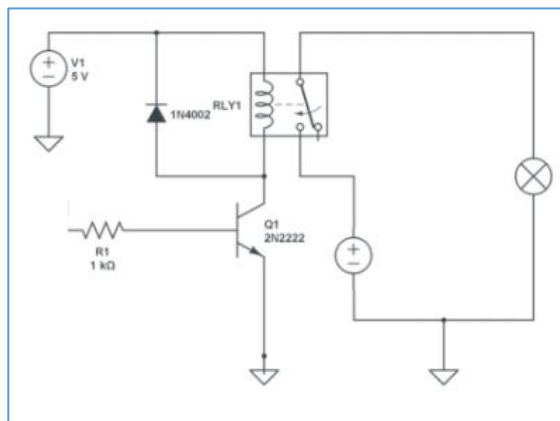


Figura 12: Circuito secundario acondicionado y circuito de iluminación(Fuente: Google)

Para el circuito de iluminación se ha escogido un interruptor modelo *power relay* SRD-3DC-SL-D. El modelo en cuestión es un encapsulado compuesto por el relé controlado por el transistor y un interruptor magnético que abre y cierra el circuito lumínico.



Figura 13: Encapsulado del interruptor SRD (Fuente: SRD data sheet ^[16]).

Finalmente para la iluminación se ha elegido un foco delantero que estará alimentado por una dinamo, esta es una medida de ahorro energético de la bicicleta. La dinamo apoyada en la rueda de la bicicleta permitirá generar electricidad para alimentar el foco.



Figura 14: Foco delantero, fijación y dinamo de la bicicleta.(Fuente: Amazon)

5.1.2.2. Diseño del circuito de control de la carga de la batería

Una de las características del producto es la posibilidad de consultar en todo momento el tanto por ciento de batería restante. Una batería típica de plomo está compuesta por diez elementos que proporcionan 1,4V cada uno cuando están totalmente cargados, y 0,9 cuando están descargados, hablando en máximos y mínimos la batería proporciona un rango de entre 9 y 14 voltios. Se ha tenido en cuenta que para leer estos datos con el microprocesador, el voltaje de entrada al puerto no debe ser superior a 5V, por lo tanto se ha diseñado un divisor de tensión que es alimentado con el voltaje de la batería como muestra la **(Figura 15)**. Para que la señal de la entrada al puerto no supere los 5V, la relación entre las resistencias tendrá que ser de 1,8 como muestra la **(ec.5)**. Debido a que el voltaje máximo de la batería es de 14V. Se ha tenido en cuenta también que para no influenciar el consumo las resistencias son del orden de megohmios.

Como muestra la **(ec.6)**. El rango de voltajes mínimo y máximo es entre 3,21V y 5V, margen aceptado para ser medido en el puerto RA que convertirá la señal analógica en señal digital. Posteriormente a través de programación software se ha preparado para que el resultado se visualice en porcentaje.

$$Relación = \frac{9V}{5V} = 1,8 \tag{Ec. 5}$$

$$V_{min} = 14V \frac{5M\Omega}{(9M\Omega + 5M\Omega)} = 5V$$

$$V_{max} = 9V \frac{5M\Omega}{(9M\Omega + 5M\Omega)} = 3,21V \tag{Ec. 6}$$

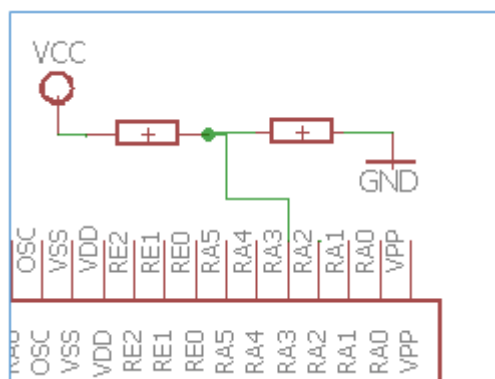


Figura 15: Circuito divisor de tensión

5.1.2.3. Diseño del circuito Temperatura

Como se ha comentado anteriormente se ha incorporado la opción de visualización de la temperatura ambiente, para ello el producto cuenta con un sensor térmico modelo LM35 que permitirá leer temperaturas entre -55°C y 150°C . El sensor se ha implementado como muestra la **(Figura 16)**, con dos diodos modelo 1N4002 y una resistencia de $18\text{k}\Omega$ que permite tener una única entrada de alimentación y el alcance máximo de valores de temperatura.

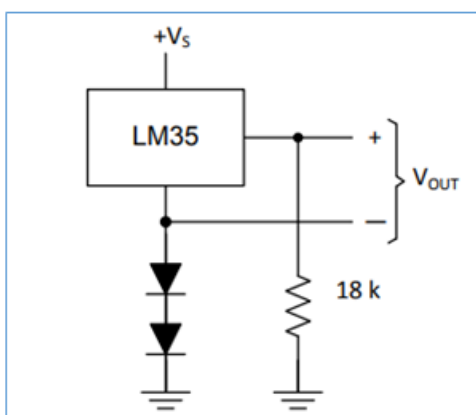


Figura 16: Circuito de acondicionamiento del LM35 (Fuente: LM35 data sheet ^[9]).

Con la anterior implementación del sensor se han obtenido dos señales de salida V_{out} que serán las que van conectadas directamente a los puertos RA0 y RA1 que permiten convertir la señal analógica en señal digital para posteriormente procesar los valores.

A nivel de cálculos, según la tabla obtenida del *data sheet*, el dispositivo proporciona 10 mV por grado centígrado que aumente la temperatura.

| PARAMETER | VALUE |
|-------------------------------|----------|
| Accuracy at 25°C | ±0.5°C |
| Accuracy from -55 °C to 150°C | ±1°C |
| Temperature Slope | 10 mV/°C |

Figura 17: Tabla con pendiente de temperatura (Fuente LM35 data sheet [9]).

$$Resolución = \frac{5V}{1024} = 4,8mV/bit \tag{Ec.7}$$

La forma de calcular la temperatura consiste en dividir la resolución entre la pendiente como muestra la (Ec.8) y multiplicarlo por el valor que se obtiene de la conversión analógica digital como muestran las siguientes ecuaciones:

$$Temperatura = Valor \frac{4,8mV/bit}{10mV/°C} = T°C \tag{Ec.8}$$

$$Temperatura = 100 \frac{4,8mV/bit}{10mV/°C} = 48 °C \tag{Ec.9}$$

$$Temperatura = 300 \frac{4,8mV/bit}{10mV/°C} = 144 °C \tag{Ec.10}$$

5.1.2.4. Diseño del circuito para determinar el recorrido

Otra función del producto es la posibilidad de poder visualizar la distancia recorrida por el usuario, para diseñar un sistema que pueda contar los kilómetros recorridos se ha instalado un sensor hall fijo en el chasis de la bicicleta y un imán fijo en la rueda, de manera que cada vuelta activará el sensor y a nivel de software se incrementa en una unidad la cuenta de ciclos siendo cada ciclo el perímetro de la rueda.

La salida de estos dispositivos tiene la característica de activarse estado *high* cuando un campo magnético externo excede el punto de operación BOP como se muestra en la **(Figura 18)**. De la misma forma el voltaje de salida se desactiva *low* cuando el campo magnético se reduce por debajo del punto BRP. En la región entre los puntos BRP y BOP se encuentra la histéresis o zona de *output state* en la que el sensor tiende a mantenerse activo. Se ha comprobado que en esa zona el tiempo de caída del estado a activo a desactivarse es inferior a 400ns y por lo tanto mientras el sensor se mantenga en histéresis es físicamente imposible que un ser humano alcance velocidades para realizar una segunda cuenta.

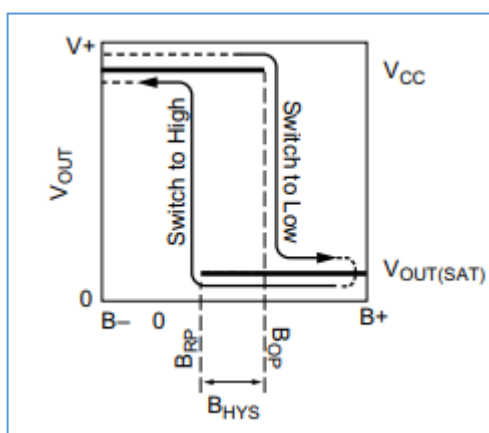


Figura 18: Ciclo de histéresis del sensor hall (Fuente: A1106EUA data sheet^[8]).

El modelo del sensor hall concretamente es el A1106EUA y se ha implementado como muestra la **(Figura 19)**, simplemente soldando una resistencia de 10kΩ entre la alimentación del sensor y la salida Vout. Esta misma salida es la encargada de conectarse en el puerto RC que recibirá la información de forma binaria a diferencia de los otros sensores estados high y low.

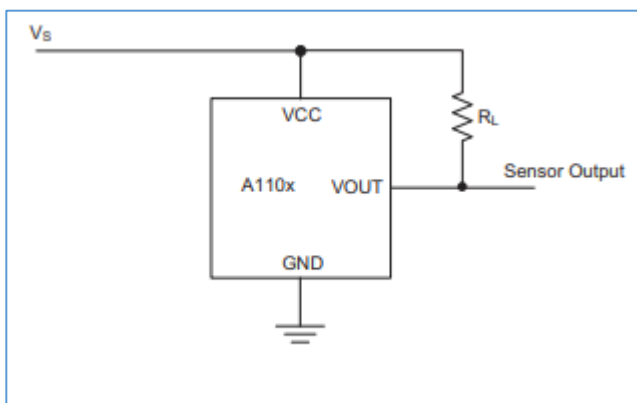


Figura 19: Implementación del sensor hall (Fuente: A1106EUA data sheet ^[8]).

5.1.2.5. Diseño del circuito del reloj de tiempo real

Como se ha descrito en las posibles soluciones adoptadas una de las características del producto es la posibilidad de visualizar la hora, para ello se ha usado un dispositivo externo, un reloj a tiempo real, concretamente el modelo DS1307 que se conectará directamente en el bus serie I2C a través de los pines SDA y SCL. El DS1307 es un circuito integrado de 8 pines que utiliza un protocolo de comunicación con el dispositivo maestro el PIC18F4550. Este pequeño chip puede contar segundos, minutos, horas, días, meses y años. Este recibe y transfiere datos en formato BCD así que después de recibirlos se han tenido que convertir a formato decimal y antes de escribir datos, convertirlos de formato digital al formato BCD.

En la siguiente (**Figura 20**), se puede apreciar el circuito acondicionado con las resistencias pull-up en el canal de comunicación serie I2C, la alimentación del chip de 5V y conexión a GND, un cristal de cuarzo externo de 32,768KHz frecuencia máxima a la que procesara los datos. Además el chip esta alimentado con una batería externa de 3V que le permite mantenerse alimentado cuando el circuito general no esté en funcionamiento.

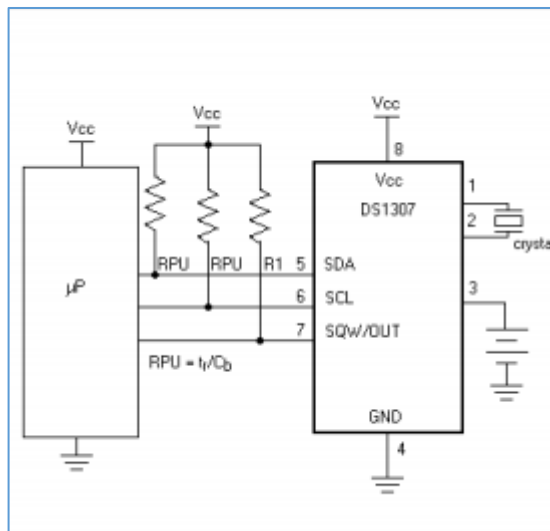


Figura 20: Típico circuito operativo del DS1307 (Fuente: DS1307 data sheet^[10]).

5.1.2.6. Diseño del circuito para una memoria externa

Para el almacenamiento de datos del producto se ha optado por el dispositivo 24LC25 que es una memoria EEPROM de 32kB de capacidad diseñada para aplicaciones de poca potencia y adquisición de datos. El acondicionamiento del chip es sencillo, en la **(Figura 21)** se muestra el empaquetado del 24LC256 el cual tiene dos pines de alimentación Vcc y GND, dos pines que se han conectado directamente al PIC18F4550 como la conexión en serie I2C, sujetos a dos resistencias pull-up como en el caso del reloj a tiempo real. Un pin WP al aire y finalmente los tres pines de selección A0, A1 y A2 que permiten seleccionar el dispositivo al que se le transmitirán o leerán datos. Estos tres pines o bits permitirán seleccionar la memoria en caso que se hayan implementado varias a la vez incrementado la memoria del dispositivo.

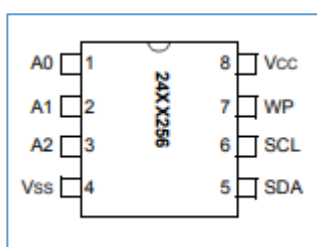


Figura 21: Empaquetado de la EEPROM 24LC256 (Fuente: 24LC256 data sheet^[12]).

En este caso se ha conectado el pin A0 a *high*, A1 y A2 a *low*. De la misma forma se ha llamado a nivel de programación como muestra el formato de control en la **(Figura 22)**. Los bytes se representan en formato hexadecimal por lo tanto el formato de escritura acabado en '0' en caso de escribir o '1' en caso de leer sería, "0010" y en hexadecimal "0xA2" siendo el número 2 la expresión de los cuatro bits anteriores y la letra A los cuatro bits de mayor importancia "1010".

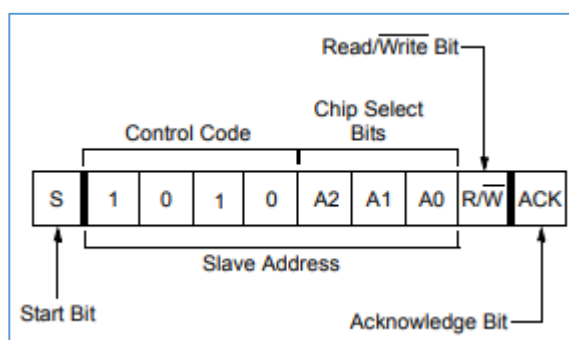


Figura 22: Formato del control del 24LC256 (Fuente: 24LC256 data sheet^[12]).

Como se ha mencionado anteriormente, la memoria externa EEPROM es de 13KB lo que significa que para la segunda parte de la programación hay memoria de almacenamiento limitado, para ello se han realizado los siguientes cálculos:

En primer lugar se han calculado los bytes que necesita un ciclo de parámetros para almacenar los datos de las cinco variables a consultar, se ha tenido en cuenta que el almacenaje se realiza de ocho bits en ocho bits, es decir en bytes enteros ya que al escribir en la memoria se selecciona el número de byte a grabar del dispositivo.

| Medida | Instrucción | Bytes |
|-------------|-------------|--------|
| Temperatura | Int8 | 1Byte |
| Distancia | Int16 | 2Bytes |
| Tiempo | Int8 | 1Byte |
| Batería | Int8 | 1Byte |
| Luz | Int8 | 1Byte |
| | | 6Bytes |

Tabla 2: Espacio que ocupa cada dato de las variables

La segunda parte del programa que realiza el ciclo de parámetros ocupa un espacio de 6Bytes cada ciclo como muestra los datos de la tabla anterior. Teniendo en cuenta los 13KBy de datos que puede almacenar, se procede a calcular el número de ciclos que puede realizar el programa antes de ocupar todo el espacio disponible de la memoria:

$$M = \frac{13000}{6} = 2166 \text{ ciclos} \quad (\text{Ec.11})$$

Finalmente, una vez se han obtenido el número de ciclos máximo que puede realizar el ciclo de parámetros, se ha establecido el periodo en el que guardara los datos en la memoria. Se ha decidido que el periodo de los ciclos dependa de la distancia recorrido, es decir, cada vez que recorre una distancia se guarden los datos de las variables en la memoria. De esta manera se ha elegido un límite de recorrido de 100Km para calcular aproximadamente cada cuanto distancia se guardan los datos.

$$Distancia = \frac{100.000}{2166} = 46 \text{ m} \quad (\text{Ec.12})$$

Cada 46 metros recorridos por el usuario el dispositivo realiza un ciclo de parámetros que almacena progresivamente en la memoria EEPROM.

5.1.2.7. Diseño del Teclado

Para el selector se ha implementado una serie de pulsadores que forman un teclado, debido a los continuos problemas de simulación y fallos en las librerías del simulador PROTEUS se ha optado definitivamente por un teclado matricial construido con nueve pulsadores simples que permiten la selección de nueve tareas distintas del producto. En la siguiente figura se presenta el teclado matricial a implantar:



Figura 23: Teclado matricial implementado (Fuente: google imágenes)

A pesar de no poder implementar a nivel de software los teclados normales que vienen de serie, se ha añadido un encapsulado al teclado matricial de forma que proporcione más facilidad al usuario a la hora de usarlo y sea más instintivo, además estará más protegido de agentes externos y dará un aire más presentable al producto.



Figura 24: Estructura física del teclado matricial. (Fuente: Google)

Para los pulsadores se ha elegido el modelo P6004 el cual tiene un funcionamiento sencillo, al ser presionado cierra el circuito correspondiente a las patas cruzadas del pulsador. Como se ha comentado anteriormente se han utilizado nueve pulsadores cuya implementación se puede representar como una matriz de 3x3 (**Figura 25**), con tres pines configurados como salida (RB2, RB3, RB4) y otros tres configurados como entrada (RB5, RB6, RB7). Los puertos de salida proporcionan un voltaje de salida que al ser pulsado el pulsador permite el paso de corriente que llega a los puertos de entrada. Al pasar los puertos de entrada de un estado bajo a un estado alto se puede identificar que pulsador está presionado según una de las nueve combinaciones posibles.

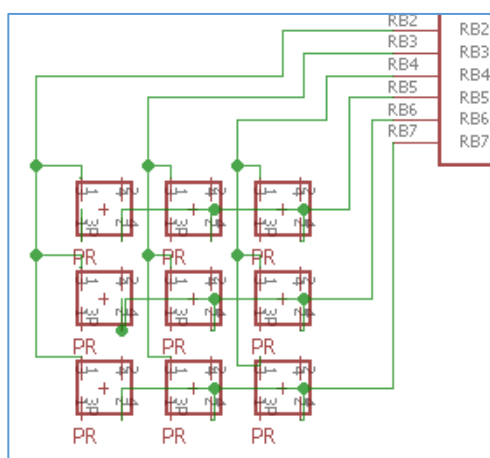


Figura 25: Estructura del teclado matricial 3x3 (Fuente: Eagle)

5.1.3. Visualización

5.1.3.1. LCD

El primer visualizador elegido por su fácil implementación y simulación es un LCD modelo PC1602A (**Figura 26**), con dos líneas de visualización horizontales y dieciséis verticales 16x2. Para su implementación simplemente se ha hecho el conexionado a los puertos (RD0, RD1, RD2, RD4, RD5) y RD7 de microcontrolador, que son los puertos aceptados por la librería disponible en el programa CCS. La librería permite una mayor comodidad a la hora de escribir el código ya que el programador se ahorra el protocolo que sigue el LCD cada vez que se quiere visualizar información. A parte también se conecta el visualizador a 5V de alimentación y al GND.

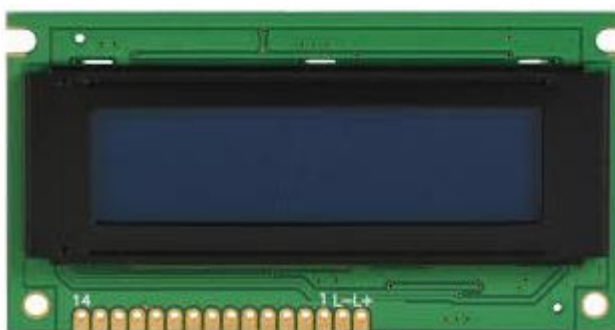


Figura 26: LCD modelo PC1602A (Fuente: Google).

5.1.3.2. HC-06

La segunda opción de visualización es a través del dispositivo HC-06 (**Figura 27**) de cuatro pines, un módulo *Bluetooth* SPP (Puerto de Protocolo Serie) diseñado para la conexión en serie inalámbrica. El *bluetooth* es la tecnología típica de intercambio de datos en distancias cortas y de forma inalámbrica entre dos dispositivos.

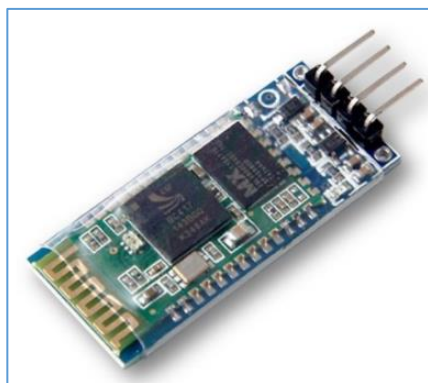


Figura 27: Dispositivo HC-06 (Fuente: google imágenes)

El modulo permite transmitir la información a 3Mbps con una ancho de banda de 2.4GHz. Como se ha comentado anteriormente el dispositivo se comunica con el microcontrolador a través de la comunicación en serie UART, puerto Tx y Rx que se han conectado de la forma que el conector de transmisión Tx va unido con el conector receptor Rx como muestra en la siguiente figura:

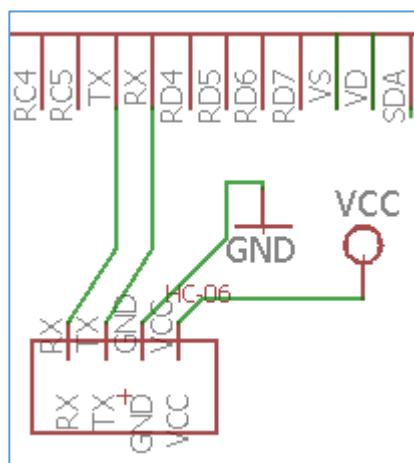
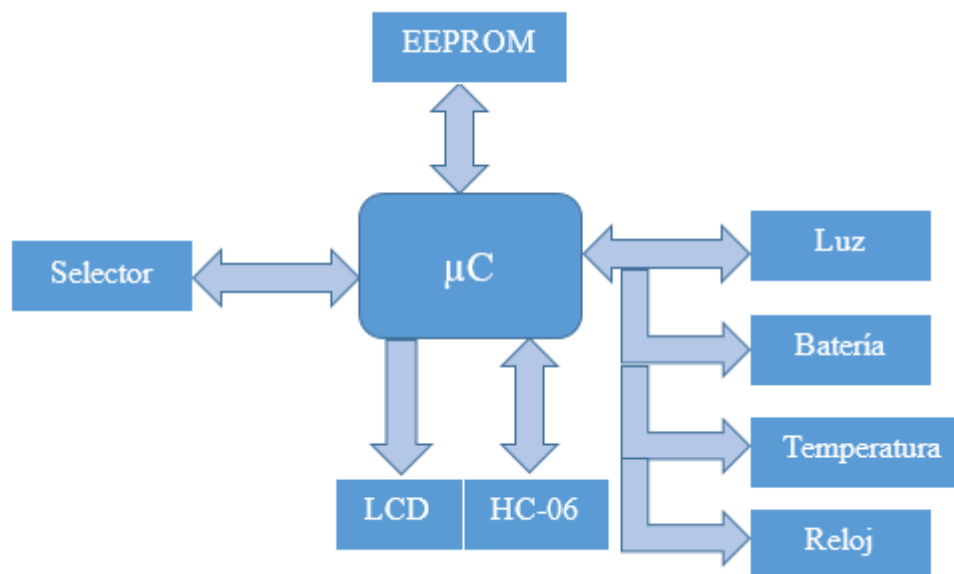


Figura 28: Conexión HC-06 con el microcontrolador (Fuente: Eagle)

5.2. Diagrama de bloques general

Seguidamente se presenta el diagrama de bloques según el funcionamiento del circuito, que básicamente a partir de microcontrolador se leen los datos de los sensores y se da una respuesta en consecuencia de los datos obtenidos ya sea emitiendo luz o en forma de visualización en el LCD o en el dispositivo móvil.



5.3. Diseño del prototipo

El diseño a producir se ha dibujado mediante el programa de diseño EAGLE, en la (Figura 29: Diseño del circuito con EAGLE (Fuente: EAGLE) se presenta el esquema a producir. Este se ha pensado y dibujado numerables veces en hoja y a mano de manera que se pueden visualizar posibles errores y comparar a escala si los distintos dimensionados de los componentes encajan unos con otros sin estorbarse. Principalmente se ha tenido en cuenta que los sensores de temperatura LM35 no sea influenciado térmicamente por el microcontrolador y que el sensor lumínico esté debidamente colocado para su exposición a la luz natural.

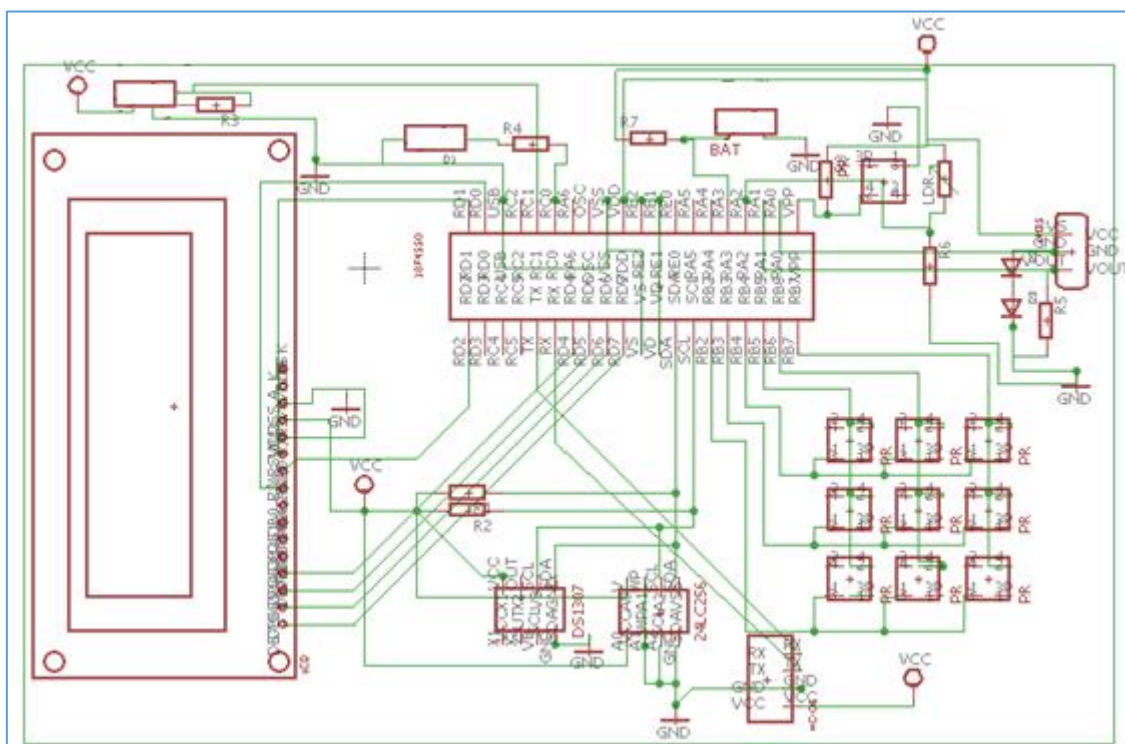


Figura 29: Diseño del circuito con EAGLE (Fuente: EAGLE)

5.4. Conexionado en el vehículo

En este apartado se comentan todas las conexiones físicas que realiza el equipo con el vehículo, empezando por el conexionado de la placa principal que tiene incorporado el LCD junto al otro método de visualización que sería el dispositivo móvil.

El dispositivo, que tiene unas dimensiones parecidas al móvil y por lo tanto de fácil incorporación en la bicicleta ya que es de poco peso y no influencia de ninguna manera la estabilidad de este, estará situado junto al manillar ya que es de fácil acceso para el usuario tanto manualmente (acceso al menú de selección) como visualmente, puede ir junto al posible dispositivo móvil incorporado también en la bicicleta. Ambos sujetos de la forma adecuada y protegidos para evitar posibles caídas o daños materiales. Además la situación permite una fácil colocación del resto de componentes que obtienen datos del vehículo.



Figura 30: Lugar de ubicación de la placa base del producto (Fuente: Render).

Por otro lado las tres cajas vacías del diseño (**Figura 29**) corresponden a los sensores o circuitos que se han colocado externamente:

- Dispositivo de iluminación:

El circuito de iluminación ya descrito en el diseño del prototipo, es un circuito externo compuesto por el foco delantero situado en la parte frontal de la bicicleta con el que se iluminará el camino en caso que el sensor lumínico no detecta suficiente luz ambiental, también se instalará la dinamo que estará fijado al chasis que sujeta la rueda de la bicicleta y estará en constante fricción con esta para alimentar el foco principal.

Se ha tenido en cuenta también la cantidad de cableado que puede poner en riesgo al usuario, por eso todo cableado estará correctamente situado siguiendo el chasis de la bicicleta bien atado para prevenir posibles molestias o incluso accidentes.

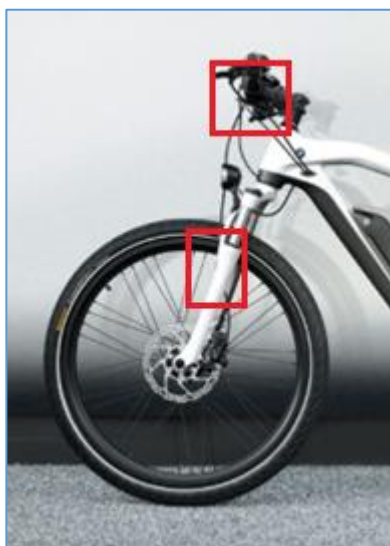


Figura 31: Situación del sistema de iluminación (Fuente: Render).

- Sensor Hall:

La instalación del sensor se ha compuesto en dos partes, en primer lugar el sensor hall que irá fijado al chasis de la rueda con el sentido de imantación hacia esta. El sensor estará protegido por un encapsulado que le permitirá no estar expuesto a los elementos externos como se muestra en la **(Figura 32)**.

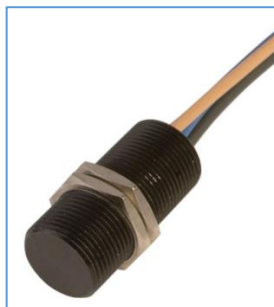


Figura 32: Sensor hall en su encapsulado (Fuente: Google).

En segundo lugar se ha instalado el imán, que ira fijado a la rueda de la bicicleta de forma fija y segura. El imán estará encarado en todo momento al sensor hall para que cada vuelta este detecte magnetismo.



Figura 33: Situación del imán (Fuente: Google).

- Conexión del control de carga de la batería:

Finalmente se ha instalado también el cableado necesario para obtener los datos de voltaje necesarios para medir el tanto por ciento de batería de la bicicleta eléctrica. Como en los casos anteriores el cableado se instala de manera que quede completamente fijo al chasis de la bicicleta sin molestar el libre movimiento del usuario.



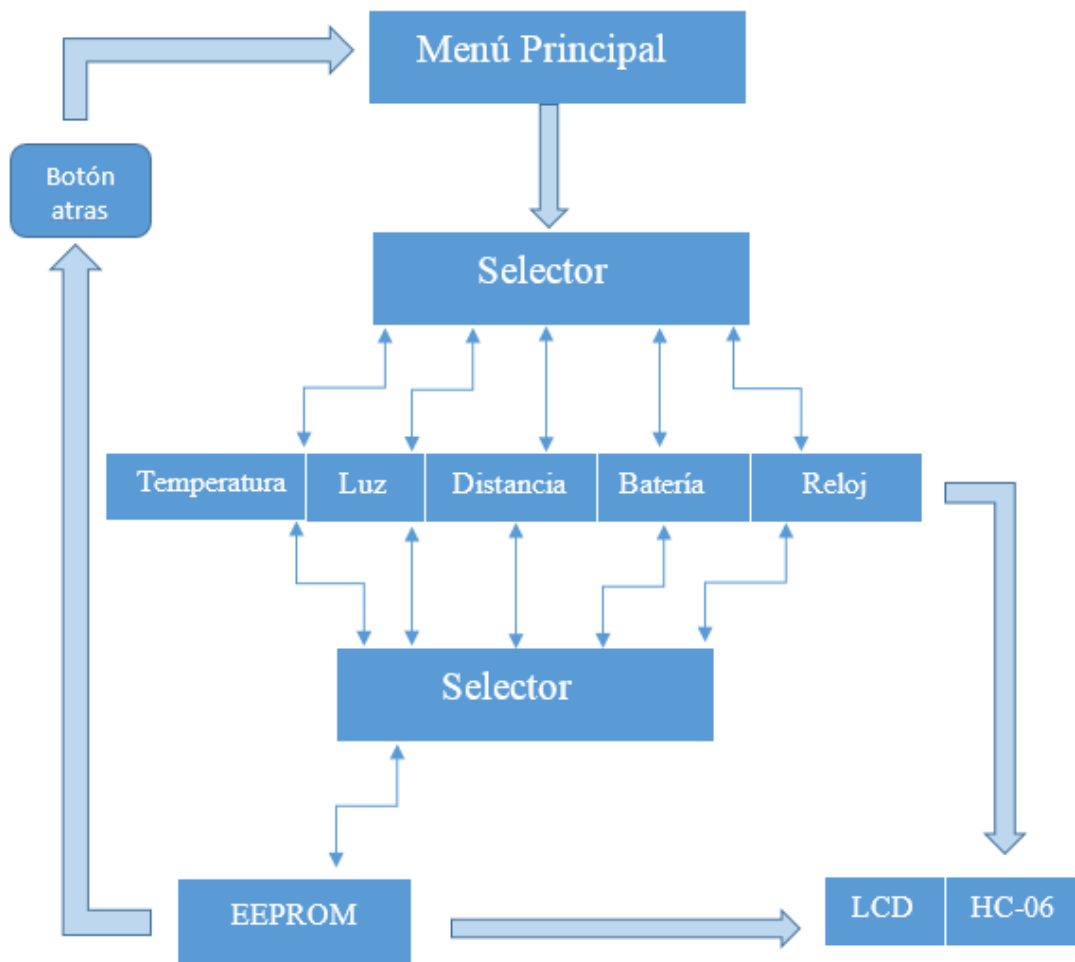
Figura 34: Conexión con la batería eléctrica (Fuente: Render).

6. Diseño del SOFTWARE

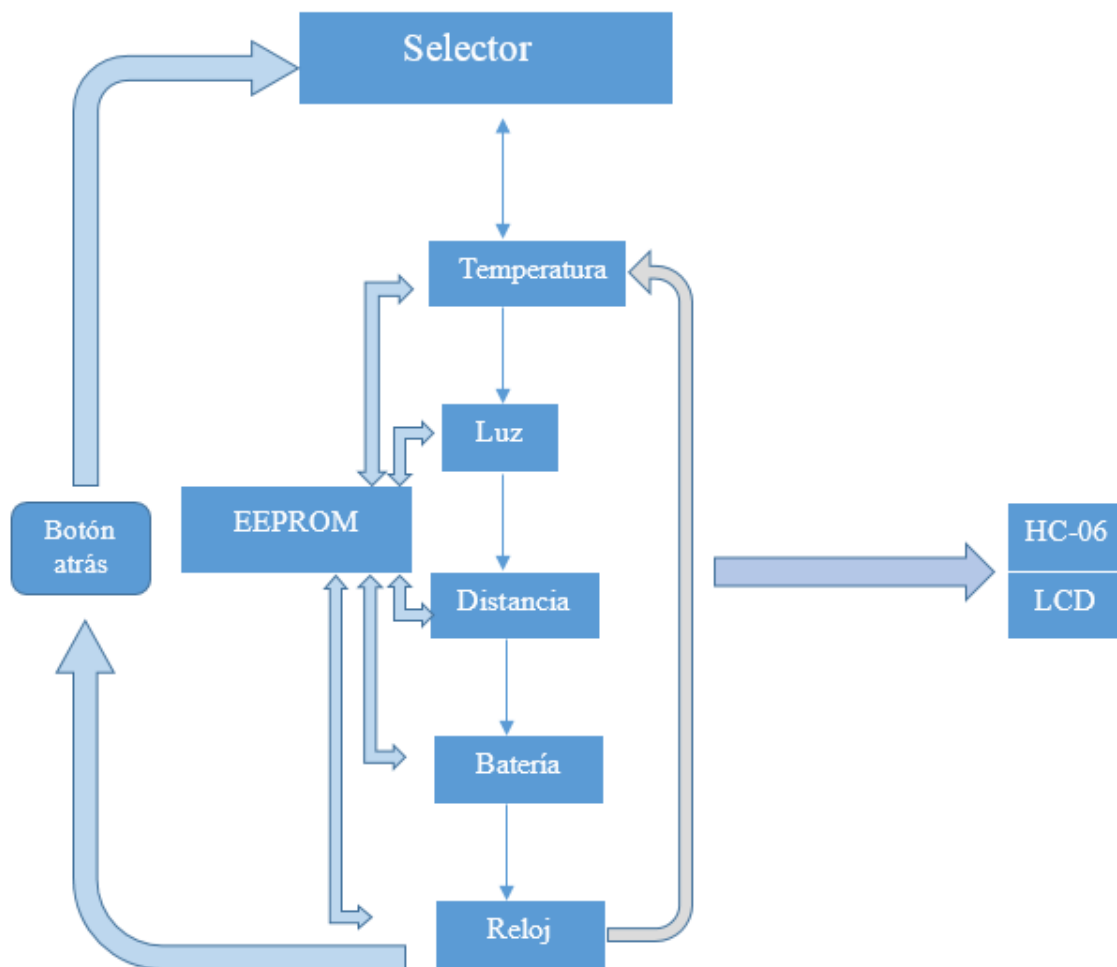
6.1. Diagrama de flujo

El diseño del software se ha dividido principalmente en dos partes:

En la primera parte se seleccionaran los distintos datos a visualizar a través de un selector y una vez dentro del bucle de visualización se podrán almacenar las variables como se muestra en el siguiente diagrama de flujo:



En segundo lugar, otra posibilidad que da el selector es el de poder realizar un registro de ciclo de parámetros en el que el programa visualizará los datos de los sensores uno a uno y al mismo tiempo almacenará el contenido de las variables como se muestra en el siguiente diagrama de flujo:



6.2. Programación App Inventor

A parte del software del microcontrolador, también se ha realizado un programa para enviar datos al dispositivo móvil que funciona de la siguiente forma:

En primer lugar se declara la variable a recibir, seguidamente se detecta la conectividad *bluetooth* y para finalizar se copian los datos a la variable que pasa a ser visualizada a la interfaz del usuario a través de un dispositivo móvil.

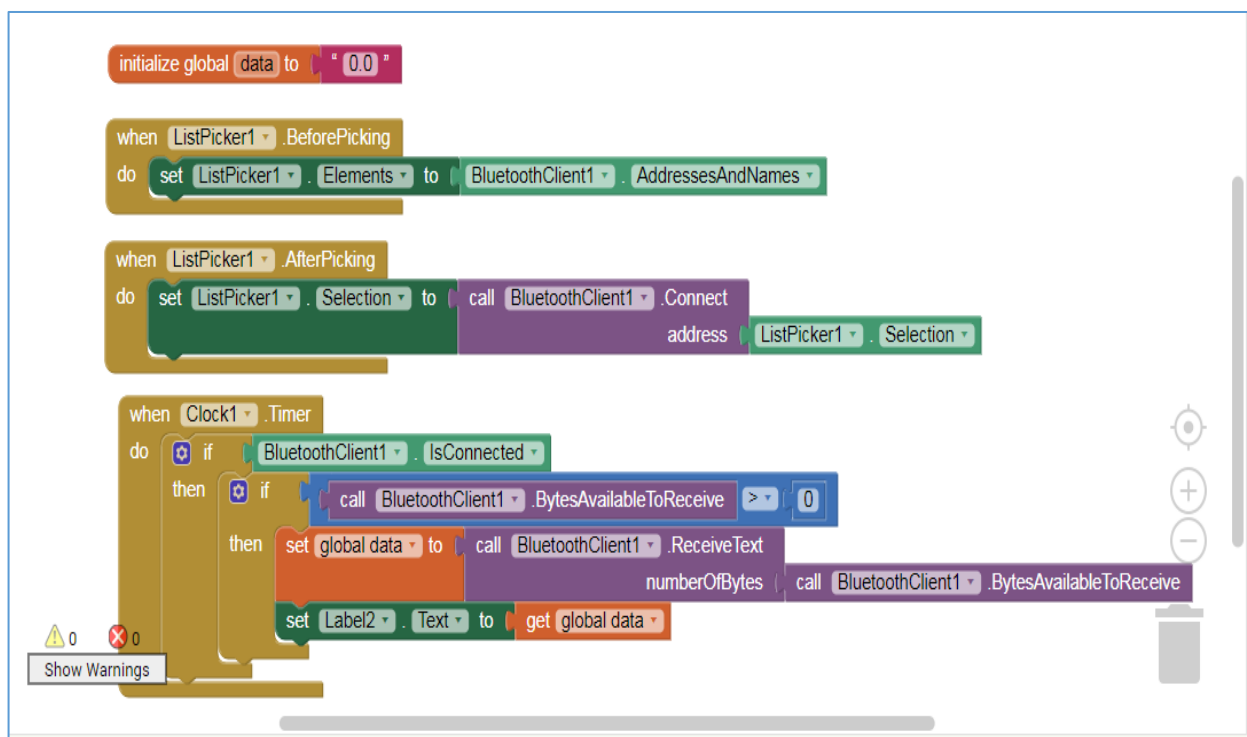


Figura 35: Programación de app inventor para el dispositivo móvil (Fuente: app inventor).

7. Diseño del prototipo de pruebas

EL diseño del prototipo de pruebas se ha realizado con el programa informático de simulación PROTEUS, este software permite diseñar toda clase de circuitos electrónicos de hardware, incluyendo microcontroladores, sensores, y todo tipo de dispositivos que están disponibles en las librerías incorporadas o librerías que se pueden descargar. También permite la posibilidad de diseñar el software que controla los microcontroladores en lenguaje C a través del programa de compilación CCS, que mediante una serie de archivos permite cargar la programación del microcontrolador a PROTEUS.

De esta forma el estudiante ha adquirido todos los conocimientos relacionados con el programa de simulación PROTEUS SUITE DESIGN y del compilador CCS, por lo tanto también el lenguaje de programación en C.

La fuente de información principal para el aprendizaje ha sido el manual de compilación y simulación de PROTEUS ^[1], el cual presenta de forma ordenada, por pasos y con ejemplos las distintas características de un microcontrolador y como son programadas y simuladas.

7.1. Diseño del Hardware

7.1.1. Microcontrolador

Para el diseño de prueba del microcontrolador se ha utilizado el que proporciona las librerías de PROTEUS, el mismo PIC 18f4550 con todas las características necesarias para realizar las simulaciones, bus serie I2C, bus serie USART, los convertidores analógico-digitales, todos los puertos:

- Puerto A: Ha permitido la incorporación de sensores, se ha utilizado básicamente de convertidor analógico digital.
- Puerto B: Se ha instalado el teclado matricial, además es la localización de bus serie I2C en los puertos RB0 y RB1.
- Puerto C: Se ha utilizado de convertidor analógico digital o de circuitos externos como la conexión a la batería o el circuito de iluminación, también se ubica el bus serie USART con los puertos Rx y Tx para el funcionamiento del dispositivo HC-06.
- Puerto D: Es el puerto donde se ubican las librerías destinadas al control del LCD.

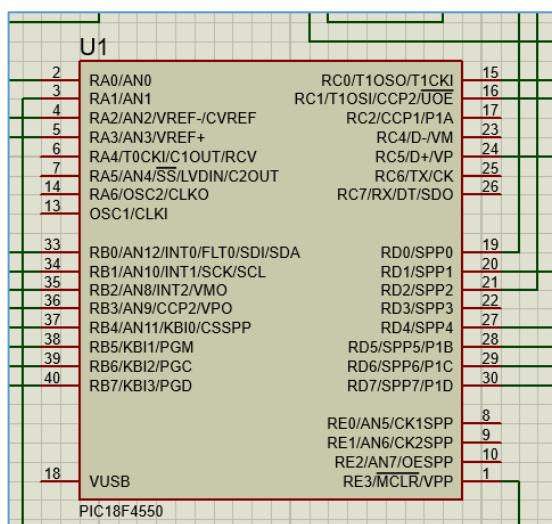


Figura 36: Diseño de prueba PIC18f4550 (Fuente: PROTEUS).

7.1.1.1. Acondicionamiento

Se ha de tener en cuenta que en el microcontrolador no aparecen todos los pines de alimentación lo que puede llevar a confusiones durante la implementación de este. Además los pines no corresponden al orden asignado real del data *sheet* del microcontrolador cosa que también se debe tener en cuenta para después ser implementado.

Para el acondicionamiento se ha realizado el circuito descrito en el diseño del prototipo con una resistencia de 10k Ω y el pulsador que genera el *reset* al ser pulsado.

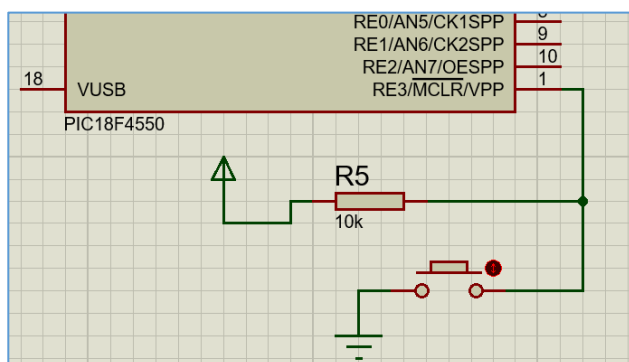


Figura 37: Circuito de Reset del 18f4550 (Fuente: PROTEUS).

7.1.2. Sensores y acondicionamiento

7.1.2.1. Diseño del circuito de prueba de iluminación

Para la simulación del sensor de luz LDR se ha utilizado este mismo sensor disponible en las librerías de PROTEUS, además el software permite incorporar una luz externa que simula la radiación lumínica como muestra la , la interacción con ella permite acercarla o alejarla permitiendo poner a prueba un rango de detección lumínica.



Figura 38: LDR disponible en las librerías PROTEUS (Fuente: PROTEUS)

Para el diseño de pruebas, se ha diseñado un divisor de tensión como muestra la (Figura 11) en el que hay una resistencia superior R1 de 10k Ω y el LDR actúa como resistencia superior de la que se obtiene el voltaje de salida. Se ha limitado a 1000 lux la iluminación a partir de la cual se podrían tener problemas de visibilidad, que como se muestra en color rojo en la (Figura 10), representan aproximadamente 800 Ω en el gráfico que relaciona la variación de la resistencia respecto la incidencia de la luz. Siendo el voltaje de alimentación de 5 voltios el cálculo se describe (ec.14), donde se deduce que por encima de 4,62V la luz de la bicicleta eléctrica debe permanecer encendida. Los cálculos se han traducido en bits en la misma ecuación donde con el resultado de la (ec.1), se ha llegado a la conclusión, que por encima aproximadamente del valor 947 se activara el circuito que enciende la luz como muestra la (ec.15) que divide el voltaje entre la resolución. Obviamente este resultado se ha utilizado para poder realizar la parte de la programación en C para controlar el valor necesario del sensor y a diferencia de la descripción del apartado del diseño del prototipo, se han simulado el sensor y la resistencia al revés.

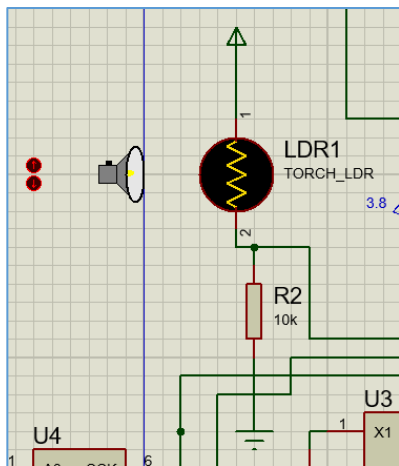


Figura 39: Circuito del LDR

$$V_{out} = V_{in} \frac{R1}{(R1 + LDR)} \tag{Ec. 13}$$

$$V_{out} = 5V \frac{10K\Omega}{(10k\Omega + 800\Omega)} = 4,629V \tag{Ec. 14}$$

$$Valor = \frac{4,629V}{4,89mV} = 947 \tag{Ec. 15}$$

Para el diseño del circuito secundario se han utilizado componentes que simulan las características reales de los componentes elegidos en el apartado de diseño del prototipo real. De la misma forma se ha utilizado una resistencia que asegura la intensidad permitida del transistor, un transistor NPN y una bobina junto a un interruptor. Debido a problemas de compatibilidad de software no se ha podido añadir el diodo que asegura que no haya sobretensiones en la bobina, de todas maneras no es un dato importante a nivel de software. En el circuito de iluminación se ha añadido una batería y una luz que simula el foco delantero de la bicicleta.

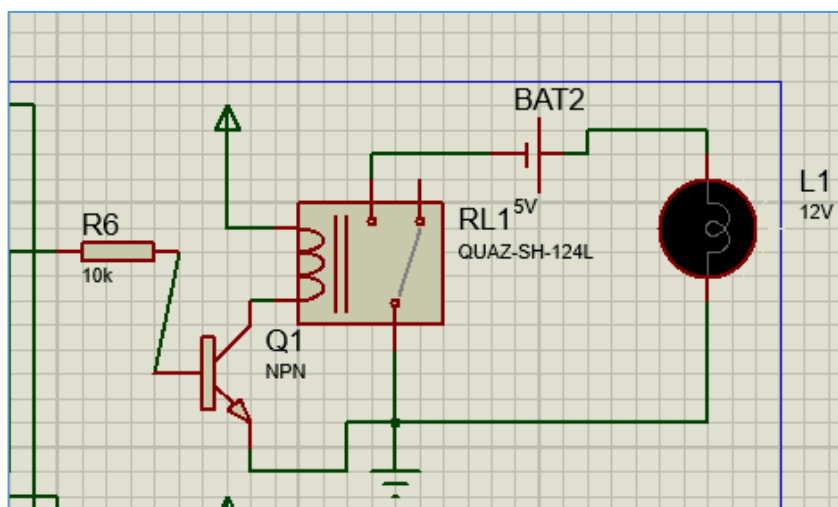


Figura 40: Circuito secundario acondicionado y circuito de iluminación

7.1.2.2. Diseño del circuito de prueba para el control de la carga de la batería

El diseño de la batería se ha realizado teniendo en cuenta que la entrada analógica del microcontrolador debe ser de unos 5V, de manera que el cálculo del tanto por ciento de la batería es proporcional al rango de valores de 0 a 5V. A nivel digital los valores del convertidor se moverán de 0 a 1024 bits. Se da un ejemplo en las siguientes ecuaciones:

$$V\% = \frac{0V}{5V} 100 = 0\% \quad \text{valor} = 0 \text{ bits} \quad \text{(Ec. 16)}$$

$$V\% = \frac{2,5V}{5V} 100 = 50\% \quad \text{valor} = 512 \text{ bits} \quad \text{(Ec. 17)}$$

$$V\% = \frac{5V}{5V} 100 = 100\% \quad \text{valor} = 1024 \text{ bits} \quad \text{(Ec. 18)}$$

Por otro lado el diseño en PROTEUS se ha realizado utilizando un *switch* que permita escoger distintos voltajes simulando distintos mementos de descarga de la batería como se muestra en la siguiente imagen:

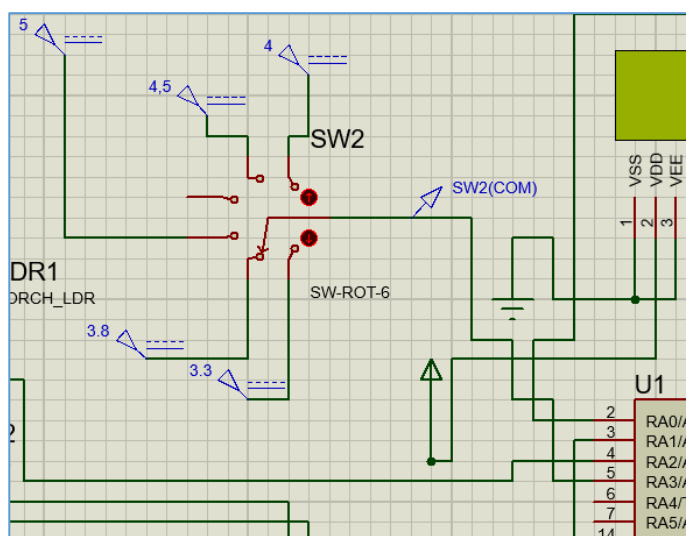


Figura 41: Circuito divisor de tensión con un potenciómetro

7.1.2.3. Diseño del circuito de prueba para el LM35

El diseño del sensor de temperatura en PROTEUS se ha realizado mediante el mismo diseño descrito en el diseño del equipo, dos diodos en serie conectados en la masa del sensor de las mismas características que los 1N4002, 1N914 en este caso, una resistencia de 18kΩ y el mismo sensor LM35 obtenido de las librerías del programa el cual tiene mucha versatilidad ya que es posible escoger una temperatura a simular en el margen de temperaturas del sensor -55°C y 150°C.

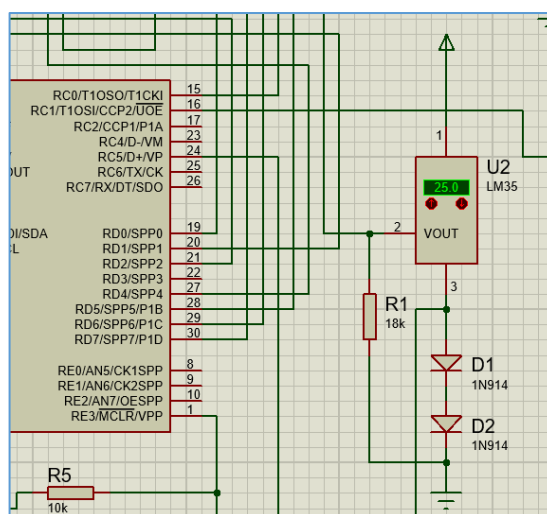


Figura 42: Circuito de acondicionamiento del LM35 (Fuente: PROTEUS)

Los cálculos del diseño de prueba del sensor LM35 son exactamente como los descritos en el diseño del equipo tanto a nivel de temperaturas positivas como de temperaturas negativas, la única diferencia es la entrada del puerto RA0 y RA1. Posteriormente el signo negativo de la temperatura se asigna a través de programación.

7.1.2.4. Diseño del circuito de prueba del sensor Hall

Para el diseño de prueba del sensor hall, se ha realizado de forma sencilla debido a que no se ha encontrado ninguna librería que corresponda al sensor escogido ni tampoco ningún sensor hall con el que se pueda interactuar como el LDR o el LM35.

De esta forma se ha elegido un *switch* con el que se pueda escoger una serie de frecuencias cuadradas de periódicas con la intención que simulen la detección de un imán por el sensor hall. Las distintas opciones permiten escoger diferentes frecuencias y poder visualizar el conteo de una forma más rápida o más lenta.

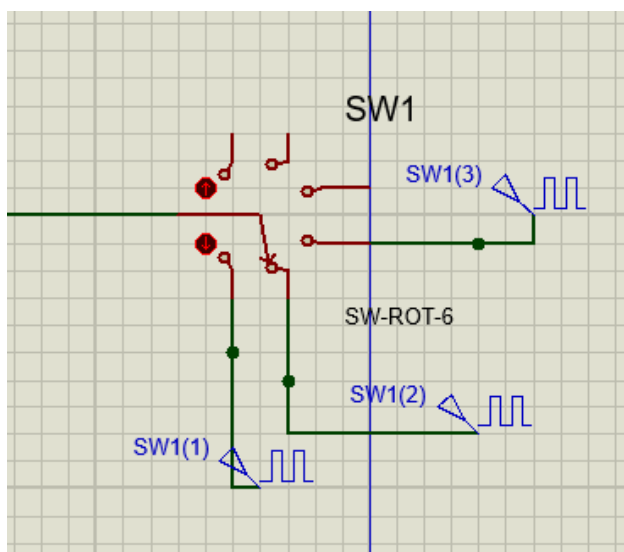


Figura 43: Implementación del sensor hall (Fuente: A1106EUA data sheet)

La frecuencia de una onda cuadrada es una magnitud que mide el número de repeticiones por unidad de tiempo y se mide en hercios (Hz). De manera que cuando más ciclos realicen en menos tiempo mayor es la frecuencia. Cada ciclo representa una vuelta de la rueda de la bicicleta. Por lo tanto cuando más alta es la frecuencia más ciclos se completan y más distancia simulada recorre el vehículo.

7.1.2.5. Diseño del circuito de prueba del DS1307

El diseño del dispositivo DS1307, reloj a tiempo real, se ha realizado mediante el uso de la librería en PROTEUS que permite la utilización del componente como se aprecia en la (Figura 44), se ha conectado a un cristal de cuarzo externo de 32,768KHz frecuencia máxima a la que procesara los datos, una batería externa de 3V que le permite mantener los datos de tiempo a tiempo real y el conexionado adecuado del bus serie I2C con dos resistencias pull-up de 4,7K Ω .

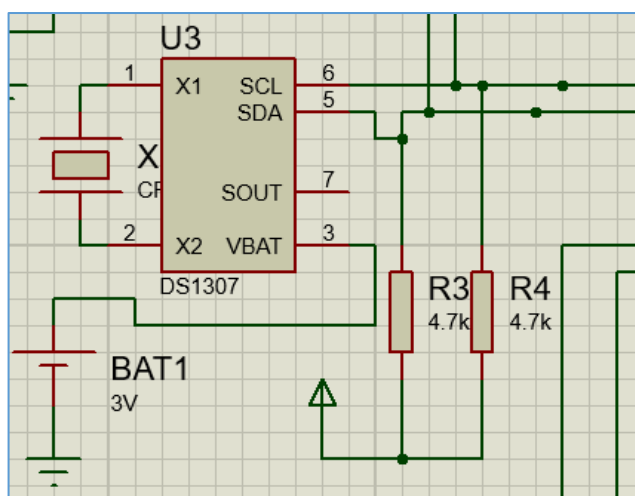


Figura 44: Diseño del DS1307 (Fuente: PROTEUS)

Para realizar todo el conexionado se ha consultado el *data sheet* del componente ^[10] ya que aunque parece muy intuitivo a través del software o de otros canales como foros de electrónica, estos al no ser fuentes oficiales puede haber errores.

7.1.2.6. Diseño del circuito de prueba del 24LC256

Para el diseño de la memoria EEPROM 24LC256, se ha utilizado el componente exacto disponible en las librerías de PROTEUS con el que se ha realizado el conexionado descrito en el diseño del equipo. Dos resistencias pull-up de 4,7k Ω para la conexión en serie del bus I2C. Además de la alimentación y el GND se han conectado los pines de selección A0, A1 y A2 que permiten seleccionar el dispositivo al que se le transmitirán o leerán datos.

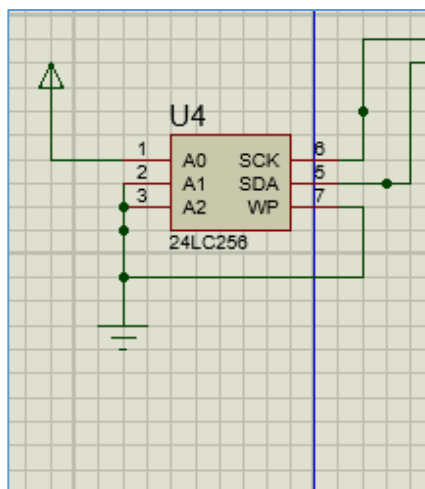


Figura 45: EEPROM 24LC256 (Fuente: PROTEUS).

Para realizar todo el conexionado se ha consultado el *data sheet* del componente ^[12] ya que aunque parece muy intuitivo a través del software o de otros canales como foros de electrónica, estos al no ser fuentes oficiales puede haber errores.

7.1.2.7. Diseño de pruebas del teclado matricial

Para el selector se ha implementado una serie de pulsadores que forman un teclado, la intención inicial fue implementar directamente un teclado matricial compacto, pero debido a los continuos problemas de simulación y fallos en las librerías del simulador PROTEUS se ha optado definitivamente por un teclado matricial casero construido con nueve pulsadores simples que permiten la selección de nueve tareas distintas del producto.

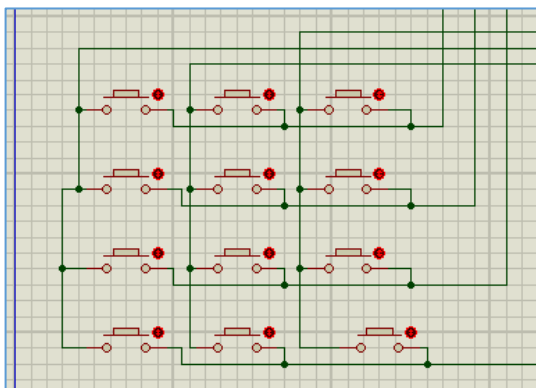


Figura 46: Teclado matricial (Fuente: PROTEUS)

El modelo de los pulsadores es el básico del programa con el que se puede simular cualquier tipo de modelo de pulsador en el mercado. En el diseño de prototipo de pruebas se ha implementado una matriz 3x4 dejando margen a posibles opciones a nivel de programación.

7.1.3. Visualización

7.1.3.1. LCD

El diseño del prototipo del LCD en PROTEUS ha sido de fácil implementación ya que las librerías del dispositivo funcionan perfectamente a través del puerto RD, a nivel de programación ha permitido resumir numerosas líneas de código de lectura y escritura en un par, haciendo posible el LCD de fácil aplicación. Además, se han realizado las conexiones necesarias de alimentación y masa.

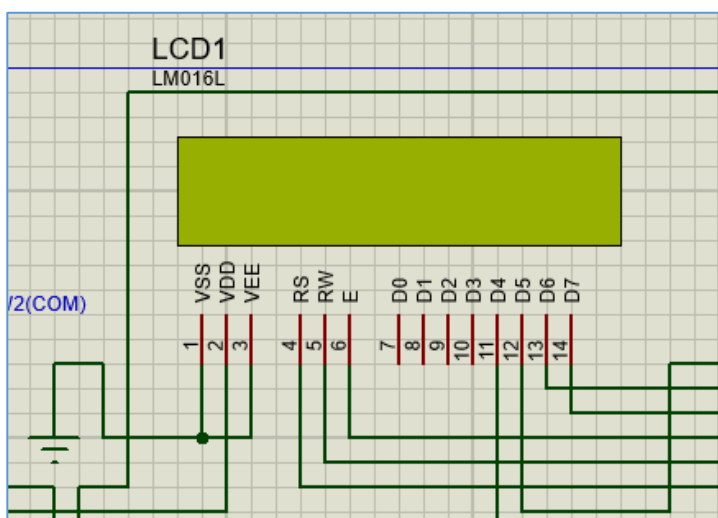


Figura 47: Diseño de prueba de LCD (Fuente: PROTEUS)

Para realizar todo el conexionado se ha consultado el *data sheet* del componente ^[11] ya que aunque parece muy intuitivo a través del software o de otros canales como foros de electrónica, estos al no ser fuentes oficiales puede haber errores.

7.1.3.2. Diseño del circuito de pruebas del HC-06

Debido a problemas de instalación de las librerías relativas a dispositivo HC-06, componente que proporciona *bluetooth* inalámbrico, no se ha podido simular en PROTEUS el programa de app inventor.

En tal caso se ha instalado un dispositivo de externo de lectura del bus serie UART, con puerto operativos RX y TX. Este permitirá la lectura de los datos enviados a través del canal y que supuestamente llegan al HC-06 para ser visualizados en el dispositivo móvil.

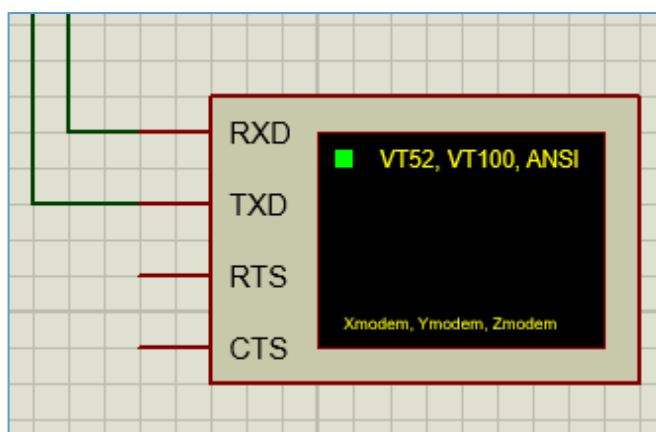


Figura 48: Dispositivo visualizador de información del canal serie UART (Fuente: PROTEUS)

7.2. Diseño de la Placa

El diseño de la placa se ha hecho con especial atención, ya que hay muchas variables que tratar y una simple mala conexión entre las muchas que hay, puede provocar un mal funcionamiento a gran escala complicado de detectar. El pre diseño se ha realizado a mano, en hoja de papel, de esta forma se han empezado a diseñar las posibles soluciones en cuanto a espacio y direcciones de los chips, sensores y toda clase de componentes.

Se han adquirido los conocimientos necesarios para poder diseñar circuitos con este software gracias principalmente a material de internet que incluyen muchos foros pero principalmente youtubers de habla inglesa e hispanohablantes.

También se han consultado principios básicos de uso del software en el manual básico del EAGLE^[2] cuya forma ordenada, instintiva y con imágenes permiten hacer consultas rápidas del funcionamiento de los comandos.

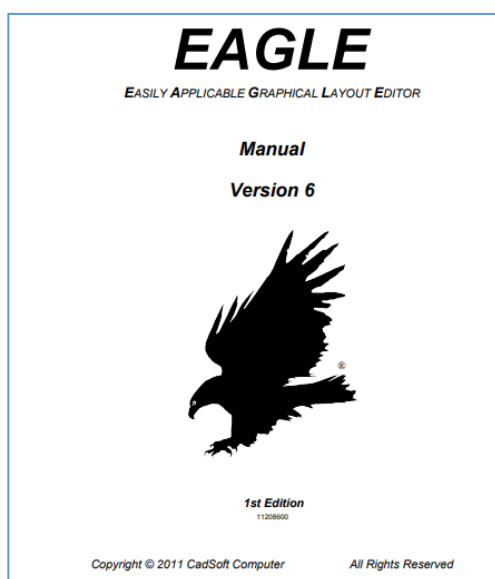


Figura 49: Manual del programa de diseño EAGLE (Fuente: EAGEL manual)

7.2.1. Inconvenientes de diseño con software

El diseño se ha llevado a cabo con el programa informático EAGLE dedicado a diseñar esquemas eléctricos y de PCB. Es importante tener en cuenta antes de empezar el diseño una serie de posibles inconvenientes que no dependen del diseñador, pero tienen posibles y múltiples soluciones.

Se incluyen los siguientes problemas a tener en cuenta:

- **Dimensionado:** Cada componente tiene un dimensionado que ha de corresponder con el sensor, chip o microcontrolador real. Se han solucionado constantes problemas de encaje pin con el agujero del fotolito. Es uno de los imprevistos que lleva más tiempo a solucionar porque la distancia entre pines puede variar en decimas de milímetro y al dibujar pines en serie los últimos quedan completamente desenchajados.
- **Espacio:** Se han solucionado problemas relativos al espacio que ocupa cada componente, sobre todo debido a que la versión gratuita del Eagle tiene un espacio limitado en el que se pueden montar.
- **Conexionado:** Como en la problemática anterior, la versión gratuita EAGLE no tiene activada una herramienta que permite el conexionado automático de todo el circuito de una forma optimizada y ordenada.
- **Librerías:** La gran mayoría de componentes por no decir todos se han diseñado manualmente, ya que el sistema de librerías de EAGLE, al menos el de prueba, no es muy eficiente.

7.2.2. Procedimiento del diseño del circuito

Como se ha comentado anteriormente el diseño del circuito y la PCB se han hecho mediante el software EAGLE. Una vez se han tenido en cuenta los posibles inconvenientes se ha continuado con los siguientes pasos para el correcto diseño del circuito:

En primer lugar se han distribuido todos los componentes de manera que teniendo en cuenta el espacio disponible, este quede totalmente optimizado en cuanto unos componentes no estén demasiado cerca de los otros.

Evidentemente el LCD se mantiene visible, el teclado formado por nueve pulsadores ha quedado colocado de una forma estructurada, elegante y no entorpece de ningún modo su uso, este es de fácil acceso. Ninguno de los componentes que pueden aumentar de temperatura especialmente el microcontrolador, están cerca del sensor de temperatura LM35. El manejo del potenciómetro es también de fácil acceso y el sensor de luz LDR no queda en ningún momento influenciado por la luz de la bicicleta. Finalmente el sensor hall se ha colocado en un extremo con espacio.

Dejando aparte todas las condiciones anteriores la principal distribución de los componentes es su proximidad a los puertos del microcontrolador, el objetivo de esta medida es posteriormente en el diseño de la PCB trazar el mínimo de circuitos, dejándolos lo más cortos posibles y sobretodo que no se entorpezcan entre ellos.

7.2.3. Procedimiento del diseño de la PCB

Para el diseño del PCB se ha utilizado también el software del EAGLE. Una vez diseñado el esquema, el mismo EAGLE tiene la opción de pasarlo a diseño de PCB en la que reúne todos los componentes con sus conexiones para que el usuario puede organizarlos de la forma que desee.

Para la organización de los componentes se ha seguido la descripción del apartado anterior, procedimiento del diseño del circuito. Además se han tomado dos medidas para la presentación de la PCB que son las siguientes:

En primer lugar y como medida de gran importancia en el diseño de la PCB, se ha decidido realizar el diseño en doble cara, básicamente por el número de conexiones que se entrelazan entre ellos haciendo imposible una capa de conexiones sencillas. Esto quiere decir que en la capa superior se han realizado una serie de conexiones junto al GND, una capa de masa general que permite conectar todas las masas directamente a la placa sin necesidad de realizar ninguna conexión directa de pista, se optimiza de esta forma el espacio para el trazado de las pistas. De la misma forma se ha hecho en la capa inferior en la que además de una serie de pistas optimizadas esta tiene la función genérica de Vcc, alimentación. Se muestran las dos capas en la (Figura 50).

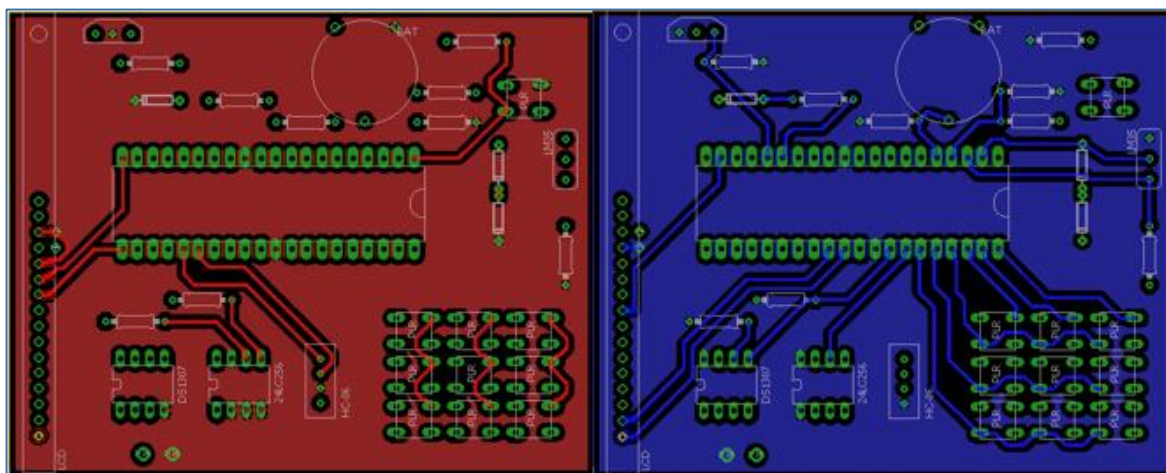


Figura 50: Doble capa de la PCB Superior e inferior (Fuente: EAGLE)

En segundo lugar se ha respetado la normativa de pistas exigida por la universidad en la que tanto la distancia entre pistas como su anchura no ha de ser inferior 0,45mm siendo la medida básica 1mm tanto de distancia como de grosor. Así también el diámetro de las vías es de 1mm como mínimo y 1,5mm de diámetro basico.

8. Simulaciones

Como ya se ha explicado el programa de simulación es el PROTEUS DESIGN SUITE la herramienta que permite diseñar el hardware de pruebas y subir la programación software del microcontrolador.

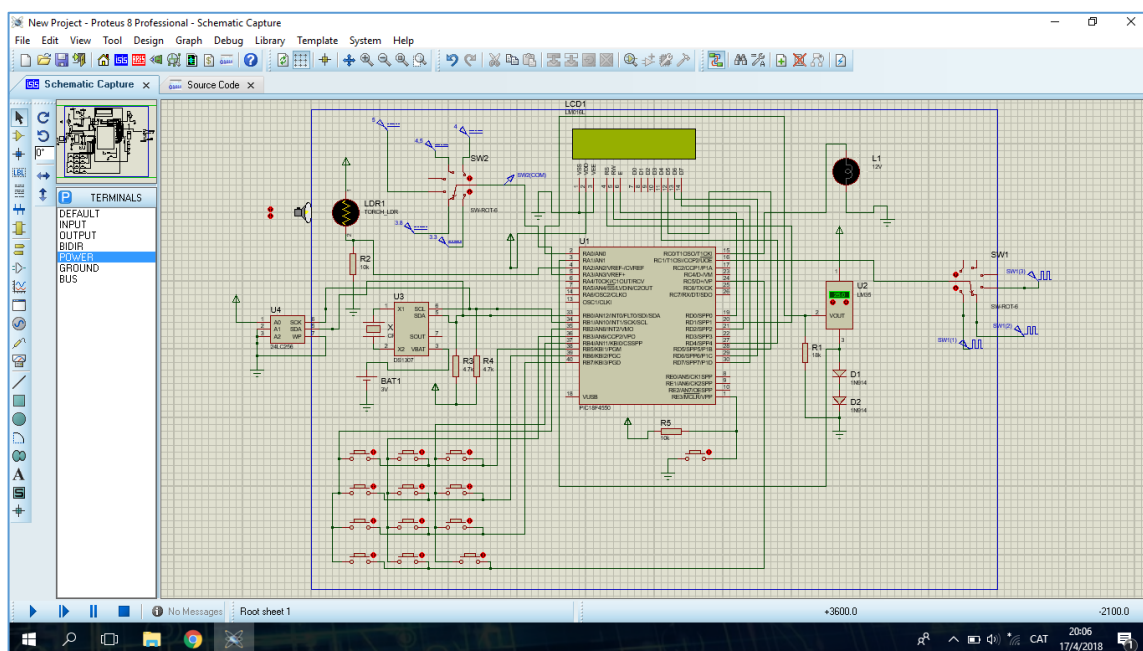


Figura 51: Software de diseño PROTEUS SUITE DESIGN.

Junto al PROTEUS se ha utilizado el compilador CCS que permite programar la programación del microcontrolador en C, este contiene los operadores estándar del lenguaje y funciones incorporadas en bibliotecas que son específicas de registros para microcontroladores PIC.

8.1. Simulación del menú principal

En primer lugar el primer sistema de simulación que debe funcionar, es el menú principal. Asociado directamente al teclado matricial, el menú consiste simplemente en el uso de una sentencia case de programación en C en el que una variable declarada como select, selecciona el modo se va a ejecutar. Se han programado los siguientes:

- Temperatura: Muestra la temperatura ambiental.
- Luz: Activa o desactiva la luz en función de la luz ambiental.
- Tiempo: Muestra tiempo y fecha actual.
- Distancia: Muestra la distancia recorrida.
- Batería: Muestra la carga de la batería actual en tanto por ciento.
- Barrido: Inicio del registro del ciclo de parámetros.
- Botón atrás: Botón para retroceder.

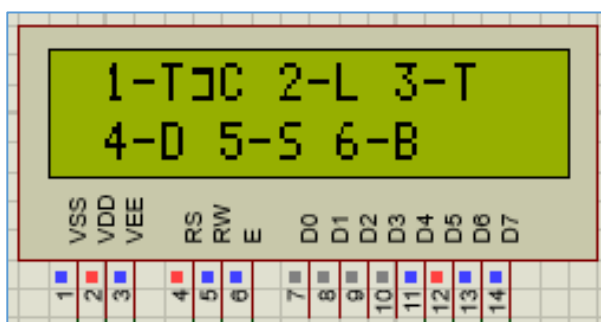


Figura 52: Visualización del menú (Fuente: PROTEUS)

8.2. Simulación y lectura de la Temperatura

La primera opción del menú muestra directamente la temperatura ambiental, señal obtenida por el sensor LM35, el programa entra en un bucle que realiza la conversión analógica digital de los puertos A0 y A1, el primero para calcular la temperatura en caso que sea positiva y el segundo para temperaturas negativas, el signo se ha añadido a parte y con sentencias de programación que debe cumplir las variables. El bucle en cuestión se repite hasta que se pulse el botón de atrás marcado como el botón número 7.

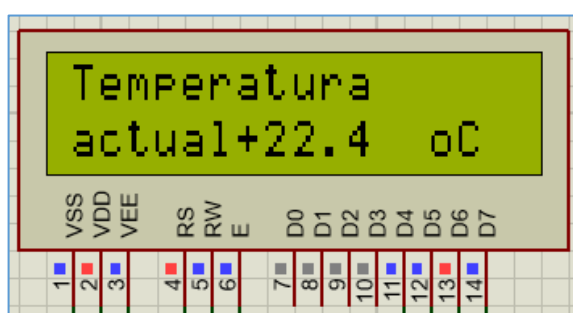


Figura 53: Lectura temperatura positiva (Fuente: PROTEUS).

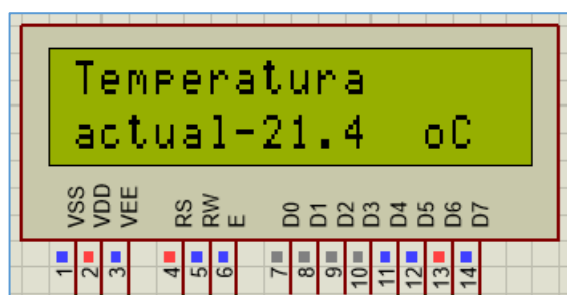


Figura 54: Lectura temperatura negativa (Fuente: PROTEUS).

Para la anterior simulación se han tenido en cuenta las librerías disponibles del LCD que permiten una fácil visualización de los valores y la posibilidad de introducir caracteres como signos para la mejor presentación del producto.

8.3. Simulación del circuito de iluminación Luz

En la segunda opción del menú, ejecuta el programa de activación lumínica en caso que no hubiera suficiente iluminación exterior gracias al sensor de luz LDR. En este caso no se ha ejecutado ningún bucle cerrado, el software detecta si después de la conversión analógica digital el valor numérico es inferior a 75 como se describe en el diseño del equipo, en tal caso se activa la iluminación.

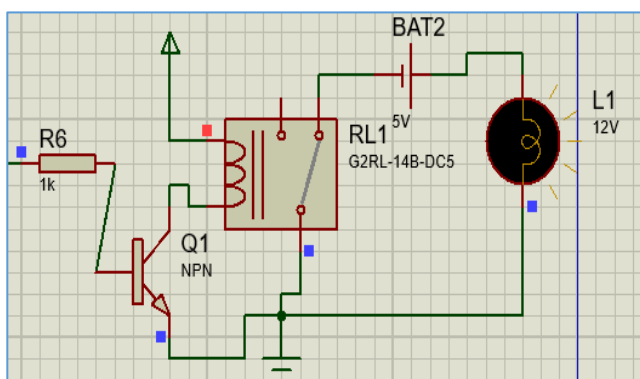


Figura 55: Simulación del foco encendido (Fuente: PROTEUS).

Para la anterior simulación se han tenido en cuenta las librerías disponibles que permiten la visualización animada de los componentes como el relé o el foco de iluminación.

8.4. Simulación del reloj a tiempo real

La tercera opción en el menú principal es la posibilidad de visualizar la hora y fecha actuales con el dispositivo DS1307. Para ello en primer lugar se tiene que configurar la información, así, una parte de la programación está destinada a ello. A través de los primeros dos botones hay la posibilidad de seleccionar los datos con el primero y cambiarlos con el segundo botón.

La programación software sigue todos los patrones de configuración del dispositivo a través del canal de transmisión de datos en serie I2C:

El procedimiento consiste en la lectura de cada dato, la conversión de binario a decimal y finalmente la visualización de este en el LCD. El procedimiento de escritura es básicamente el mismo excepto la conversión de decimal a binario de los datos a escribir.

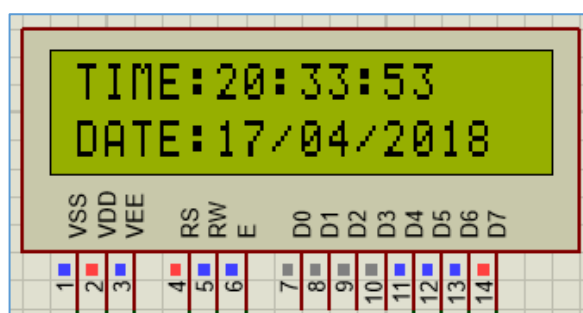


Figura 56: Visualización de fecha y hora (Fuente: PROTEUS)

Para la anterior simulación se han tenido en cuenta las librerías disponibles del LCD que permiten una fácil visualización de los valores y la posibilidad de introducir caracteres para la mejor presentación del producto.

8.5. Simulación y lectura de la distancia recorrida

La cuarta opción de visualización es la que permite al usuario informarse del recorrido realizado. Como en los casos anterior el programa ejecutado entra en un bucle cerrado hasta que se salga de este clicando el botón de atrás. Una vez dentro del programa se inicia el conteo cada vez que el sensor Hall es activado, es decir, cada vez que la rueda realiza una vuelta y el imán pasa por delante del sensor. El sistema consta de dos variables, la primera multiplica las veces que se ha activado el sensor hall por el perímetro de la rueda en metros, obtenido la distancia recorrida. La segunda variable permite contar los kilómetros cada vez que la primera llega a 1000 metros y es reiniciada.

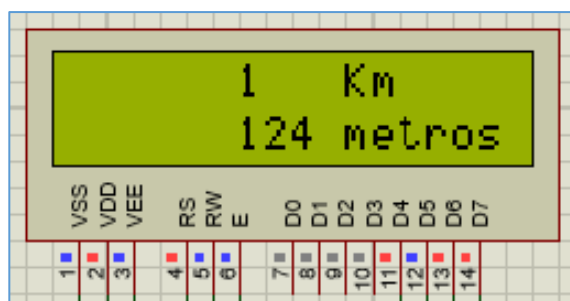


Figura 57: Visualización del recorrido (Fuente: PROTEUS).

Para la anterior simulación se han tenido en cuenta las librerías disponibles del LCD que permiten una fácil visualización de los valores y la posibilidad de introducir caracteres para la mejor presentación del producto.

8.6. Simulación del control de la carga de la batería

La quinta opción del menú es la simulación de la medición del porcentaje de batería restante que se ha calculado como se ha explicado en el diseño del prototipo de pruebas en la que hay un divisor de tensión con un potenciómetro regulable que simula una batería cargada. Al igual que en las otras opciones el cálculo del porcentaje se realiza dentro de un bucle del que solo se puede salir si se pulsa el botón de atrás.

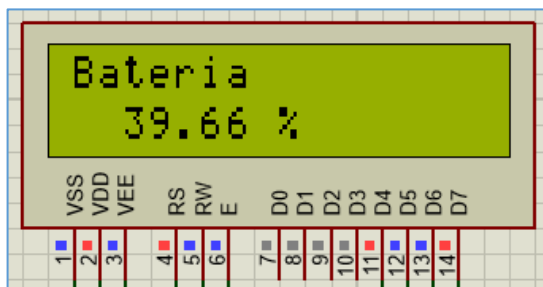


Figura 58: Visualización de porcentaje de batería (Fuente: PROTEUS).

Por otro lado en caso que la batería descienda del 20% se otra sentencia informa al usuario que se debe recargar.

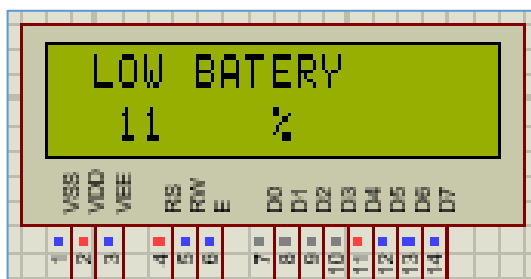


Figura 59: Visualización del porcentaje de la batería por debajo de 20% (Fuente: PROTEUS).

8.7. Simulación del registro del ciclo lectura de parámetros

El sexto botón es el destinado a la simulación del registro del ciclo de parámetros del programa que permite obtener datos de las cinco variables disponibles y guardarlas en la memoria externa 24LC256. La simulación se ha podido realizar y visualizar en el LCD, es básicamente equivalente a las simulaciones obtenidas en este mismo apartado de simulaciones a través del software PROTEUS, la única diferencia es que van rotando y se van guardando en la memoria.

El ciclo se ha realizado mediante una sentencia case que salta de una a la otra en función de la distancia recorrida por la bicicleta, de esta manera se cumple el objetivo de almacenar los datos cada vez que se recorre una distancia.

Como en los casos anteriores el ciclo de datos se realiza dentro de un bucle del que se puede salir pulsado el botón de volver.

Para el almacenamiento de los datos se ha realizado un algoritmo en el que cada lectura de dato se incrementa en una unidad el posicionamiento de guardado en la memoria, ocupando el siguiente Byte disponible de la EEPROM.

8.8. Simulación de guardado en la memoria externa

Para la verificación de la simulación para guardar datos en la memoria EEPROM, dispositivo 24LC256, se ha realizado un algoritmo extra en la adquisición de datos para el sensor de temperatura. Este consiste en la posibilidad de almacenar el valor si se pulsa el segundo botón y una vez se ha guardado, es visualizado directamente de la memoria:

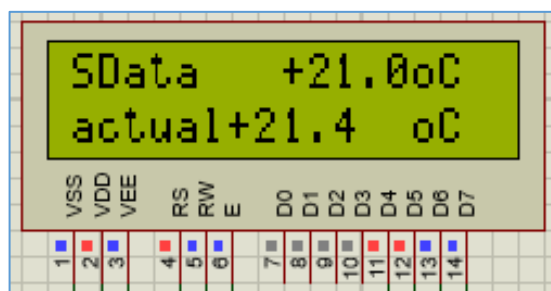


Figura 60: Visualización de los datos guardados (Fuente: PROTEUS).

De la misma forma clicando el botón, se sale del bucle de autoguardado del valor a visualizar:

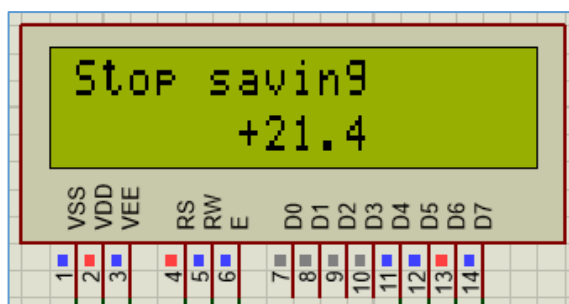


Figura 61: Visualización de la finalización de los datos guardados (Fuente: PROTEUS).

8.9. Simulación del botón de retroceso

El último botón disponible, el número 7, es el botón de “atrás”. Es el botón que permite parar la visualización de todos los sensores y volver al menú principal.

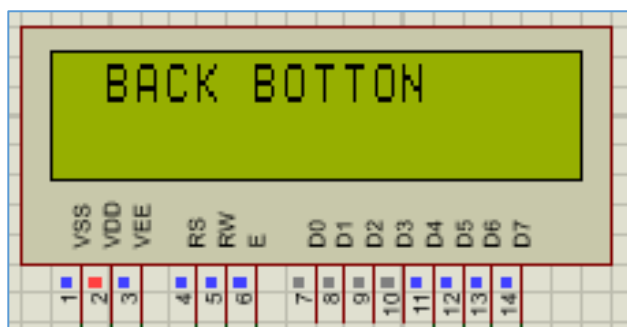


Figura 62: Visualización del botón para volver (Fuente: PROTEUS).

8.10. Simulación del dispositivo HC-06

Como se ha comentado en el diseño del prototipo de pruebas la simulación del HC-06 se realiza con un dispositivo disponible en PROTEUS que permite visualizar la información de la salida del puerto UART, o del pin de transmisión en serie Tx. En la siguiente imagen se visualiza la batería cargada al 77%.

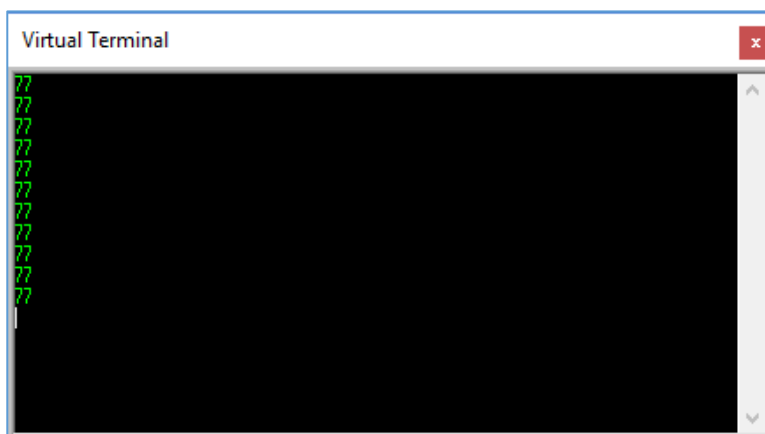


Figura 63: Visualización del porcentaje de batería a través del canal serie Tx (Fuente: PROTEUS)

9. Implementación

9.1. Grabado en la placa

9.1.1. Fotolito

Una vez se ha diseñado la PCB mediante el software EAGLE, se ha impreso el fotolito de las dos caras en formato pdf, se han quitado todas las siluetas de los componentes y puntos de referencia, en el fotolito solo quedan las pistas, vías, y los puntos de unión como se muestra en la Figura 64. Una vez impreso en papel transparente se unen las dos caras con doble copia para asegurar que la parte con tinta queda totalmente opaca. En medio de esta unión es donde se pondrá la placa positiva para realizar la insolación a doble cara.

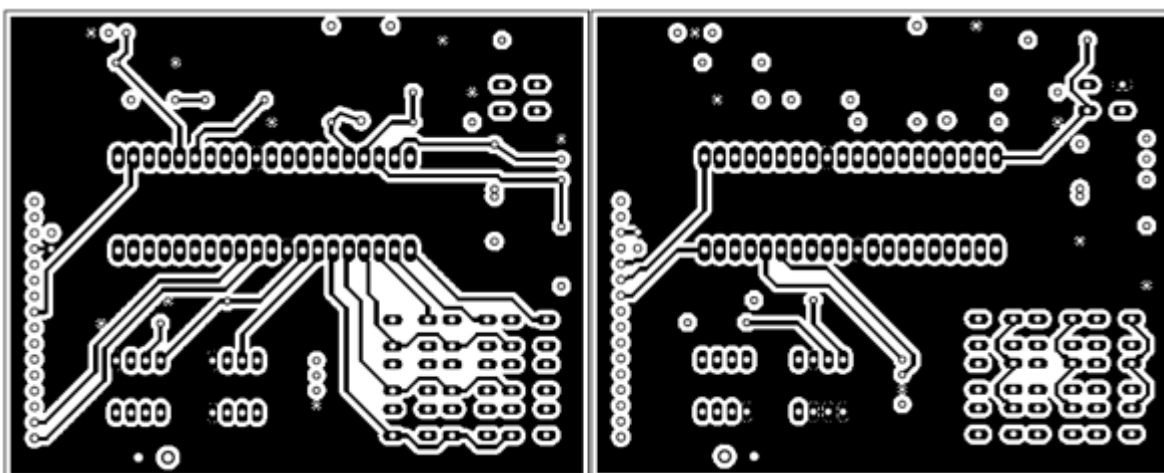


Figura 64: Doble cara del fotolito inferior y superior (Fuente:EAGLE)

9.1.2. Grabado

El procedimiento se ha llevado a cabo por el departamento de electrónica de la universidad que son los profesionales que tienen acceso a los dispositivos y materiales para realizar el grabado de la placa. El proceso de insolación es de una duración aproximadamente de tres minutos en la que la placa se coloca bajo la luz ultravioleta con la doble capa de fotolito para marcar el circuito diseñado. Seguidamente se realiza el revelado con el que se ha sumergido la placa en agua templada con sosa caustica, se ha sujetado la placa con cuidado ya que debe afectar a las dos caras. En menos de un minuto el barniz expuesto a la luz ultravioleta de la placa empieza a oscurecerse y a desaparecer, se ha tenido sumo cuidado de no dejar más tiempo del debido la placa bajo la solución ya que puede afectar al resto del barniz que cubre las pistas.

Finalmente se ha realizado el atacado con una solución ácida de agua, ácido clorhídrico y dióxido de hidrógeno que permitirá disolver el cobre que no es parte de las pistas del circuito. Se ha tenido especial cuidado con la solución ya que es muy corrosiva. Una vez retirado el cobre se repite el procedimiento de rebelado para quitar el barniz que cubre las pistas y de esta manera obtener definitivamente el circuito.



Figura 65: Insoladora para placas positivas (Fuente: Google).

9.2. Preparación de la placa

Una vez se ha obtenido la placa procesada se han comprobado posibles desperfectos de esta como partes granuladas del cobre que no cubren la superficie de la resina, se han cubierto con un tinte conductor. Otros desperfectos como pequeños cortocircuitos que puede provocar el cobre no disuelto.

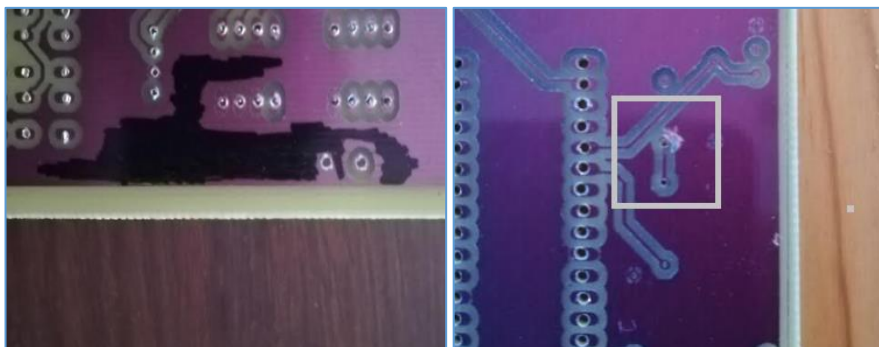


Figura 66: Tinte conductor y cortocircuito. (Fuente: Fotografía).

Para la correcta conexión de todas las pista, y para detectar posibles cortocircuitos se ha utilizado un función de un multímetro que emite un sonido en caso de que haya conexión de cable a cable.



Figura 67: Multímetro (Fuente: Fotografía).

Una vez se han corregido errores de conexionado se ha procedido a agujerear la placa, para ello se tiene en cuenta el diámetro de los agujeros establecido en el diseño de la PCB con el software EAGLE, se han escogido las brocas adecuadas y se ha procedido a taladrar la placa.



Figura 68: Brocas de 1,2mm o inferiores (Fuente: fotografía)

Una vez se han realizado los agujeros de cada componente se ha comprobado que el mismo entre y quede ajustado en su sitio, de manera que componentes con más pines como el microcontrolador o el LCD se repasen los agujeros o incluso se aumente el diámetro de estos en caso que no cuadren todos los pines. Finalmente la placa queda como muestra la siguiente figura:

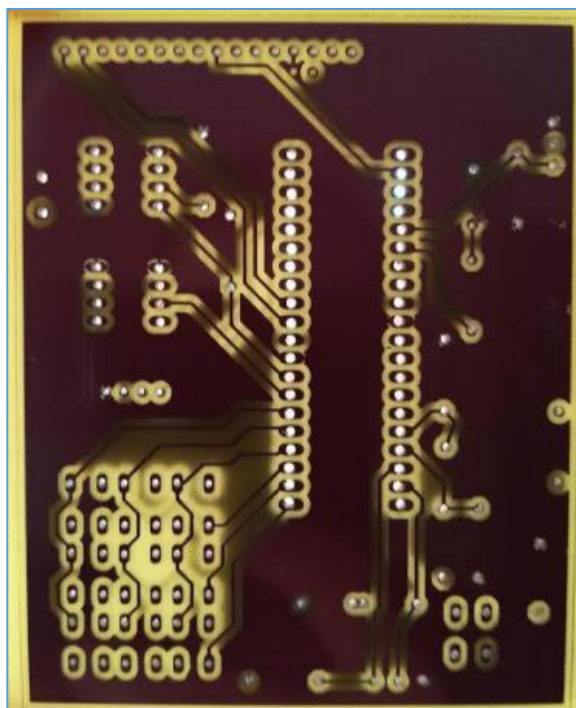


Figura 69: Placa agujereada (Fuente: Fotografía)

Para acabar los componentes se han ido soldando de uno en uno de manera que por ejemplo haya el espacio suficiente entre los pulsadores para el soldador y el estaño. Las patas de los componentes se han cortado para que quede lo más presentable posible.

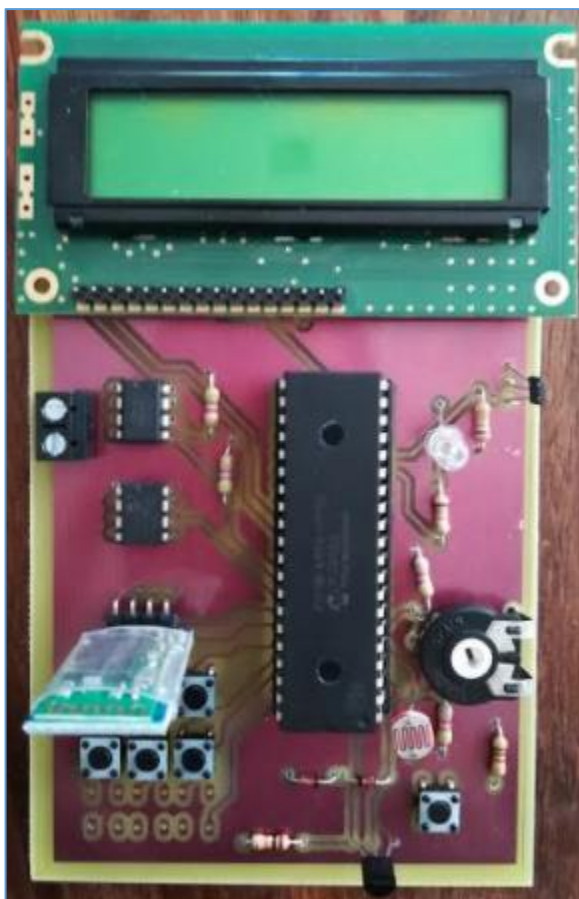


Figura 70: Placa con todos los componentes (Fuente: Fotografía)

9.3. Implementación del hardware

En primer lugar la implementación física del producto se ha realizado en una *protoboard* en la que se han ido comprobando el correcto funcionamiento del software y de cada sensor. Para evitar una gran multitud de cableado que moleste y entorpezca el trabajo, se han implementado los sensores uno por uno sin contar con los pulsadores debido a que la disponibilidad de sus cuatro patas hacia muy complicado su conexionado a la *protoboard*. Se ha decidido realizar el funcionamiento de pulsador manualmente con un cable que conecte las distintas combinaciones de los puertos RB2 a RB7.

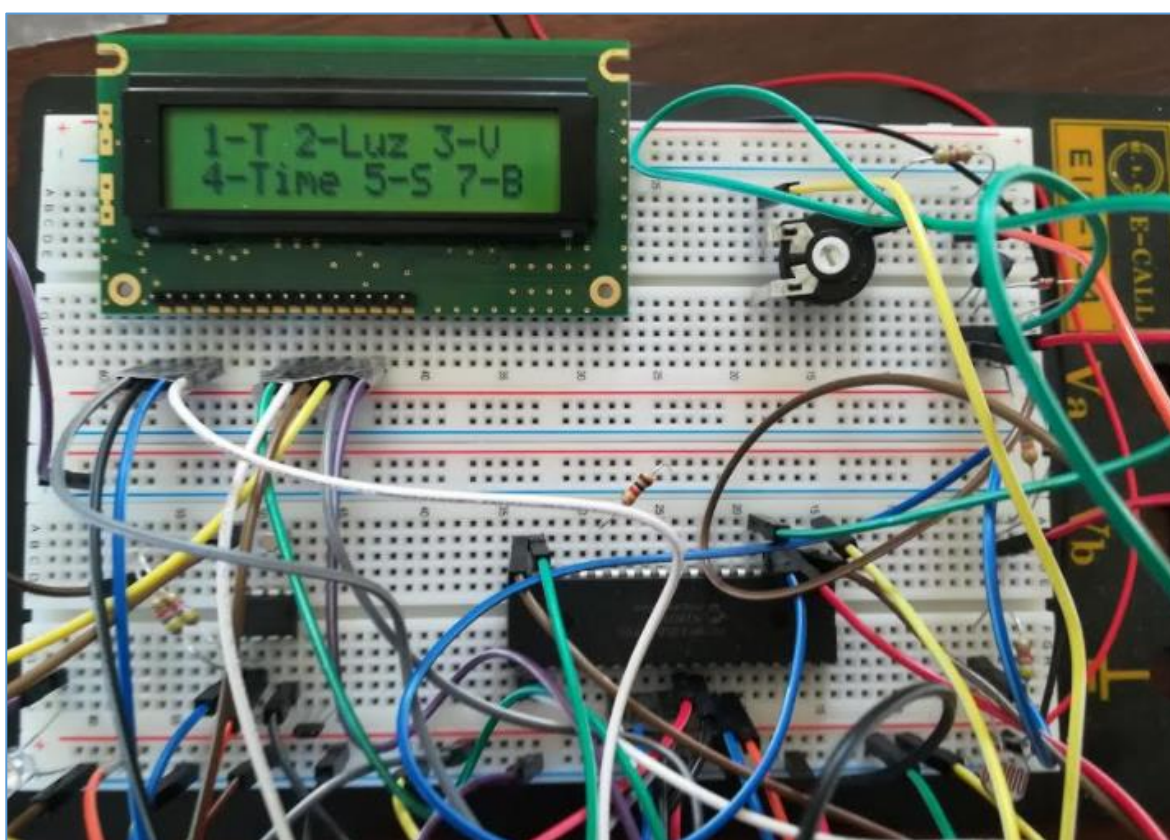


Figura 71: Implementación en protoboard (Fuente: fotografía).

Por otro lado se ha montado la alimentación del circuito, con una fuente de 4,5V ha sido suficiente ya que se puede alimentar todos los componentes del circuito, en general como el microcontrolador el rango operativo es de 2V a 5,5V.

El encapsulado de la alimentación está compuesto por tres pilas de 1,5V conectadas en series como se muestra en la imagen:



Figura 72: Alimentación del circuito (Fuente: fotografía).

Finalmente para la conexión de la alimentación con la placa se ha soldado una regleta a un extremo de esta para facilitar su conexión y desconexión.

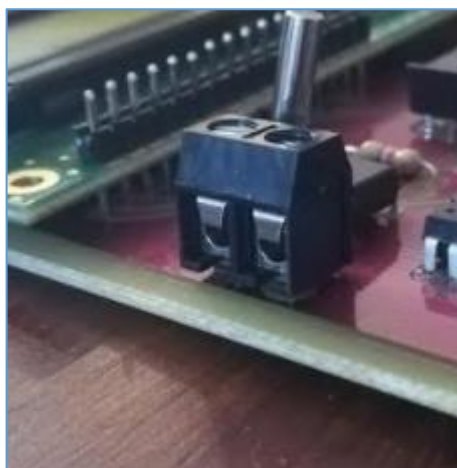


Figura 73: Regleta de alimentación (Fuente: fotografía).

9.3.1. Microcontrolador

Para la programación del microcontrolador se ha utilizado el software del compilador PICK IT3, se ha descargado el programa y se ha realizado el conexionado con el microcontrolador (Figura 74: Conexionado PICK IT3 con el 18f4550 (Fuente: Google)Figura 74). Del número 1 al 5: reset, pines de alimentación y masa, pin de lectura y pin de escritura.

El cableado de cinco pines se ha unido por defecto para tener una mayor facilidad al utilizarlo y se ha destinado una pequeña protboard preparada para programar el microcontrolador y agilizar el trabajo.

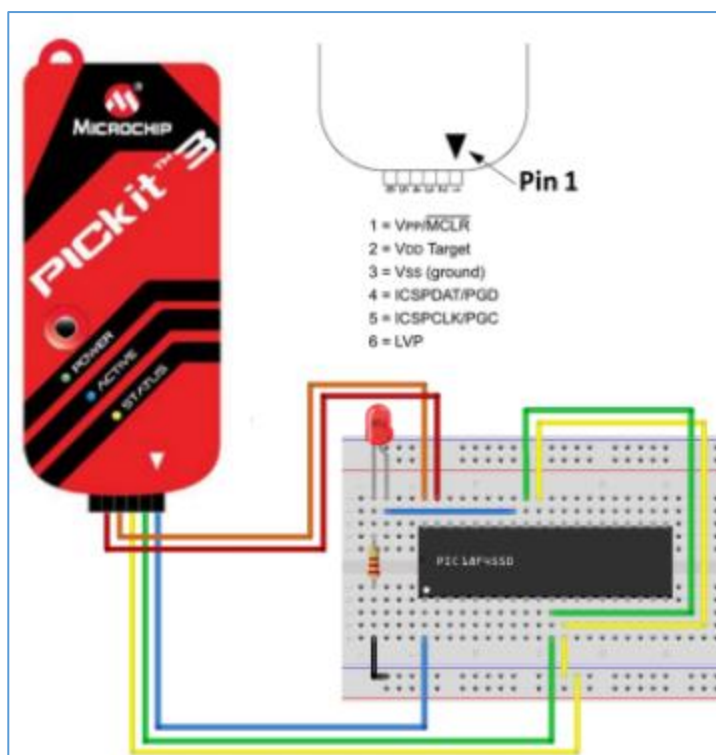


Figura 74: Conexionado PICK IT3 con el 18f4550 (Fuente: Google)

9.3.2. Implementación de los componentes

Implementación del PIC 18F4550

La implementación del PIC en la placa consiste básicamente en soldar el soporte al que ira sujeto el microcontrolador. Se han hecho agujeros con brocas mayores a 1 mm para facilitar el encaje de todos los 40 pines. El objetivo del soporte es la posibilidad de quitar y poner el PIC para poder ser reprogramado. El único inconveniente a la hora de encajarlo es estar atento al sentido del microcontrolador ya que el diseño del PIC está dirigido hacia abajo.



Figura 75: Soporte del microprocesador (Fuente: fotografía).

Implementación del LDR

La implementación final del sensor de luz se ha realizado tal cual se describe en el diseño del equipo con el LDR y una resistencia formando un divisor de tensión.



Figura 76: Implementación LDR (Fuente: fotografía).

La diferencia significativa es en el segundo circuito, tanto la parte de control: resistencia, transistor bobina y diodo, como la parte de iluminación: batería y foco, se han substituido por una resistencia y un LED que simbolizaran la luz del foco.

El motivo del cambio es debido a la falta de tiempo y falta de recursos económicos, pero principalmente al espacio limitado que proporciona la versión gratuita del software de diseño EAGLE, con el que se han comprimido mucho espacio.



Figura 77: Implementación del circuito de iluminación (fuente: fotografía)

Para la implementación del LED se ha calculado la resistencia que limita la corriente que podría quemar el componente, además, se ha tenido en cuenta el sentido del LED que funciona como diodo emisor de luz con un Ánodo y Cátodo:

$$R = \frac{5V - 2V}{20mA} = 150\Omega \quad \text{(Ec. 19)}$$

Implementación del circuito de control de carga de la batería

Una batería típica de plomo para bicicletas eléctricas tiene un coste económico estimado de unos 200 euros, por eso se ha optado por la sustitución de esta por un potenciómetro que simulara el estado de carga. El circuito se ha diseñado con un divisor de tensión alimentado con 5V, una resistencia superior fija de 10k Ω y un potenciómetro variable que oscila entre unos 100 Ω y 1 M Ω . De esta manera la caída de tensión en el potenciómetro será regulable como muestran las (ec.21) y (ec.23), entre el 1% y el 100%.

$$V_{min} = 5V \frac{100\Omega}{(10k\Omega + 100\Omega)} = 49,5mV \quad (\text{Ec.20})$$

$$V\% = \frac{0,0495V}{5V} \approx 1\% \quad (\text{Ec.21})$$

$$V_{max} = 5V \frac{1M\Omega}{(10k\Omega + 1M\Omega)} = 4,95V \quad (\text{Ec.22})$$

$$V\% = \frac{4,95V}{5V} \approx 100\% \quad (\text{Ec.23})$$

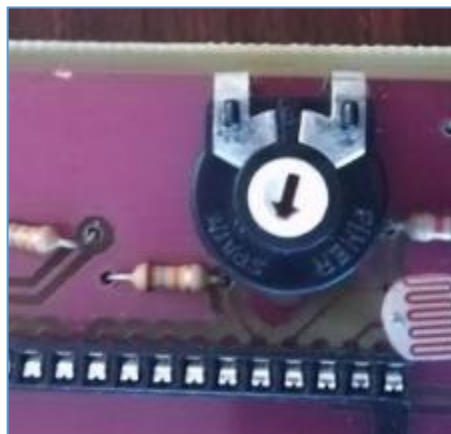


Figura 78: Implementación del potenciómetro (Fuente: fotografía)

Implementación del circuito del DS1307

Para la implementación del reloj a tiempo real se ha soldado el componente a la placa en primer lugar y una vez fijo se ha el cristal de cuarzo y la batería externa en los pines correspondientes:



Figura 81: Implementación DS1307 (Fuente: fotografía).

Implementación del circuito del 24LC256

La implementación de la memoria EEPROM se ha realizado como se describe en el diseño del equipo. La memoria se ha colocado también como se ha descrito en el apartado de diseño de la placa:



Figura 82: Implementación 24LC256 (Fuente fotografía)

Implementación del teclado matricial

La implementación del teclado matricial se ha realizado como se describe en el diseño del equipo. Finalmente se han colocado nueve pulsadores, se han recortado las patas de los pulsadores para poder colocarlos adecuadamente y no queden parcialmente flotado encima de la placa:

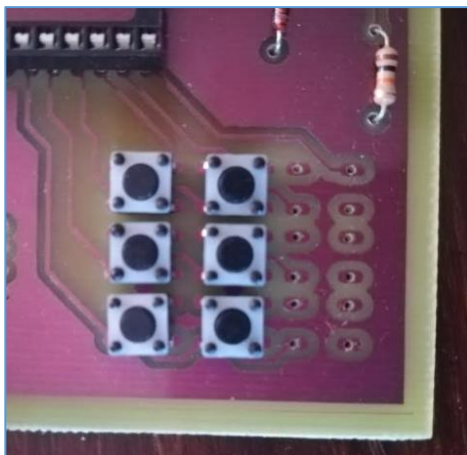


Figura 83: Implementación teclado matricial (Fuente fotografía).

Implementación del LCD

La implementación del LCD al igual que el microcontrolador, se han realizado agujeros mayores a 1 mm para poder encajar todos los pines, para el resto se ha dejado igual al diseño del prototipo y de la placa:



Figura 84: Implementación LCD (Fuente: fotografía)

Implementación del HC-06

La implementación del módulo de *bluetooth* se ha implementado como se describe en el diseño del equipo, el dispositivo queda instalado en posición vertical así que se tendrá sumo cuidado a la hora de transportar el producto:



Figura 85: Implementación del HC-06 (Fuente: fotografía)

10. Comunicación a través del HC-06

La comunicación inalámbrica de *bluetooth* a través del dispositivo HC-06 necesita una interfaz de usuario en el dispositivo móvil, con la que se recibirán los datos enviados desde el dispositivo y el usuario podrá interactuar para conectar y desconectar el *bluetooth*.

La interfaz se ha diseñado también con la aplicación app inventor con la que se ha programado la transmisión de datos al dispositivo móvil descrito en el apartado del diseño del software.

Se ha diseñado una interfaz de usuario sencilla con la que se aprecie rápidamente la información a visualizar debido a que el usuario debe estar atento al control del vehículo.

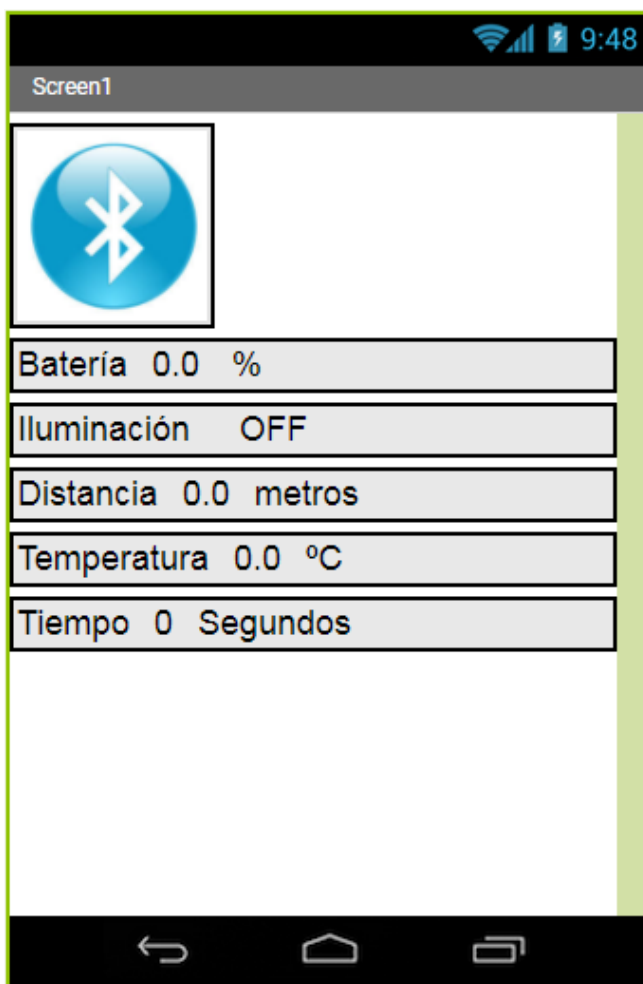


Figura 86: Interfaz de usuario (Fuente: App inventor).

La interfaz cuenta en primer lugar con un botón de conexión a *bluetooth* del dispositivo que le permitirá seleccionar los canales de conexión en este caso el módulo HC-06.

En segundo lugar tendrá las variables a visualizar, el valor de la variable puede ser recibirse desde el software individual de cada componente como del barrido general.

La descarga de la interfaz es sencilla, para ello el usuario debe descargarse la aplicación de App inventor disponible en cualquier *play store* de un dispositivo móvil y seguidamente introducir el código del producto. Automáticamente se instalará el programa en su dispositivo con el que tendrá ya acceso a la interfaz.

Finalmente se debe activar el *bluetooth* del móvil y una vez reconocida la identificación del producto introducir la contraseña 1234.

11. Normativa

El producto final a comercializar cumple con las normativas siguientes respecto a los grados de protección IP y sustancias peligrosas:

CEI 60529 *Degrees of Protection*: Es la normativa internacional que hace referencia al grado de protección del dispositivo. En el caso de este producto se tomara un grado de protección IP67.

- IP: Significa las siglas en ingles de *Ingress Protection*.
- El valor 6 indica el grado de protección en polvo que bajo ninguna circunstancia debe entrar.
- EL valor 7 indica el grado de protección frente a líquidos, en el que debe resistir sin filtración alguna ser sumergido 1 metro durante 30 minutos.

Rohs 2002/95/CE: Es la normativa de restricción de ciertas sustancias peligrosas en dispositivos electrónicos. La normativa restringe las siguientes seis sustancias:

- Plomo
- Mercurio
- Cadmio
- Cromo IV
- PBB
- PBDE

12. Posibles mejoras

Una de las posibles mejoras del proyecto discutidas ya el principio de este es la posibilidad de transmitir datos a un ordenador. Para ello se utilizaría un *serial port expander* que hace la función de multiplexor 8:1, este dispositivo permite utilizar múltiples componentes en los puertos de transmisión de datos serie UART con pines Tx y Rx. Así con tres puertos extra se podrían implementar hasta 8 dispositivos en el bus serie.

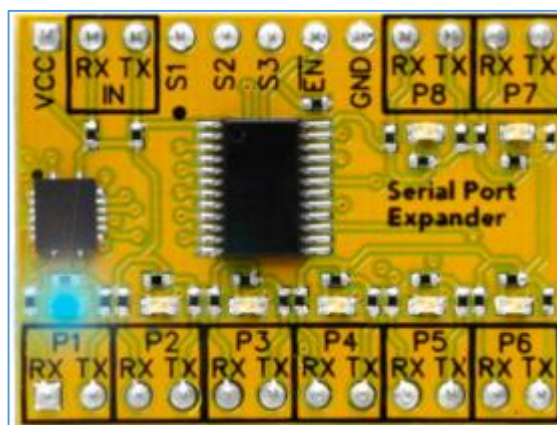


Figura 87: Multiplexor 8:1 para expandir puerto serie (Fuente: Serial port expander datasheet ^[17]).

Una vez se expanda el puerto serie se conectaría el dispositivo MAX 232, es un convertidor de voltaje que permite adaptar señales del puerto serie RS232 a salidas TTL compatibles con circuitos digitales. Lo que permitiría conectar el ordenador al microcontrolador externo.

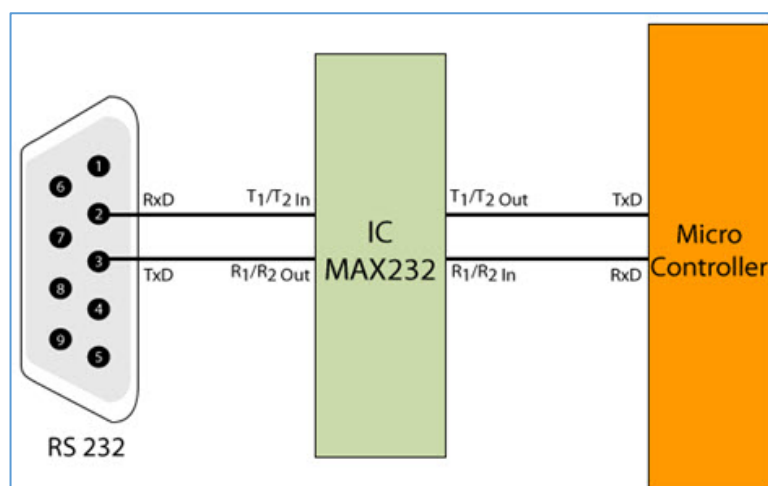


Figura 88: Interfaz de conexión (Fuente: google)

13. Marketing y venta del producto

Este apartado se ha dedicado a los pasos que se van a seguir para la comercialización del producto ya que una buena campaña de marketing aplicando conocimientos de venta a este tipo de producto, son esenciales para que llegue al usuario.

En primer lugar se ha optado por la venta online del producto ya que hoy en día no es necesario la opción de tener una tienda física además de los costes asociados a esta. Para ello se ha creado una cuenta en la página web shopify.com. Shopify es una empresa de Canadá dedicada al comercio electrónico que permite crear y editar la tienda online y obviamente presentar los productos. A diferencia de otros métodos más complicados, esta empresa cubre todo el sistema de seguridad electrónica a la hora de comprar y vender el producto.



Figura 89: Sitio web Shopify con la presentación del dispositivo (Fuente: Shopify web^[13]).

Una vez creada la tienda online, el propietario tiene el link de esta para referenciarla a otros sitios web, campañas de marketing, anuncios etc.

En segundo lugar se ha pensado los distintos métodos para anunciar el producto:

- Instagram: A través de Instagram se crea la página del producto en el que se añadirán fotografías y videos con su funcionamiento, es importante que se muestren personas con las que el consumidor pueda asociarse o crear vínculo.
- Facebook: De la misma forma que en Instagram se utilizara FaceBook para anunciar el producto.
- Google: Se anunciará también el producto en imágenes de google, este método es de pago a diferencia de las redes sociales.
- YouTube: Se podrá anunciar el producto través de anuncios y videos en YouTube.
- FaceBook Ads: Es el método más efectivo y también de pago. FaceBook Ads cuenta con toda la información de los usuarios de FaceBook: Edad, genero, aficiones, amistades, likes, localización, gustos, familiares etc. Lo que permite filtrar todos los usuarios que no están interesados en el producto y de esta forma también centrarse más en los que por ejemplo visitan usualmente páginas de bicicletas eléctricas.

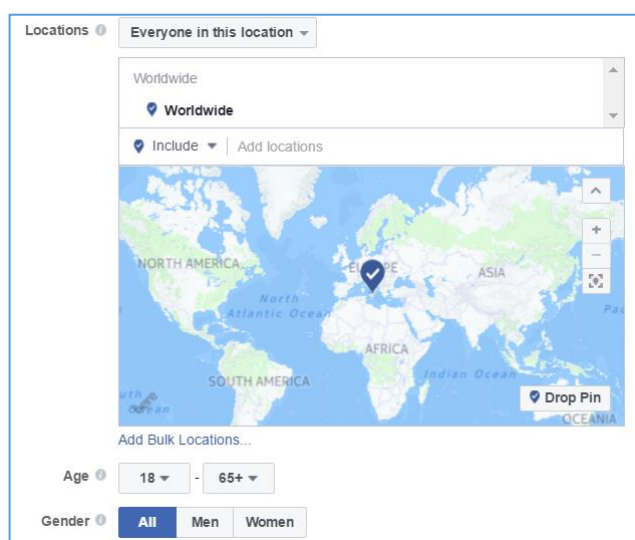


Figura 90: Método de filtrado por localización (Fuente: FaceBook ads).

Una vez establecidos y optimizados los anuncios, estos son referenciados a la página web de manera que el usuario al clicar va directamente a la Shopify web del producto.

14. Planificación y diagrama Grantt

Para la planificación del proyecto se han seguido los pasos obvios y razonables para poder realizarlo de forma continuada, con el mismo método de ejecución en todos los casos, planificando una estructura previa y realizando el trabajo después.

La estructura general empieza por el diseño tanto de software como de hardware, simulando la estructura una vez los primeros pasos se han realizado y antes de implementar físicamente, de esta manera se corrigen fallos. Finalmente se ha implementado y se ha pulido los pequeños errores.

| Tarea | Fecha inicio | Fecha final | Septiembre | | Octubre | |
|--|--------------|-------------|------------|--------|---------|--------|
| | | | 01-sep | 15-sep | 01-oct | 15-oct |
| Tiempo dedicado al diseño del software | 01/09/2018 | 30/10/2018 | | | | |
| Tiempo dedicado a la simulación | 15/09/2018 | 30/10/2018 | | | | |
| Tiempo dedicado al diseño del hardware | 15/10/2018 | 30/12/2018 | | | | |
| Tiempo dedicado al montaje del hardware | 15/01/2018 | 30/02/2018 | | | | |
| Tiempo dedicado a la depuración del software | 01/02/2018 | 30/02/2018 | | | | |
| Tiempo dedicado a pruebas funcionales | 15/02/2018 | 14/04/2018 | | | | |
| Tiempo dedicado a la confección de documentación | 01/03/2018 | 30/04/2018 | | | | |

| Noviembre | | Diciembre | | Enero | | Febrero | | Marzo | | Abril | |
|-----------|--------|-----------|--------|-------|-------|---------|--------|-------|-------|-------|-------|
| 01-nov | 15-nov | 01-dic | 15-dic | 01-En | 15-En | 01-feb | 15-feb | 01-Ma | 15-Ma | 01-Ab | 15-Ab |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

15. Conclusiones

En general respecto los resultados obtenidos, se deduce que el proyecto cumple con los objetivos y especificaciones básicas marcadas al inicio de este. A nivel de producto el dispositivo adquiere, procesa y permite la visualización a tiempo real de la información que interesa al usuario en los dos tipos de visualizaciones especificadas.

Por otro lado se han adquirido todos los conocimientos necesarios por parte del alumno en todos los niveles de aprendizaje. A parte de los conocimientos relativos a la organización, responsabilidad, gestión del tiempo, gestión de recursos. El alumno se ha desenvuelto con mayor soltura frente a problemas de programación en software y de implementación en hardware, la realización de este proyecto ha permitido encarar de forma más sencilla retos previamente vistos como complicados. Se valora muy positivamente la posibilidad de respecto los temas tratados y aprendidos durante la realización de este proyecto, tener la capacidad de ampliar horizontes y visualizar proyectos más ambiciosos.

Respecto a la implementación real del prototipo, se ha podido implementar la mayoría de los componentes sin la necesidad de disponer de un gran capital económico, de la misma forma se ha podido utilizar todos los programas de diseño y simulación permitiendo a cualquier persona diseñar e implementar un proyecto de estas dimensiones.

16. Memoria económica

16.1. Coste de los componentes

Listado de materiales:

| Material | Coste | Material | Coste |
|-------------|--------|-----------------|---------|
| PIC | 8,00 € | 24CL256 | 1,35 € |
| Pines LCD | 0,57 € | GL7516 | 1,20 € |
| Regleta Al. | 0,58 € | PC1602A | 11,90 € |
| Tornillos | 0,20 € | Placa | 5,90 € |
| RES 18K | 0,05 € | Porta pila Al. | 0,96 € |
| RES 4,7K | 0,05 € | Porta pila lit. | 1,00 € |
| HC-06 | 9,60 € | Pila litio | 0,30 € |
| A1106EUAT | 1,00 € | Potenciómetro | 0,28 € |
| 1N414 | 0,17 € | RES 1k | 0,05 € |
| LM35 | 1,20 € | RES 10k | 0,05 € |
| P6004 | 2,30 € | LED | 0,25 € |
| DS1307 | 3,25 € | | |
| Total | | 50,21 € | |

Listado e material adicional:

| Material Adicional | coste |
|--------------------|---------|
| Cableado | 3,35 € |
| Multímetro | 12,31 € |
| cableado | 0,23 € |
| protoboard | 17,00 € |
| soldador | 25,00 € |
| Total | 57,89 € |

16.2. Coste del prototipo

Para realizar el cálculo de cada situación se ha tenido en cuenta que un ingeniero junior recién licenciado o digamos con menos experiencia tendría un sueldo de unos 30€ la hora en lo que concierne a actividades e diseño y de 10€ al resto de actividades.

Para el cálculo del importe de los gastos generales y de los equipos, se ha supuesto que el ingeniero en cuestión es un autónomo con unos gastos asociados al alquiler, agua, luz además de los equipos utilizados para la implementación del dispositivo.

| | nº horas | €/h | € |
|---|----------|---------|-------------|
| Importe por el tiempo dedicado al diseño del software | 60 | 30,00 € | 1.800,00 € |
| Importe el tiempo dedicado a la simulación | 40 | 10,00 € | 400,00 € |
| Importe por el tiempo dedicado al diseño del hardware | 60 | 30,00 € | 1.800,00 € |
| Importe el tiempo dedicado al montaje del hardware | 40 | 10,00 € | 400,00 € |
| Importe el tiempo dedicado a la depuración del software | 100 | 10,00 € | 1.000,00 € |
| Importe el tiempo dedicado a pruebas funcionales | 100 | 10,00 € | 1.000,00 € |
| Importe el tiempo dedicado a la confección de documentación | 200 | 10,00 € | 2.000,00 € |
| Importe de los materiales | | | 26,97 € |
| Importe de los gastos generales y amortización de equipos | | | 757,89 € |
| IVA | | 21% | |
| Total | | | 11.113,68 € |

16.3. Coste de una producción seriada

Para el cálculo de los importes de montaje y pruebas funcionales, representa que son las operaciones realizadas por un operario de línea que repasa el montaje realizado por una producción seriada y de la misma manera para las pruebas funcionales, tardado media y un cuarto de hora para cada operación.

Sobre el precio requerido para amortizar la unidad, se ha de tener en cuenta que la selección de los componentes en algunos casos son de un coste demasiado elevado debido a la simple adaptación al material disponible ya sea por reaprovecharlo o por ser el único disponible. Por otro lado hay componentes de los que se podría reducir costes como el mismo microcontrolador ya que en un principio no se contaba con el número exacto de puertos a utilizar.

| | nº horas | €/h | € |
|--|----------|---------|--------|
| Importe el tiempo dedicado al montaje del hardware | 0,5 | 10,00 € | 5,00 € |
| Importe el tiempo dedicado a pruebas funcionales | 0,25 | 10,00 € | 2,50 € |

| | |
|---|---------|
| Importe del material de la unidad | 50,21 € |
| IVA | 21% |
| Amortización sobre una unidad de producción del coste total | 69,83 € |

Bibliografía

- [1]. Eduardo García Breijo. *Compilador_C_CCS_y_simulador_PROTEUS_para_microcontroladores* [Consultado Septiembre 2017]
- [2]. Cadsoft computer. *Eagle_Manual*. [Consultado Febrero 2018]
- [3]. Enrique Vicente Bonet Estevan. *Lenguaje_C*. [Consultado Septiembre 2017]
- [4]. MIT. *AppInventorTutorials*. [Consultado Febrero 2018]
- [5]. Manual Pick it 3. [Consultado Febrero 2018] Disponible en:
<http://ww1.microchip.com/downloads/en/DeviceDoc/51795B.pdf>
- [6]. 18f4550 data sheet. [Consultado Noviembre 2017] Disponible en:
<http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>
- [7]. LDR data sheet. [Consultado Noviembre 2017] Disponible en:
<http://kennarar.vma.is/thor/v2011/vgr402/ldr.pdf>
- [8]. Sensor hall data sheet. [Consultado Noviembre 2017] Disponible en:
<http://www.ti.com/lit/ds/symlink/drv5023.pdf>
- [9]. LM35 data sheet. [Consultado Noviembre 2017] Disponible en:
<http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [10]. Ds1307 data sheet. [Consultado Noviembre 2017] Disponible en:
<https://www.sparkfun.com/datasheets/Components/DS1307.pdf>
- [11]. LCD data sheet. [Consultado Noviembre 2017] Disponible en:
<https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>
- [12]. EEPROM 24LC256. [Consultado Noviembre 2017] Disponible en:
<https://www.google.es/search?rlz=1C1GIWA>
- [13]. Shopify webside. [Consultado Marzo 2018] Disponible en:
<https://upcomponent.myshopify.com/admin/auth/login>
- [14]. Ward van Wanrooij. *The C book*. Segunda edición originariamente publicado el 1991 por Addison Wesley. [Consultado Septiembre 2017]
- [15]. Maurizio di Paolo Emilio. *Data acquisition*. A Reference For DAQ And Analog & Digital Signal Conditioning Primera edición. Originalmente publicado Marzo 2013. [Consultado Octubre 2017]
- [16]. SRD data sheet. [Consultado Octubre 2017] Disponible en:
<http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf>
- [17]. Serial port expander data sheet [Consultado Marzo 2018] Disponible en:
https://www.atlascientific.com/files/datasheets/circuit/serial_port_expander_datasheet.pdf