

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL
Y AUTOMÁTICA

TRABAJO FIN DE GRADO

AUTONIVELACIÓN DE MESA CON SENSORES ULTRASÓNICOS. DISEÑO Y SIMULACIÓN

Alumna: Vidaurre, Gutiérrez, Famara

Directora: Otaegi, Aizpeolea, Aloña

Curso: 2020-2021

Fecha: 4, noviembre, 2020

RESUMEN

En el sector de la hostelería, la presencia de los pavimentos irregulares presenta un problema de estabilidad. Debido a ello, las mesas sufren un desequilibrio. Para solucionar este problema, en este proyecto, se ha diseñado el control de la mesa llegando a la automatización del nivelado con un software libre como es el Arduino y simulado con Proteus Design Suite. Debido a que se tuvo que controlar varios elementos electrónicos de diferentes características (motor, actuadores lineales, sensores...), se ha realizado el diseño del programa en el que se trata la señal recibida del sensor para activar el mecanismo y finalmente nivelar la mesa.

El primer paso ha sido el planteamiento del diagrama de flujo para observar en una idea general la programación.

En segundo lugar, se ha planteado el organigrama y el diagrama de bloques, concretando de esta forma las necesidades del proyecto y visualizando los elementos que serán necesarios para su desarrollo.

Una vez se han establecidos los criterios y los elementos tanto mecánicos como electrónicos, se ha realizado un estudio teórico de estos mecanismos. De esta forma, se han valorado las características de cada elemento y se ha elegido de forma justificada cada uno de ellos, llevando a cabo el control electrónico mediante la placa de desarrollo Arduino Mega; el sistema de actuación con disposición de cuatro motores, cuatro sensores de distancia ultrasonidos y cuatro actuadores lineales; y la alimentación mediante una batería recargable.

En tercer lugar se ha realizado el conexionado de cada elemento, llevando a cabo la programación en Arduino y la posterior simulación en Proteus comprobando el correcto funcionamiento y la compatibilidad de elementos.

De esta forma, se ha llegado a la autonivelación procesando las señales de los dispositivos, y se ha solucionado el desnivelado.

Palabras clave: Automatización, control, autonivelación

LABURPENA

Ostalaritzaren sektorean, zoladura irregularren presentziak egonkortasun-arazoak dakartza. Hori dela eta, mahaiak desorekatu daitezke. Arazo hori konpontzeko, proiektu honetan mahaiaren kontrola diseinatu da, Arduino software librez lagunduta eta Proteus Design Suite bitartez simulatuta. Proiektuaren diseinuan zehar hainbat eragingailu desberdin kontrolatu behar izan dira (besteak beste, motorrak, eragingailu linealak eta sentsoak). Gauzak horrela, sentsoetik hartutako seinalea tratatzen duen programaren diseinua egin da mekanismoa aktibatzeke eta, azkenik, mahaia nibelatzeko.

Lehenengo urratsa fluxu-diagrama planteatzea izan da, programazio orokorraren antolaketa egiteko.

Bigarrenik, organigrama eta blokeen diagrama planteatu dira, horrela, proiektuaren beharrak zehaztu eta haren garapenerako beharrezkoak diren elementuak irudikatzeke.

Irizpideak eta elementu mekaniko eta elektronikoak ezarri ostean, horien analisi teorikoa egin da. Horrela, elementu bakoitzaren ezaugarriak balioetsi dira eta modu justifikatuan hautatu. Kontrol elektronikoa Arduino Mega garapen txartelaren bidez egin da; sistema eragileak lau motor, lau ultrasoinuzko distantzia-sentsore eta lau eragile lineal ditu; elikadura, berriz kargatzeko bateria baten bitartez egiten da.

Hirugarrenik, elementu bakoitzaren konexioa egin da; programazioa Arduinoz eta ondorengo simulazioa Proteus Design Suiten bitartez, funtzionamendua zuzena dela eta elementuen arteko bateragarritasuna egiaztatzeke.

Gauzak horrela, auto-nibelazioa lortu da, gailuen seinaleak prozesatuz eta desoreka konpondu da.

Gako-hitzak: automatizazio, kontrol, autonibelazio

ABSTRACT

In the hotel and catering sector, the presence of uneven pavements, causes a stability problem. Due to that, the tables have an imbalance issue. To solve this situation, through this project, the control of the table has been designed, reaching to the automation of the levelling with free softwares such as Arduino, and it is simulated with Proteus Design Suite. Since several electronic elements of different characteristics had to be controlled (motor, linear actuators, sensors...), a design of the program has been carried out, in which the signal that is received from the sensor is treated to activate the mechanism and finally, level the table.

The first step has been the approach of the flow chart to observe the programming in a general idea.

Secondly, the organization chart and the block diagram have been proposed, specifying in this way, the needs of the project and visualizing the elements that will be necessary for its development.

Once the criteria and the mechanical and electronic elements have been established, a theoretical study of these mechanisms has been carried out. In this way, the characteristics of each element have been assessed, and each one of them has been chosen in a justified way, carrying out the electronic control through the Arduino Mega development board; the actuation system with a layout of four motors, four ultrasonic distance sensors and four linear actuators; and the power supply through a rechargeable battery.

Thirdly, the connection of each element has been made, carrying out the programming in Arduino and the subsequent simulation in Proteus, checking the correct operation and compatibility of the elements.

In this way, self-levelling has been achieved by processing the signals from the devices, and the unevenness has been solved.

Keywords: Automation, control, self-levelling

ÍNDICE DE CONTENIDO

| | |
|--|----|
| 1. INTRODUCCION | 1 |
| 2. OBJETIVOS..... | 3 |
| 3. REQUISITOS DE PARTIDA..... | 4 |
| 4. FUNDAMENTOS TEÓRICOS | 8 |
| 4.1 Microcontrolador | 8 |
| 4.3 Motor..... | 17 |
| 4.4 Sensor..... | 19 |
| 4.5 Pulsador..... | 23 |
| 4.6 Batería | 24 |
| 5. SELECCIÓN JUSTIFICADA DE COMPONENTES..... | 26 |
| 5.1 Microcontrolador | 26 |
| 5.2 Actuador..... | 27 |
| 5.3 Motor..... | 28 |
| 5.4 Sensor..... | 30 |
| 5.5 Activador (pulsador) | 33 |
| 5.6 Batería | 34 |
| 6. DISEÑO | 37 |
| 6.1 Conexionado de los elementos seleccionados | 37 |
| 6.1.1 <i>Arduino</i> | 37 |

| | |
|--|----|
| 6.1.2 <i>Sensor</i> | 43 |
| 6.1.3 <i>Motor</i> | 45 |
| 6.1.4 <i>Leds para el aviso de errores</i> | 48 |
| 6.2 Diseño gráfico | 49 |
| 6.3 Diagrama de flujo..... | 51 |
| 6.4 Programación | 57 |
| 7. SIMULACIÓN..... | 60 |
| 8. PRESUPUESTO | 64 |
| 8.1. Coste en recursos humanos | 64 |
| 8.2. Coste de software asociado | 65 |
| 8.3. Coste de material..... | 66 |
| 9. DIAGRAMA DE GANTT | 67 |
| 10. CONCLUSIONES | 72 |
| 11. BIBLIOGRAFÍA | 74 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1- Mesa con irregularidad del suelo | 4 |
| Figura 2- Organigrama abstracto del autoajuste del sistema | 5 |
| Figura 3- Diagrama de bloques global..... | 6 |
| Figura 4- Mesa acero como punto de partida para la autonivelación | 7 |
| Figura 5- Pinout Arduino Uno | 10 |
| Figura 6- Pinout Arduino Mega..... | 12 |
| Figura 7- Partes motor DC..... | 18 |
| Figura 8- Representación transductor de distancia | 19 |
| Figura 9 - Esquema sensores proximidad | 20 |
| Figura 10- Actuador lineal 12V | 28 |
| Figura 11- Gráfica carga-intensidad motor..... | 29 |
| Figura 12- Gráfica carga-velocidad motor | 29 |
| Figura 13- Sensor HC-SR04..... | 31 |
| Figura 14-Funcionamiento HC-SR04..... | 32 |
| Figura 15- Pulsador seleccionado | 33 |
| Figura 16- Batería recargable | 34 |
| Figura 17- Tensión salida step-down 10kHz | 39 |
| Figura 18- Tensión salida step-down 1kHz | 39 |
| Figura 19- Simulación step-down Proteus..... | 40 |

| | |
|---|----|
| Figura 20- Conexionado regulador tensión LM7809CT | 40 |
| Figura 21- Característica regulador de tensión LM7809 | 41 |
| Figura 22- Disipador térmico HS115-ND | 42 |
| Figura 23- Circuito Pull Down pulsador | 43 |
| Figura 24- Conexión sensores ampliado..... | 44 |
| Figura 25- Conexionado sensores..... | 44 |
| Figura 26- Esquema general controlador motor y arduino | 46 |
| Figura 27- Conexionado del driver motor uno del actuador..... | 46 |
| Figura 28- Pines efecto hall | 47 |
| Figura 29- Conexionado sensor de tensión FZ0430..... | 49 |
| Figura 30- Prototipo mesa | 50 |
| Figura 31- Vista superior prototipo mesa | 50 |
| Figura 32- Vista inferior prototipo mesa | 51 |
| Figura 33- Diagrama de flujo | 56 |
| Figura 34- Simulación pulsador y led..... | 60 |
| Figura 35- Simulación lectura sensor ultrasonido 140cm | 61 |
| Figura 36- Simulación lectura sensor ultrasonido 379cm | 61 |
| Figura 37- Simulación motor con driver | 62 |
| Figura 38- Simulación proyecto | 63 |
| Figura 39- Diagrama de Gantt | 71 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1- Descripción de los pines de Arduino Uno..... | 10 |
| Tabla 2- Descripción pines Arduino Mega..... | 15 |
| Tabla 3- Comparativa actuadores | 17 |
| Tabla 4- Comparativa Brushless-Brushed | 18 |
| Tabla 5- Comparativa sensores..... | 22 |
| Tabla 6- Material a detectar | 23 |
| Tabla 7- Comparativa baterías..... | 25 |
| Tabla 8- Comparación Arduino Uno y Mega | 26 |
| Tabla 9- Características motor..... | 28 |
| Tabla 10- Características sensor HC-SR04 | 31 |
| Tabla 11- Pines sensor HC- SR04 | 32 |
| Tabla 12- Voltaje e intensidades elementos | 37 |
| Tabla 13- Características HS115-ND | 41 |
| Tabla 14- Pines empleados en sensores..... | 44 |
| Tabla 15- Referencia pines motor conexión estándar..... | 45 |
| Tabla 16- Pines conexión motor 1, 2, 3 y 4..... | 47 |
| Tabla 17- Referencia pines conexión efecto hall..... | 47 |
| Tabla 21- Error por aproximación de tiempo | 59 |
| Tabla 22- Presupuesto recursos humanos..... | 64 |

| | |
|---|----|
| Tabla 23- Coste de software asociado | 65 |
| Tabla 24- Coste elemento ejecución software | 65 |
| Tabla 25- Coste de material..... | 66 |
| Tabla 26- Coste total..... | 66 |

1. INTRODUCCION

El presente proyecto surge con el objetivo de solucionar el problema ocasionado por el pavimento irregular en hostelería. Éste supone una desnivelación de las mesas en las que se da un servicio de terraza, pudiendo ocasionar la caída de las consumiciones. La idea surgió tras la observación de personas que regentan locales de hostelería próximas a mi entorno familiar.

La solución establecida actualmente es la utilización de cuñas de diferentes materiales en una de las extremidades de la mesa, así como el movimiento de ésta a una zona con menor inclinación logrando de esta forma su estabilización.

Como consecuencia, surgió la idea de diseñar un sistema de control de automatización de la estabilización de una mesa sobre suelo irregular. Para el desarrollo de esta idea se tienen en cuenta conocimientos adquiridos de control, programación e instrumentación.

El documento se estructura de la siguiente forma:

- En primer lugar se establecen los objetivos iniciales estableciendo la idea general del proyecto para el posterior desarrollo.
- En segundo lugar, se realiza un diseño de abstracción de los elementos empleados para poder profundizar más en cada uno de los componentes.
- En tercer lugar, se determinan los diferentes mecanismos, como los motores que permitirán el movimiento, los sensores con los que se realizará la lectura de la distancia y el microprocesador a emplear para realizar la programación. Se llevará a cabo la selección y justificación de los elementos citados, así como un diseño detallado tras haber estudiado las posibilidades que existen en cuanto a los distintos componentes en función de las características deseadas.
- En cuarto lugar, se realiza el diagrama de flujo para proceder posteriormente a la programación del microcontrolador procesando de esta forma las señales recibidas y decretando las señales de salida y simulación de los componentes, finalizando con ello la realización del proyecto. Además, también se llevarán a cabo simulaciones para la

observación del funcionamiento y la programación de los elementos que componen el proyecto.

- Finalmente, se introduce el diagrama de Gantt pudiendo observar las tareas realizadas en la evolución del tiempo. Tras ello, se realizan las conclusiones analizando los objetivos por cumplir y las complicaciones en la realización del proyecto.

2. OBJETIVOS

El objetivo principal de este proyecto es el diseño, desarrollo y programación de un sistema de control capaz de gobernar cualquier mesa equipada con cuatro motores, cuatro sensores y un sistema mecánico de ascenso/descenso, permitiendo así su equilibrio.

Dicho objetivo principal puede dividirse en objetivos secundarios a corto plazo y realistas, pequeñas aportaciones que permiten llegar al objetivo principal, presentadas a continuación:

- El diseño y desarrollo de un control eficiente y eficaz para la nivelación automática de la mesa.
- Realización de un organigrama para el desarrollo principal de la estructura del proyecto.
- Búsqueda de los diferentes elementos para la elaboración del proyecto como sensores, actuadores, microcontrolador y alimentación con la posterior elección de los más acertados.
- Visualización de los posibles errores para la correcta programación, realizando simulaciones concretas con Proteus Design Suite.

Como propósito personal se desea hacer uso de los conocimientos adquiridos durante el grado en las asignaturas relacionadas con la automatización, control, instrumentación y circuitos eléctricos. También se quiere hacer frente a las dificultades planteadas durante el trabajo de hostelería, facilitando el trabajo y aportando comodidad tanto al cliente como al trabajador, siendo aplicable a otros ámbitos laborales.

3. REQUISITOS DE PARTIDA

La estabilización de la mesa se lleva a cabo mediante el estudio de los sistemas electrónicos necesarios. De esta forma, cumpliendo los objetivos, se llega a los resultados deseados para la correcta nivelación. Para ello, son necesarios conocimientos sobre dispositivos electrónicos, automatización y control.

La Figura 1 es una representación gráfica de la dificultad del equilibrio que presenta la mesa ante las irregularidades del pavimento, observando como alguna de las patas que componen la mesa está al aire, con lo que la estabilidad peligra.

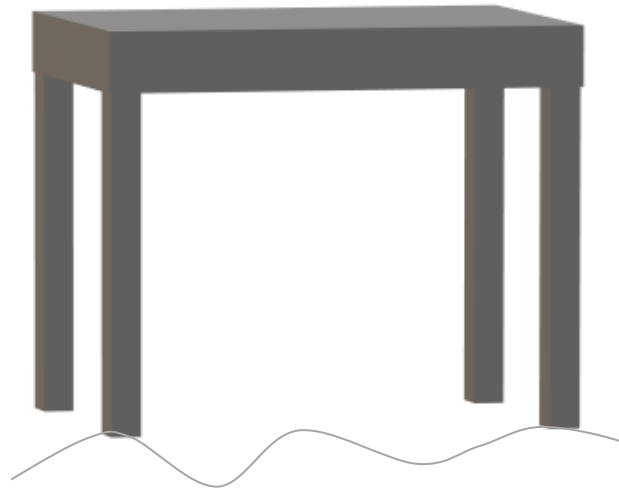


Figura 1- Mesa con irregularidad del suelo

Para la realización del proyecto se presenta un organigrama en la Figura 2, estableciendo el objetivo técnico a alcanzar tras haber observado los problemas que supone el desnivel de la mesa debido a la irregularidad del pavimento, y los elementos necesarios tanto electrónicos como mecánicos para la posible solución.

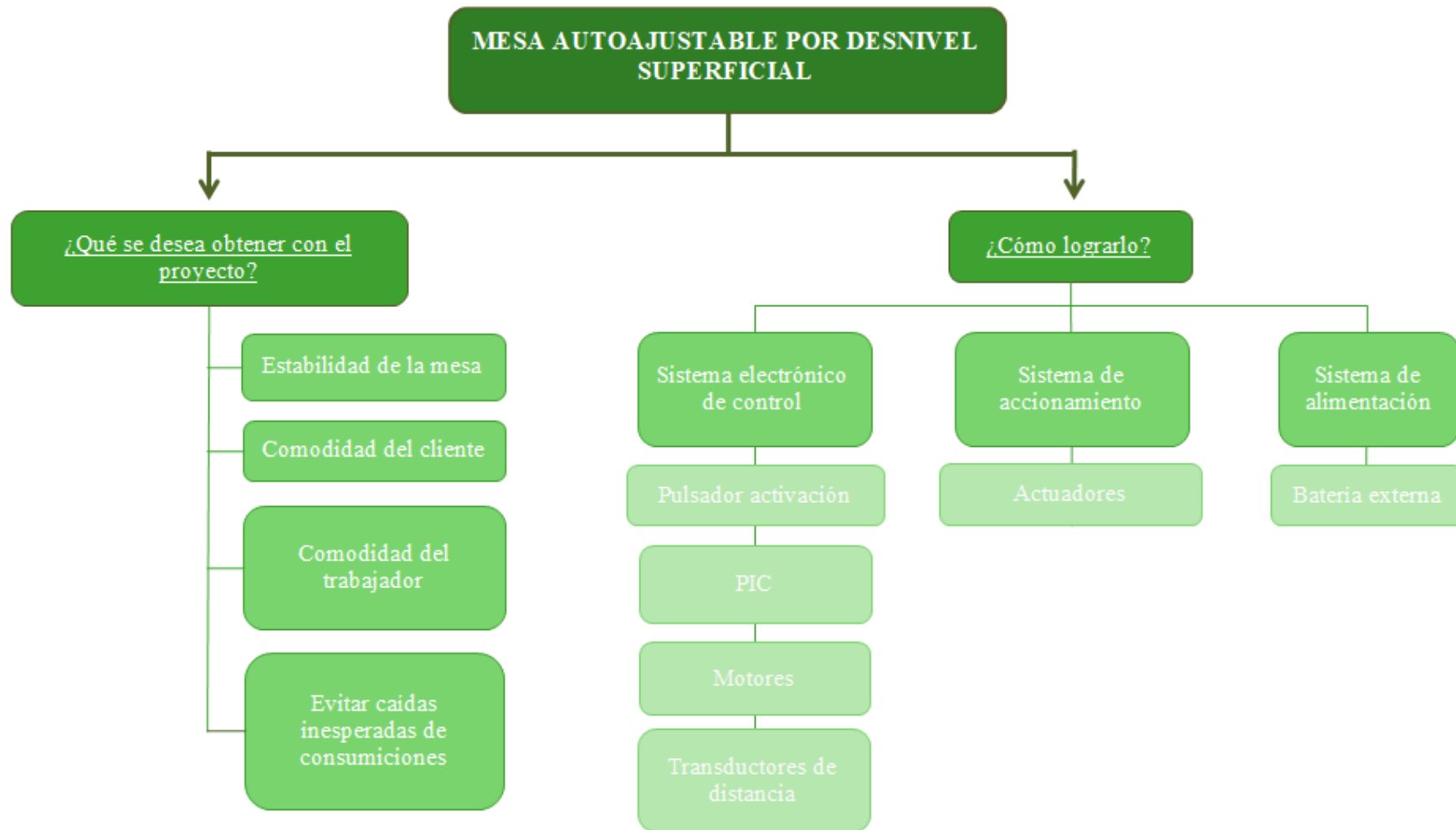


Figura 2- Organigrama abstracto del autoajuste del sistema

El organigrama funcional está subdividido en dos bifurcaciones principales:

- La primera bifurcación hace referencia a la finalidad que se desea conseguir con el proyecto, enfocándose en la facilitación laboral que se puede obtener en el campo de hostelería, tanto dirigido a la comodidad de los clientes como de los trabajadores.

- La segunda bifurcación se centra en cómo se logrará la finalidad. Para ello se presentan tres sistemas: El control de la mesa, el accionamiento que permitirá la estabilización y la alimentación necesaria. Las tres ramificaciones a su vez contienen un subnivel con los componentes que conforman estos tres sistemas.

El sistema electrónico de control se realizará con un microcontrolador PIC integrado en una placa. Además, la activación del nivelado se ejecutará mediante un pulsador, la lectura de la altura, mediante los sensores; y la respuesta a la lectura será efectuada por los motores. El sistema de accionamiento permitirá proceder al movimiento, tras la recepción de la señal procesada por el controlador. Este movimiento será realizado por los actuadores.

El sistema de alimentación permitirá obtener la energía necesaria para el funcionamiento de cada sistema. La energía eléctrica será suministrada por una batería.

Tras el análisis del organigrama, se presenta un diagrama de bloques global en la Figura 3 en el que se observan los principales elementos que componen la estructura para el funcionamiento. La inestabilidad es detectada por el sensor tras pulsar el activador. La lectura del sensor la realiza un microcontrolador, creando de esta manera la señal que recibe el motor para la realización del movimiento de las patas para la nivelación de altura.

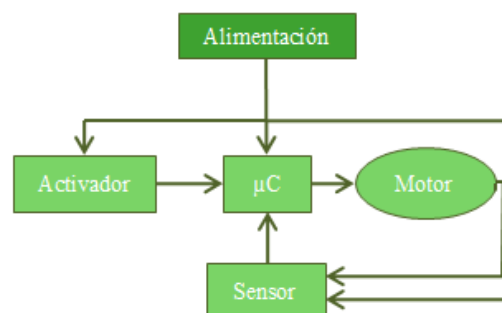


Figura 3- Diagrama de bloques global

Para realizar la programación y solucionar el problema mencionado de estabilización, se tiene como punto de partida la mesa con la que se trabaja en dicho establecimiento, teniendo en cuenta sus características, determinantes para la elección de los elementos que constituirán el proyecto.

- Medidas: Alto: 720 mm; Ancho: 600 mm; Profundo: 600 mm
- Altura de las patas: 681mm.
- Peso: 8,5 kg peso
- Cuatro patas, cuatro elementos a regular



Figura 4- Mesa acero como punto de partida para la autonivelación

El peso será una característica determinante para la selección de los elementos de elevación. Por ello, se estima que en torno a una mesa como máximo por norma general hay 10 personas. El recipiente más pesado que se encuentra en el establecimiento pesa 880 gr, cuya suma con el peso del líquido será de 1,4 kg. Por ello, se estima que el peso máximo que tendrá la superficie será de 36,5 kg, teniendo en cuenta 20 recipientes llenos más el peso de la mesa.

4. FUNDAMENTOS TEÓRICOS

En este apartado se presentan las características de cada elemento que formará parte del proyecto. Entre ellos se encuentran el microcontrolador, el actuador, el motor, el sensor, el activador y la batería.

4.1 Microcontrolador

Es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora.

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales seis pueden ser utilizadas como salidas PWM (modulación por ancho de pulsos) y otras seis son entradas analógicas. Además, incluye un reloj de 16 MHz, un conector USB, un conector de alimentación, un conector ICSP (In circuit Serial Programming) y un botón de reseteo. El conector ICSP sirve para programar el BootLoader del microprocesador y así poder cargar los programas directamente en el micro sin tener necesidad de programas externos.

La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, únicamente se debe conectar la placa a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

- *Características Técnicas:*

- Micro controlador: ATmega328
- Voltaje: 5V
- Voltaje de entrada recomendado: 7-12V
- Voltaje límite de entrada: 6-20V
- Digital I/O Pins: 14, de los cuales 6 son salida PWM
- Entradas Analógicas: 6
- DC Current per I/O Pin: 40 mA

- DC Current parar 3.3V Pin: 50 mA
- Memoria Flash: 32 kB de los cuales 0.5 kB son utilizados para el arranque.
- SRAM: 2 kB
- EEPROM: 1 kB
- Velocidad de reloj: 16 MHz

| Nombre del pin | Descripción |
|---------------------|---|
| V_{in} | Tensión de alimentación de arduino cuando se va a utilizar una fuente de alimentación externa |
| 3.3V | Fuente de alimentación de 3.3V. Corriente máxima de 50mA. |
| 5V | Alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V. |
| Reset | Sirve para resetear el microcontrolador |
| A0 - A5 | 6 entradas analógicas que varían entre 0V y 5V |
| Digital pins 0 - 13 | Pines digitales que pueden configurarse como entradas o salidas |
| 0(Rx), 1(Tx) | Estos pines no se pueden utilizar como e/s digitales si se están utilizando comunicación serie (por ejemplo Serial.begin). |
| 2, 3 | Interrupciones externas |
| 3, 5 6, 9, 11 | Salidas PWM de 8 bits |
| 13 | Único pin que tiene un led conectado en serie a la entrada, si se quiere testear algo mediante el brillo de un led, se puede utilizar este pin |

| | |
|--|--|
| 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) | Comunicación SPI |
| A4 (SDA), A5 (SCA) | Comunicación TWI |
| AREF | Mantiene una tensión de referencia. Es útil para realizar conversiones analógicas a digitales. |

Tabla 1- Descripción de los pines de Arduino Uno

Pinout

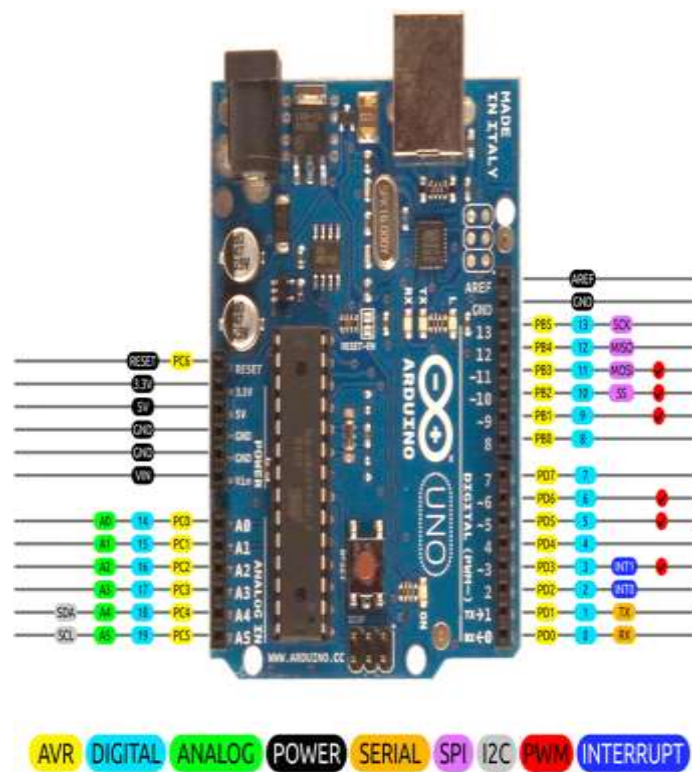


Figura 5- Pinout Arduino Uno

Arduino Mega es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S),

analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring.

El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino. Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC. Además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Características técnicas

- ✓ Microcontrolador: ATmega2560
- ✓ Voltaje Operativo: 5V
- ✓ Voltaje de Entrada: 7-12V
- ✓ Voltaje de Entrada (límites): 6-20V
- ✓ Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- ✓ Pines análogos de entrada: 16
- ✓ Corriente DC por cada Pin Entrada/Salida: 40 mA
- ✓ Corriente DC entregada en el Pin 3.3V: 50 mA
- ✓ Memoria Flash: 256 KB (8KB usados por el bootloader)
- ✓ SRAM: 8KB
- ✓ EEPROM: 4KB
- ✓ Velocidad de reloj: 16 MHz

| Nombre del pin | Descripción |
|------------------|--|
| V_{in} | Tensión de alimentación de arduino cuando se va a utilizar una fuente de alimentación externa |
| 3.3V | Fuente de alimentación de 3.3V. Corriente máxima de 50mA. |
| 5V | Fuente de alimentación de 5V |
| Reset | Sirve para resetear el microcontrolador |
| 0(Rx), 1(Tx) | Serie 0. Usados para recibir (RX) y transmitir (TX) datos a través de puerto serie TTL. Estos pines están conectados a los pines correspondientes del chip FTDI USB-to-TTL |
| 19 (Rx), 18 (Tx) | Serie 1. Usados para recibir (RX) transmitir (TX) datos a través de puerto serie TTL |
| 17 (Rx), 16 (Tx) | Serie 2. Usados para recibir (RX) transmitir (TX) datos a través de puerto serie TTL |
| 15 (Rx), 14 (Tx) | Serie 3. Usados para recibir (RX) transmitir (TX) datos a través de puerto serie TTL |
| 2 | Interrupción externa (interrupción 0). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |

| | |
|---------|--|
| 3 | Interrupción externa (interrupción 1). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |
| 18 | Interrupción externa (interrupción 5). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |
| 19 | Interrupción externa (interrupción 4). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |
| 20 | Interrupción externa (interrupción 3). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |
| 21 | Interrupción externa (interrupción 2). Este pin se puede configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH (5V) o viceversa), o en cambios de valor. Ver la función attachinterrupt() para más detalles. |
| 0 al 13 | Salidas PWM de 8 bits de resolución (valores de 0 a 255) |

| | |
|---|---|
| 13 | Único pin que tiene un led conectado en serie a la entrada, si se quiere testear algo mediante el brillo de un led, se puede utilizar este pin. |
| SPI: 50 (SS), 51 (MOSI), 52 (MISO), 53 (SCK) | Proporcionan comunicación SPI, usando librerías SPI |
| A0 - A15 | 16 entradas analógicas con una resolución de 10 bits (1024 valores). Por defecto se mide desde 0V a 5V, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función analogReference(). |
| 20 (SDA) y 21 (SCL) | Soporte para el protocolo de comunicación I2C (TWI) usando la librería Wire. |
| AREF | Voltaje de referencia para las entradas analógicas |

Tabla 2- Descripción pines Arduino Mega

4.2 Actuador

Un actuador es un dispositivo mecánico cuya función es proporcionar fuerza para mover o “actuar” sobre otro dispositivo mecánico.

Con los avances en tecnología y el paso a la era de la automatización, el funcionamiento de los dispositivos a partir de la fuerza de trabajo humana se han quedado atrás para dar lugar a la automatización de estos elementos mediante actuadores.

La fuerza que provoca un actuador proviene de tres fuentes posibles: presión neumática, hidráulica o fuerza motriz eléctrica. Por lo que, dependiendo del origen de la fuerza, el actuador se denominará neumático, hidráulico o eléctrico. A continuación se procede a la información de estos tres tipos de actuadores:

- Actuadores electrónicos

Son accionados por medio de corrientes eléctricas. Algunos consumen mucha energía, por lo que para este tipo de casos se utilizan controladores.

Su estructura es más simple en comparación con la de los actuadores hidráulicos y neumáticos, ya que solo requieren energía eléctrica como fuente de energía.

Se suelen utilizar en los robots o para controlar tareas cotidianas.

- Actuadores neumáticos

Convierten la energía del aire comprimido en trabajo mecánico. Los modelos neumáticos son simples posicionamientos. El trabajo por un actuador de estas características puede ser lineal o rotativo.

- Actuadores hidráulicos

Utilizan líquidos a presión para generar fuerza mecánica. Uno de los líquidos más utilizados es el aceite. Se emplean cuando lo que se necesita es potencia y en máquinas de gran peso y tamaño. Requieren demasiado suministro de energía, así como de un mantenimiento periódico.

A continuación se presenta una tabla comparativa con sus correspondientes características:

| | Eléctrico | Neumático | Hidráulico |
|------------------|---------------------|------------------|-------------------|
| Energía | Corriente eléctrica | Aire a presión | Aceite |
| Carga | Baja | Media | Elevada |
| Velocidad | Más lentos | Rápidos | Rápidos |
| Precisión | Buena | Buena | Buena |
| Precio | Bajo | Bajo | Elevado |

Tabla 3- Comparativa actuadores

Según el tipo de fuerza originada se distribuyen en:

- Lineales: Generan una fuerza en línea recta, como lo haría un pistón. Permiten elevar altas cargas y su recorrido puede ser largo. En cambio, son mecanismos más lentos de movimiento que otros como por ejemplo el actuador magnético.

- Rotatorios: Generan una fuerza rotatoria, como lo haría un motor eléctrico.

Debido a que el movimiento debe ser lineal para la compensación de la mesa y que el proyecto no requiere una velocidad elevada de movimiento para el correcto funcionamiento se determina que los actuadores lineales serán los elementos que ejerzan el movimiento.

4.3 Motor

Es la parte de una máquina capaz de transformar algún tipo de energía (eléctrica, de combustibles fósiles, etc.), en energía mecánica capaz de realizar un trabajo.

Los actuadores lineales tienen integrados un motor DC con escobilla por lo que convierte la energía eléctrica en mecánica. Se compone de dos partes: el estator y el rotor. El estator es la parte mecánica del motor donde están los polos del imán. El rotor

es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas.

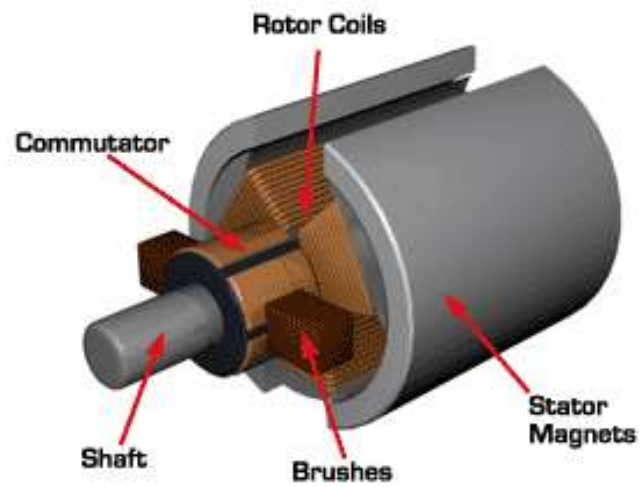


Figura 7- Partes motor DC

La tabla siguiente muestra las diferencias principales entre los tipos de motores DC:

| | DC Brushless | DC Brushed |
|------------------------------|--------------|---------------|
| Vida útil | Mayor | Menor |
| Mantenimiento | Menor | Mayor |
| Precio | Más caro | Más económico |
| Circuito control electrónico | Más complejo | Más simple |

Tabla 4- Comparativa Brushless-Brushed

4.4 Sensor

Un sensor es un dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno. En este caso, lo que se desea medir es una distancia. Por ello, se determina la utilización de sensores de proximidad, también llamados transductores de proximidad.

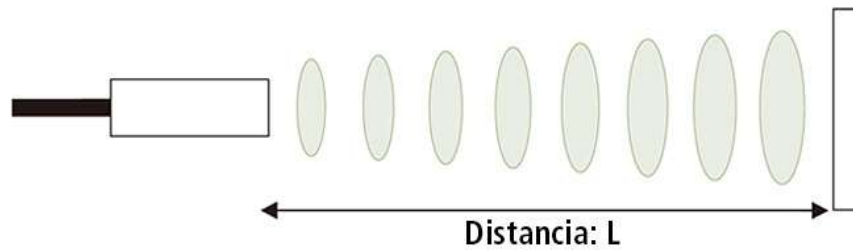


Figura 8- Representación transductor de distancia

Los sensores proporcionan una señal eléctrica según la variación de distancia que exista. Los sensores de proximidad permiten detectar la presencia de un objeto a una cierta distancia, dependiendo de la tecnología que utilicemos la distancia de detección será mayor o menor. En el siguiente esquema se observa una clasificación de los sensores de proximidad.



Figura 9 - Esquema sensores proximidad

Para valorar qué tipo es el más adecuado, se presenta una descripción de cada uno de los que no requieren contacto físico.

- *Sensores magnéticos:*

Reaccionan ante los campos magnéticos de imanes permanentes o electroimanes. El método más simple y usual de detectar un campo magnético es utilizar un interruptor reed. Dos laminillas flexibles de material magnético se unen por medio de un campo magnético externo, estableciendo un contacto eléctrico.

- *Sensores inductivos:*

Detectan la posición electrónicamente. Son capaces de detectar cualquier objeto metálico a una cierta distancia sin necesidad de contacto. En el caso de los sensores

magnéticos el elemento a detectar tenía que tener la fuente de magnetismo, en este caso de los inductivos el propio sensor el que emite una señal magnética.

- *Sensores capacitivos:*

Son detectores de posición electrónicos, capaces de detectar cualquier objeto a una cierta distancia sin necesidad de contacto. Si un material metálico o no, se introduce dentro de la zona activa frente al sensor, la capacidad C del condensador cambia de valor.

- *Sensores ópticos*

Utilizan medios ópticos y electrónicos para la detección de objetos utilizando luz roja o infrarroja. Los emisores de luz suelen ser diodos semiconductores (LEDs) y los elementos receptores son fotodiodos y fototransistores.

La luz infrarroja es invisible, en cambio, la luz roja tiene la ventaja que es visible. Además, la luz infrarroja es menos susceptible a las interferencias de luz ambiente.

Estos a su vez se subdividen en sensores de barrera, de retroreflexión y de reflexión directa. El primero consta de emisor y receptor separados en distintos dispositivos. En cambio, los dos últimos tienen el emisor y receptor en el mismo dispositivo, teniendo el de retroreflexión un reflector en la distancia.

- *Sensores ultrasónicos*

El funcionamiento está basado en la emisión y reflexión de ondas acústicas que emite el emisor, rebotan en el objeto y llegan hasta el receptor. Se mide y se evalúa el tiempo que tarda en desplazarse el sonido.

La tabla comparativa de estos sensores se muestra a continuación:

| | | Alcance | Material a detectar | Interferencia | Vida útil de contacto | Velocidad de conmutación |
|--------------------|--------------------------|-----------------------------|---|--|------------------------------|---|
| Magnético | | Máx. 100mm | Objetos imantados | Sensible a interferencia magnética cerca al sensor | 5 millones de ciclos | 2ms |
| Inductivo | | 0.8- 250mm | Objeto con componentes metálicos | Sensible a campos electromagnéticos | Mayor que los magnéticos | Conmutación típica: 0.1s-0.2ms y máxima de 0.05ms |
| Capacitivo | | 5-60mm | Objetos con constante dieléctrica > 1 | Sensibles a la suciedad | Muy elevada | 3.3ms |
| Óptico | Barrera | 100 m | Todo tipo, sin ser muy transparentes | Sensible a la suciedad | Elevada | De 50 a 0.1ms |
| | Retroreflexión | 10 m | Todo tipo, sin ser muy transparentes | Sensible a la suciedad | Elevada | De 100 a 1ms |
| | Reflexión directa | 50 mm a 2m | Todo tipo con un grado de reflectividad | Sensibles a la suciedad | Elevada | De 100 a 2 ms |
| Ultrasónico | | 50 mm- máxima de 10 m | Amplia gama de materiales | Cambios por temperatura y humedad | Muy larga | De 1s a 8ms |

Tabla 5- Comparativa sensores

También se debe analizar el tipo de material del que se compone el objeto a detectar y a qué distancia se dispondrá:

| Material | | Distancia | Tipo de sensor |
|----------------|--------------|-----------|----------------------|
| Sólido | Metálico | < 50 mm | Inductivo |
| | | > 50 mm | Óptico o ultrasónico |
| | No metálico | < 50 mm | Capacitivo |
| | | > 50 mm | Óptico o ultrasónico |
| | Magnético | < 50 mm | Magnético |
| | | > 50 mm | Óptico o ultrasónico |
| Líquido | Transparente | < 50 mm | Capacitivo |
| | | > 50 mm | Ultrasónico |
| | Opaco | < 50 mm | Capacitivo |
| | | > 50 mm | Óptico |

Tabla 6- Material a detectar

4.5 Pulsador

Un pulsador es un interruptor o switch cuya función es permitir o interrumpir el paso de la corriente eléctrica de manera momentánea, a diferencia de un switch común, un pulsador solo realiza su trabajo mientras lo tenga presionado, es decir sin enclavamiento. Existen pulsadores normalmente cerrados, cuyo circuito permanece cerrado en estado de reposo y se abre cuando se presiona; y normalmente abiertos, cuyo circuito está abierto en estado de reposo y se cierra cuándo se presiona.

4.6 Batería

Una batería eléctrica, también llamada pila o acumulador eléctrico, es un artefacto compuesto por celdas electroquímicas capaces de convertir la energía química en su interior en energía eléctrica, mediante la acumulación de corriente alterna. De esta manera, sirven para alimentar distintos circuitos eléctricos, dependiendo de su tamaño y potencia.

A continuación se realiza una clasificación atendiendo a su carga, existiendo dos tipos:

- *Batería primaria*

Su reacción electroquímica es irreversible, por ello después de que la batería se ha descargado no puede volver a cargarse.

- *Batería secundaria*

Su reacción electroquímica es reversible, por ello después de que la batería se ha descargado puede ser cargada mediante corriente continua de una fuente externa.

Según los elementos empleados en su fabricación se pueden distinguir baterías de:

- *Plomo-ácido*

- *Nickel-cadmio(NiCd)*

- *Nickel-hidrurometálico(Ni-Mh)*

- *Ion-Litio(Li-ion)*

- *Polímero-Litio(Li-poly)*

La comparación de las diferentes características de cada una de ellas es la siguiente:

| | Plomo | Ni-Cd | Ni-Mh | Li-ion | Li-Po |
|--|----------------|----------------|-----------------|------------------|------------------|
| Energía | 30-50 Wh/Kg | 48-80 Wh/kg | 60-120 Wh/kg | 110-160 Wh/kg | 100-130 Wh/kg |
| Tensión | 2 V | 1,25 V | 1,25 V | 3,16 V | 3,7 V |
| Número de cargas (Duración) | 1000 | 500 | 1000 | 4000 | 5000 |
| Tiempo de carga | 8-16h | 10-14h | 2-4h | 2-4h | 1-1,5h |
| Auto descarga | 5 % | 30% | 20% | 25% | 10% |

Tabla 7- Comparativa baterías

5. SELECCIÓN JUSTIFICADA DE COMPONENTES

En base a los conocimientos mostrados en la Figura 3 en el diagrama de bloques, y los conceptos básicos establecidos, se presentan los elementos empleados en el proyecto serán los siguientes:

- Microcontrolador
- Cuatro actuadores
- Cuatro motores
- Cuatro sensores
- Pulsador para la activación de la autonivelación
- Batería de carga del sistema.

A continuación se presentan las diferentes opciones y los motivos de la determinación de éstas.

5.1 Microcontrolador

Los PIC que se valoran para el desarrollo de este proyecto son el 16, 24 y los integrados en la placa Arduino Uno y Mega. La infinidad de opciones, librerías, información y componentes compatibles hacen que el PIC a seleccionar sea el integrado en la placa de Arduino.

A continuación se presentan las diferencias entre el Arduino Uno y Mega que se han analizado para la elección de uno de ellos.

| | Arduino Uno | Arduino Mega |
|----------------|--------------------|---------------------|
| Memoria Flash | 32 KB | 256 KB |
| Memoria SRAM | 2 KB | 8 KB |
| Memoria EEPROM | 1 KB | 4 KB |
| I/O digital | 14 (4 salidas PWM) | 54 (15 salidas PWM) |

Tabla 8- Comparación Arduino Uno y Mega

La primera característica a valorar es el número de pines digitales, no siendo suficiente el número de pines de Arduino Uno. Además, también se evalúa la memoria SRAM, la cual al ser 4 veces mayor ofrece más espacio para crear y manipular variables cuando se ejecuta.

5.2 Actuador

La elección del actuador depende de los siguientes factores:

- Carga a elevar: 8,5 kg de mesa, más los dispositivos y consumiciones. Como se ha establecido en requerimientos iniciales, será de en torno a 36,5 kg. Se establece una carga no superior a 50 kg, dejando un ligero margen en el peso.
- Velocidad de movimiento: Distancia que recorre el actuador durante un periodo de tiempo. No se precisa de un valor elevado, ya que la necesidad del proyecto no exige un movimiento rápido.
- Seguridad del dispositivo: Empleo de dos finales de carrera de seguridad, los cuales son dispositivos electrónicos, neumáticos o mecánicos situados al final del recorrido con el objetivo de que el actuador no se dañe cuando está en sus límites y se le manda una señal al motor para seguir en movimiento.
- Largo: Longitud del actuador sin tener en cuenta el recorrido.
- Recorrido: Longitud extensible del actuador.

Estos dos últimos términos están relacionados, ya que para hacer viable el proyecto el recorrido más el largo debe ser superior a la longitud de la pata (681 mm).

A continuación se especifican las características explicadas en el actuador seleccionado:

- Carga máxima: 1780 N, 181 kg
- Velocidad de movimiento: 24,892 mm/s con carga máxima
- Seguridad del dispositivo: Dos finales de carrera de seguridad eléctricos.
- Largo: 606 mm
- Recorrido: 406 mm
- Peso: 3,2 kg

De esta manera, en el inicio el actuador tendrá una determinada longitud (largo+recorrido) para llegar a tener la longitud de la pata (681mm) sin llegar a utilizar gran parte del recorrido de éste.



Figura 10- Actuador lineal 12V

5.3 Motor

El motor incorporado con el actuador seleccionado es un motor DC. Debido al precio y al control electrónico que se precisa para el motor DC brushed se valora como suficiente dicha opción. Características del motor:

| Carga (N) | | Corriente sin carga (A) | Corriente con carga total (A) | Velocidad (mm/s) | |
|-----------|----------|-------------------------|-------------------------------|------------------|-----------------|
| Dinámica | Estática | 12VDC | 12VDC | Sin carga | Con carga total |
| 1780 | 1780 | 4 | 12 | 24,892 | 19,812 |

Tabla 9- Características motor

La Figura 11 muestra la relación entre la carga y la intensidad para el correcto funcionamiento del motor.

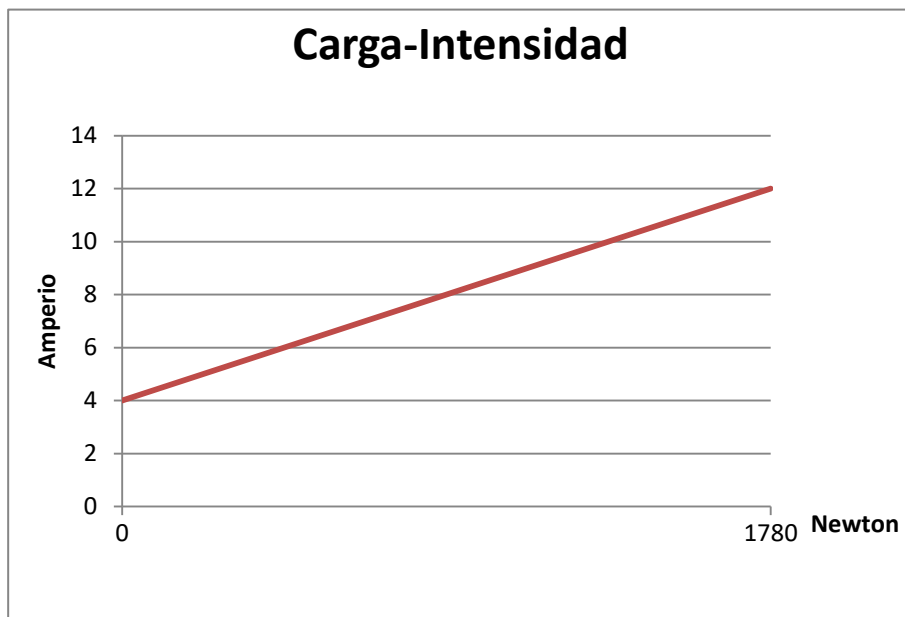


Figura 11- Gráfica carga-intensidad motor

La Figura 12 representa la velocidad a la que va a realizar el motor su movimiento dependiendo de la carga a soportar.

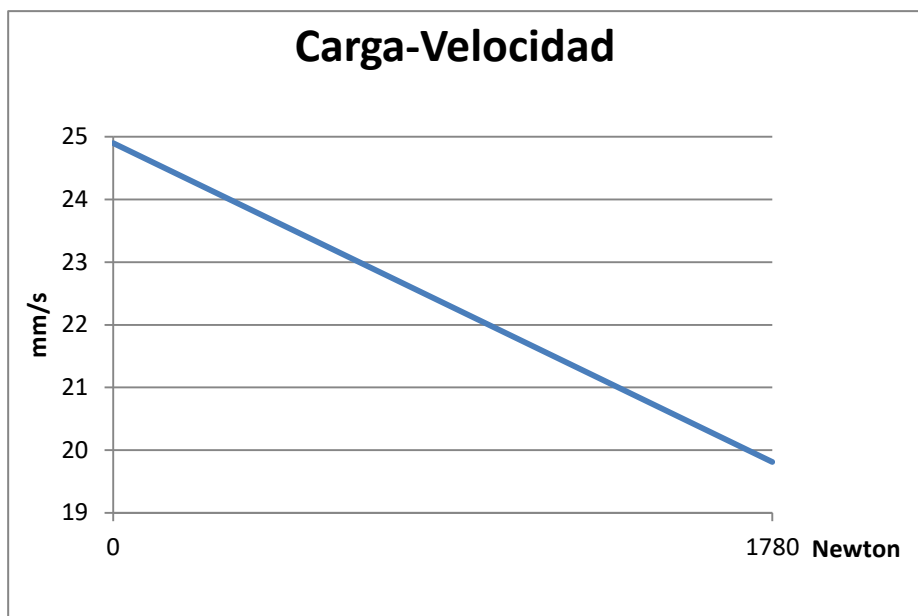


Figura 12- Gráfica carga-velocidad motor

Teniendo en cuenta que el motor con una carga total de 1780 N consume 12 A, se estima que la máxima carga del proyecto será de 50 kg. Para la obtención del valor del consumo se calcula la ecuación de la recta.

$$50kg = 490N \quad (1)$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{4 - 12}{0 - 1780} = \frac{2}{445} \quad (2)$$

$$(y - y_1) = m * (x - x_1) \rightarrow y - 12 = \frac{2}{445} * (x - 1780) \quad (3)$$

$$y = 6,21 A \quad (4)$$

5.4 Sensor

La selección del sensor en el proyecto viene determinada tras el análisis de los siguientes requerimientos:

- Tipo de material a detectar: La superficie a detectar es el suelo. Este material es un material sólido no metálico.

- Distancia a la que se sitúa: La distancia a medir es mayor de 50 mm. Además, la distancia a medir será de 68,1 cm cuando se trate de la menor distancia y 72 cm para el valor máximo.

- Interferencias ambientales: Para ello se deberá observar a que temperatura ambiente puede trabajar el sensor, y la influencia de la luz externa sobre éste.

En los sensores ópticos es muy importante el tipo y la cantidad de luz externa al sensor (luz ambiente), sobre la cual va a actuar el sensor, ya que dependiendo del tipo de luz un sensor puede funcionar correctamente o no.

En cambio, en los sensores ultrasónicos se debe tener en cuenta si ponemos dos sensores ultrasónicos cercanos, ya que estos podría interferir uno con el otro. El sonido ambiente no suele afectar a este tipo de sensores por el rango de frecuencias que utilizan.

Debido a los conceptos anteriores mencionados y observando la tabla comparativa se selecciona el sensor ultrasónico por cumplir con la finalidad del proyecto.

Las características del sensor HC-SR04, los pines y el funcionamiento son:

| Características | |
|--|-------------------|
| Alimentación | +5V DC |
| Frecuencia de ultrasonido | 40 KHz |
| Consumo (en reposo) | < 2 mA |
| Consumo (trabajando) | 15mA |
| Ángulo de apertura | < 15° |
| Rango de medición | 2cm a 400cm |
| Resolución | ±0,3cm |
| Duración mínima del pulso de disparo TRIG (nivel TTL) | 10 μ S |
| Duración del pulso ECO de salida (nivel TTL) | 100-25000 μ S |
| Tiempo mínimo de espera entre una medida y el inicio de otra | 50ms |

Tabla 10- Características sensor HC-SR04



Figura 13- Sensor HC-SR04

| Pines | |
|-------|---------------------------|
| VCC | +5V DC |
| TRIG | Disparo del ultrasonido |
| ECHO | Recepción del ultrasonido |
| GND | 0V |

Tabla 11- Pines sensor HC- SR04

En la Figura 14 se aprecia cómo solo es necesario aplicar un pulso de 10 us en el pin trigger para comenzar con la medición. A continuación el sensor envía una serie de 8 pulsos de 40 kHz y pone el pin de “Echo” a nivel alto. El pin “Echo” permanecerá a nivel alto hasta que se reciba el eco de los pulsos de 40 kHz. Para saber a la distancia a la que se encuentra el objeto, solo hay que medir el tiempo al que está el pin Echo a nivel alto.

Su funcionamiento se basa en generar una onda sónica en el emisor mediante un pulso en el pin “Trigger”. Esta onda, al encontrarse con un obstáculo rebotará volviendo al sensor y siendo registrada por el receptor, traduciéndose ésta en un pulso en el pin “Echo”.

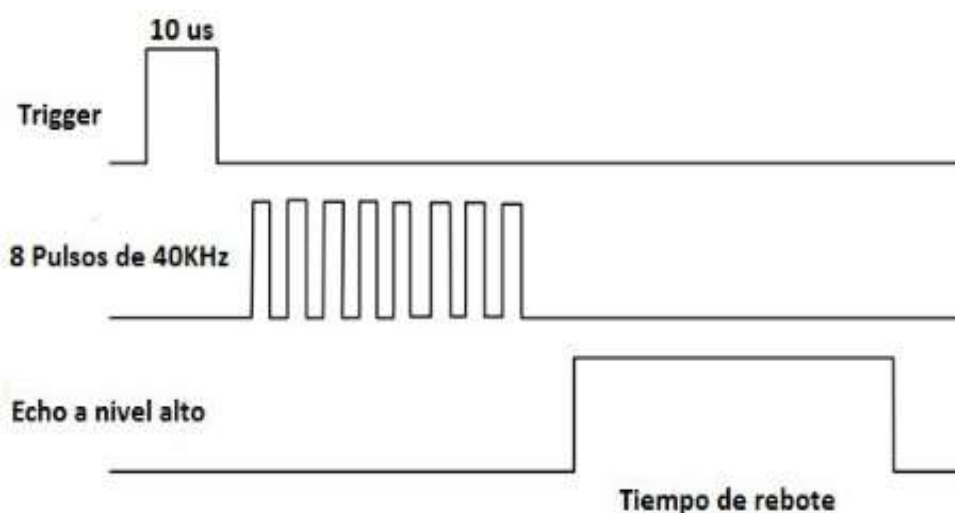


Figura 14-Funcionamiento HC-SR04

Primero, se envía un Pulso "1" de al menos de 10us por el pin "Trigger". Segundo, el sensor enviará 8 Pulsos de 40 kHz (Ultrasonido) y coloca su salida "Echo" a nivel alto. Esta salida "Echo" se mantendrá en alto hasta recibir el eco reflejado por el obstáculo a lo cual el sensor pondrá su pin Echo a bajo.

5.5 Activador (pulsador)

El pulsador a seleccionar debe tener un tamaño pequeño ya que de esta forma se podrá aprovechar mejor el espacio de la superficie con las bebidas, presentando las siguientes propiedades:

- Tamaño: 12x12x7 mm
- Corriente máxima: 50mA
- Tensión DC nom: 24V
- Fuerza de funcionamiento necesaria: 1,27 N
- Resistencia de contacto:100Mohm
- Pulsador N/A



Figura 15- Pulsador seleccionado

5.6 Batería

Las características a valorar para la elección de la batería son las siguientes:

- Cantidad de energía que puede almacenar (Wh)
- Máxima corriente que puede entregar, llamada descarga (Ah)
- Profundidad de descarga que puede sostener, siendo la energía que puede extraerse de una batería.
- Tiempo de carga.
- Dimensión
- Necesidades de las partes a alimentar
- Elementos a alimentar: Arduino Mega, sensores y motores.

Debido al planteamiento del proyecto se ha seleccionado una batería secundaria de Ion-Litio con las siguientes cualidades:

- Cantidad de energía: 250W/h
- Máxima corriente: 20000mAh
- Tiempo de carga: 10h
- Voltaje de entrada y salida: 12 V
- Dimensión: 22,2x14,2x4,2cm
- Peso:621g



Figura 16- Batería recargable

Las ventajas de esta batería son:

- **Peso:** Las baterías de ión litio concentran una gran densidad de energía y pueden así emitir una tensión tres veces superior a las baterías de níquel. Con la misma potencia, una batería de ión litio pesa hasta un 60% menos. Así pues se obtiene un mejor rendimiento con un peso reducido.

- **Ausencia de efecto memoria:** Las baterías de ión litio no son sensibles al efecto memoria. Así, no debe tenerse en cuenta ninguna precaución de carga. Se puede recargar la batería cuando más convenga, sin tener que esperar a que se descargue completamente, ni a proceder a un ciclo de carga completo de la batería.

- **Autodescargas leves o los efectos del almacenamiento en la capacidad de la batería:** Las baterías de litio son poco sensibles a la autodescarga. Una batería de este tipo pierde alrededor del 5% de su capacidad el primer día y, después, progresivamente el 10% de capacidad por mes.

- **Efecto leve de las temperaturas extremas cuando se utilizan las baterías:** Una de las grandes ventajas es que resiste bien a las utilizaciones a temperaturas extremas. Al utilizarla a -20 °C una batería de ión litio pierde como media un 35 % de su capacidad inicial, mientras que una batería de níquel pierde un 80 % de su capacidad.

Una de las partes a tener en cuenta para afrontar el proyecto con Arduino es la alimentación y la autonomía de la batería. Además, se deberá considerar la alimentación de sensores, actuadores y pulsador teniendo en cuenta las limitaciones eléctricas del microcontrolador.

Observando el valor de la máxima corriente por hora que es capaz de aportar la batería (20Ah) y haciendo referencia al cálculo de la página 26 de la intensidad de consumo para una carga de 50 kg, se realiza el siguiente cálculo para observar la duración de la batería.

$$20A * 1 h = 6,21A * x h \quad (5)$$

$$x = \frac{20}{6,21} = 3,22 \text{ horas} \quad (6)$$

La duración de la batería para el consumo del motor será de 3,22 horas. Esta duración es apropiada para la ejecución del proyecto, ya que el funcionamiento del motor se dará durante segundos, mientras se active la regulación mediante el pulsador. Además, la nivelación de la mesa se realizará con menor carga de 50 kg, ya que este peso ha sido estimado a la alza, y por lo tanto, habrá un menor consumo de intensidad, con lo que la duración de la batería será mayor.

6. DISEÑO

Una vez analizados los requisitos del proyecto y seleccionado los componentes, se procede a realizar la conexión, programación y simulación del proyecto tras haber analizado los diferentes elementos que van a formar parte de éste. Para ello es importante comenzar con la conexión de los elementos, dando paso al diagrama de flujo con su posterior programación y simulación en Proteus.

6.1 Conexionado de los elementos seleccionados

Los elementos seleccionados son estudiados para realizar su conexión en base a la alimentación recibida.

| | Alimentación | Corriente |
|----------|--------------|---------------------------------|
| Batería | 12V DC | 15A |
| Motor | 12V DC | 12 A |
| Sensor | 5V DC | Reposo: 2mA Trabajando: 15mA |
| Pulsador | 24V DC máx | 50 mA máx |
| Arduino | Vin: 9V | 800 mA máx |

Tabla 12- Voltaje e intensidades elementos

6.1.1 Arduino

La alimentación de éste se puede hacer de diferentes formas:

- Aplicar 6-12V en la clavija jack que disponen Arduino UNO, Mega, entre otros modelos
- Aplicar 6-12V entre el pin GND y el pin RAW (pin Vin en Arduino Mini)
- Alimentar mediante USB
- Aplicar 5V en el pin 5V (3,3V en ciertos modelos)

Se determina realizarla por el pin “Vin” al que llegará una tensión de 9V soportando un máximo de 1A para que de esta manera no se dañe.

Al haber seleccionado una batería de 12 V, se deberá reducir este valor y adecuar la intensidad de entrada. Para ello, se utilizará un circuito step-down, siendo éste un tipo de fuente de alimentación conmutada que tiene como objetivo reducir la tensión que alimenta al sistema, es decir, la tensión de salida es menor o igual que la tensión de entrada.

Para el diseño de este circuito es necesario establecer los parámetros como:

- Duty cycle, modula el ancho del pulso generado con el bloque PWM El voltaje de salida promedio aumentará a medida que el ciclo de trabajo, D , aumenta.

$$D = \frac{V_{salida}}{V_{entrada}} = \frac{9}{12} = 0,75 \quad (7)$$

- Resistencia

$$P = V * IR = 9 * 1 \text{ (intensidad entrada arduino)} = 9 W \quad (8)$$

$$R = \frac{V_0^2}{P} = \frac{9^2}{9} = 9 \Omega \quad (9)$$

Teniendo en cuenta que se tomará un valor comercial, se determinan 10 Ω de resistencia.

- Bobina

$$L_{min} = \frac{(1-D)RT}{2} = \frac{(1-0,75)*10*\left(\frac{1}{10*10^3}\right)}{2} = 125 \mu H \quad (10)$$

Teniendo en cuenta los valores comerciales, se toma una bobina de 150 uH.

- Rizado

$$\frac{\Delta v_o}{v_o} = 0,05 \quad (11)$$

- Condensador

$$\frac{\Delta v_o}{v_o} = \frac{(1-D)T^2}{8LC} \rightarrow 0,05 = \frac{(1-0,75)\left(\frac{1}{10 \times 10^3}\right)^2}{8 \times 150 \times 10^{-6} \times C} \rightarrow C = 41,66 \mu F \quad (12)$$

Al igual que con la inductancia, teniendo en cuenta los valores estándares comerciales, se toma un condensador de 47 μF .

- Frecuencia

Se selecciona una frecuencia de 10 kHz, debido a que con mayores frecuencias de conmutación, la ondulación de voltaje a la salida disminuirá. En la Figura 17 y Figura 18 se muestra la justificación de la frecuencia, mostrando la salida en el caso de 1 kHz frente a 10 kHz.

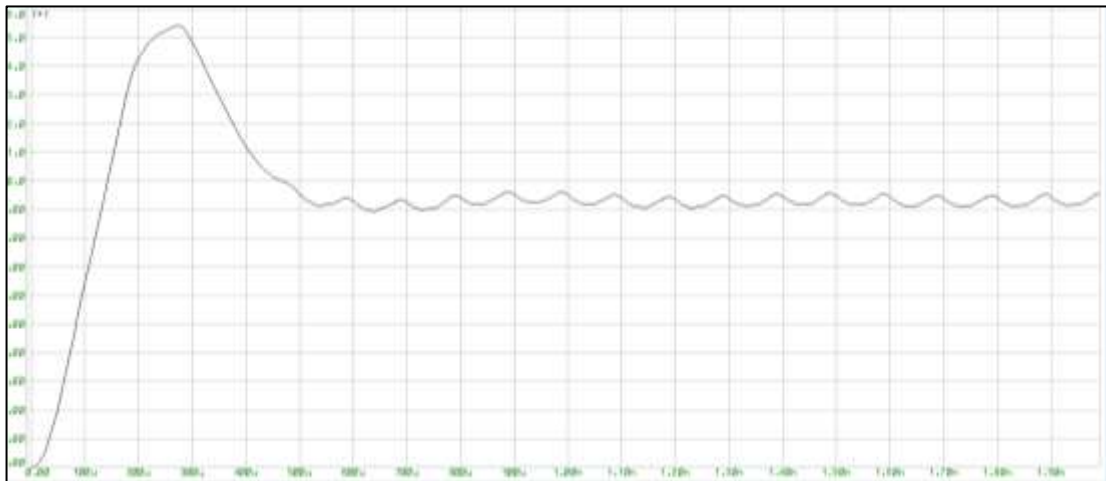


Figura 17- Tensión salida step-down 10kHz

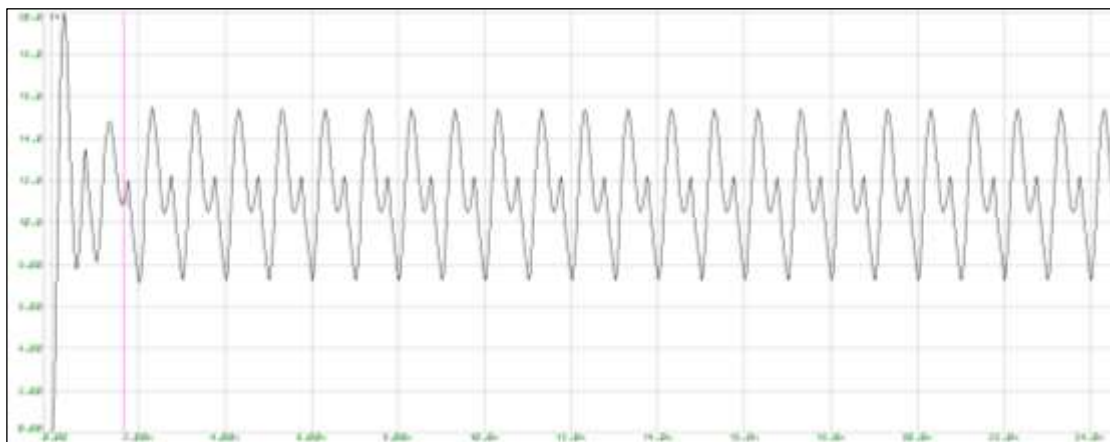


Figura 18- Tensión salida step-down 1kHz

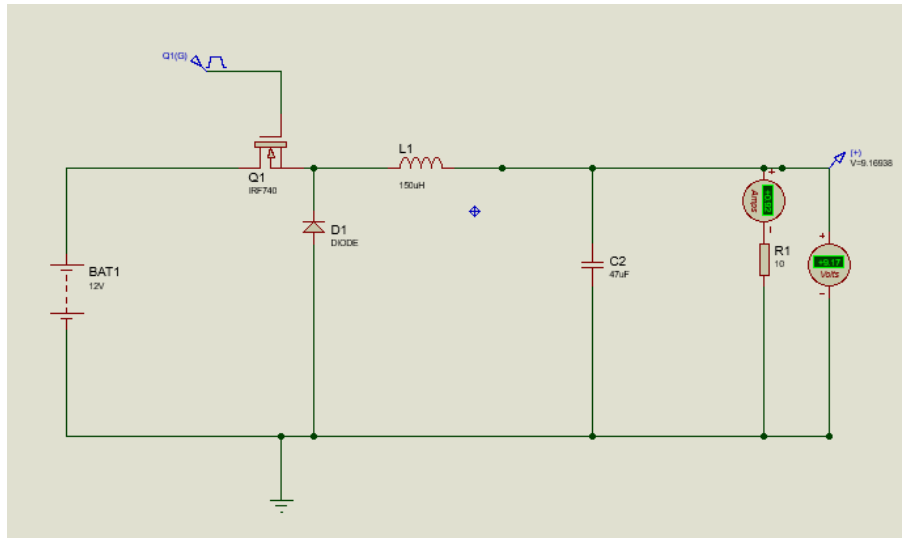


Figura 19- Simulación step-down Proteus

Como se puede observar en la Figura 17, hay un pico de tensión que puede dañar el Arduino.

Tras el estudio del step down se determina la utilización de un regulador lineal de tensión. De esta forma, se realiza un montaje más sencillo, evitando el pico de tensión y logrando la finalidad, contemplando dónde se va a realizar el proyecto.

El regulador seleccionado es el LM7809CT, cuyo conexionado se muestra en la Figura 20. Éste regulador proporciona una salida de +9V.

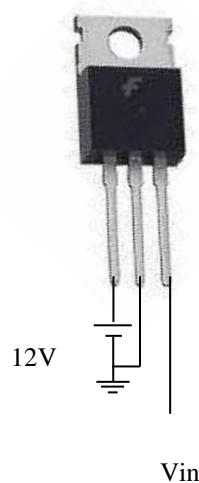


Figura 20- Conexionado regulador tensión LM7809CT

Para comprobar que la temperatura que adquiere el dispositivo entra dentro de unos márgenes de trabajo aceptables, se plantea el balance de temperatura. En la Figura 21 se aprecian los valores de las resistencias térmicas obtenidas del datasheet.

| Parameter | Symbol | Value | Unit |
|--|-----------------|------------|------|
| Input Voltage | V_i | -35 | V |
| Thermal Resistance Junction-Case (Note1) | $R_{\theta JC}$ | 5 | °C/W |
| Thermal Resistance Junction-Air (Note1, 2) | $R_{\theta JA}$ | 65 | |
| Operating Temperature Range | T_{OPR} | 0 ~ +125 | °C |
| Storage Temperature Range | T_{STG} | -65 ~ +150 | °C |

Figura 21- Característica regulador de tensión LM7809

$$\Delta V * I_{in} = (V_{in} - V_{out}) * I_{out} = (15 - 9) * 0,5 = 3 \text{ W} \quad (13)$$

$$T^a \text{ resistencias} = R_{jc} + R_{ja} = 5 \frac{^{\circ}\text{C}}{\text{W}} + 65 \frac{^{\circ}\text{C}}{\text{W}} = 70^{\circ}\text{C/W} \quad (14)$$

$$\begin{aligned} T^a \text{ real} &= (R_{jc} + R_{ja}) * \text{Potencia disipación} + T^a \text{ ambiente} = \\ &= 70 \frac{^{\circ}\text{C}}{\text{W}} * 3\text{W} + 25^{\circ}\text{C} = 235^{\circ}\text{C} \end{aligned} \quad (15)$$

El regulador seleccionado trabaja en condiciones óptimas con una temperatura máxima de 125 °C por lo que será necesario un disipador de temperatura. Se selecciona el HS115-ND y se realizan los cálculos para ver si es óptimo:

| | Valores |
|---|-------------|
| Paquete enfriado | TO-220 |
| Longitud | 19,05 mm |
| Ancho | 19,05 mm |
| Potencia disipada según aumento T^a | 2,5W a 60°C |
| Resistencia térmica en condiciones naturales | 24°C/W |
| Material | Aluminio |

Tabla 13- Características HS115-ND



Figura 22- Disipador térmico HS115-ND

$$R_{ja} = R_{jc} + R_{cs} + R_{sa} = 5 \frac{^{\circ}\text{C}}{\text{W}} + 1 \frac{^{\circ}\text{C}}{\text{W}} + 24 \frac{^{\circ}\text{C}}{\text{W}} = 30 \frac{^{\circ}\text{C}}{\text{W}} \quad (16)$$

$$T^{\text{a real}} = 25^{\circ}\text{C} + 30 \frac{^{\circ}\text{C}}{\text{W}} * 3\text{W} = 115^{\circ}\text{C} \quad (17)$$

Con el disipador se ha logrado bajar la temperatura de 235 °C a 115 °C. Con esta temperatura, el regulador lineal de tensión puede realizar de una manera óptima su trabajo.

6.1.2 Pulsador

La conexión del pulsador no puede ser una conexión directa al pin de entrada ya que cuando el interruptor esté abierto, el pin estará totalmente desconectado de cualquier tensión, produciendo un estado de alta impedancia. Esto produce que la entrada esté en un estado indeterminado, es decir, pudiendo asumir cualquier valor.

Por tanto, para evitar esta situación, se realizará un montaje con resistencia Pull Down. Esta resistencia fuerza LOW cuando el pulsador está abierto. Cuando está cerrado el PIN se pone a HIGH, y la intensidad que circula se ve limitada por esta resistencia.

El valor de esta resistencia se determina teniendo en cuenta que la corriente que se desea obtener es de 0,5 mA:

$$V = I * R \rightarrow 5\text{V} = 0,005\text{A} * R \rightarrow R = 10\text{k}\Omega \quad (18)$$

En la Figura 23 se muestra cómo se conecta el pulsador a la placa de Arduino Mega, de modo que el terminal 22 está conectado a tierra mediante la resistencia R1 de 10 kΩ

(circuito pull down), pero al activar el pulsador al terminal 22 se pondrá en estado lógico HIGH. Este pulsador será alimentado a 5V desde la placa de Arduino Mega.

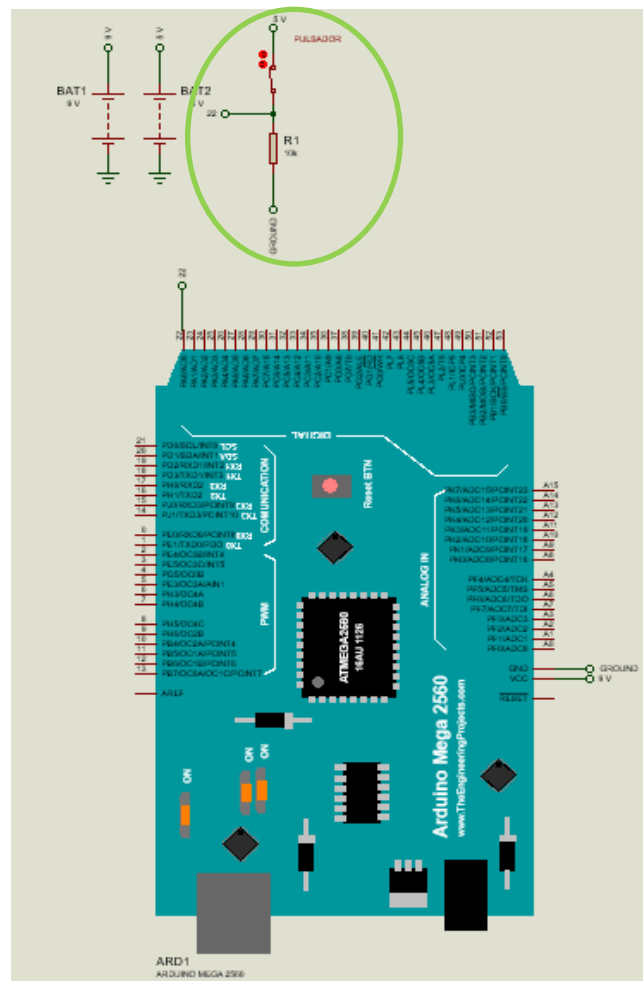


Figura 23- Circuito Pull Down pulsador

6.1.2 Sensor

La tensión de alimentación del sensor es de 5V, por lo que puede ir directamente conectado al pin que proporciona 5V del microprocesador. La conexión del sensor con Arduino viene dada por cuatro pines:

- Alimentación de 5 voltios.
- Tierra o común del circuito.
- Señal de disparo (trig).
- Señal de eco (echo).

| | Trig | Echo |
|----------|---------------|----------------|
| Sensor 1 | Pin 23 salida | Pin 24 entrada |
| Sensor 2 | Pin 25 salida | Pin 26 entrada |
| Sensor 3 | Pin 27 salida | Pin 28 entrada |
| Sensor 4 | Pin 29 salida | Pin 30 entrada |

Tabla 14- Pines empleados en sensores

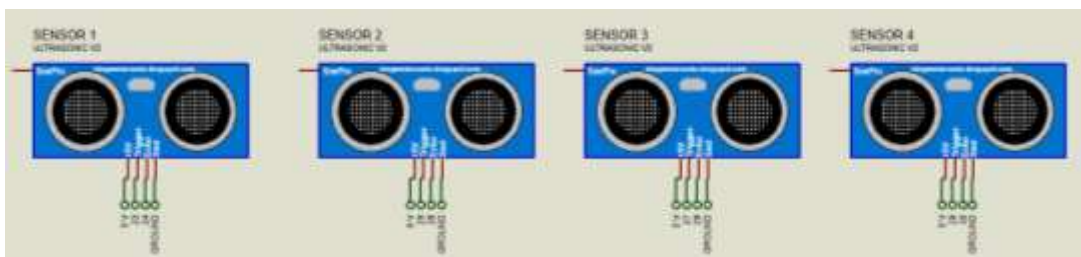


Figura 24- Conexión sensores ampliado

En la Figura 25 se observa el circuito del pulsador de activación, la conexión de los sensores ultrasonidos y la alimentación de 5V y 9V.

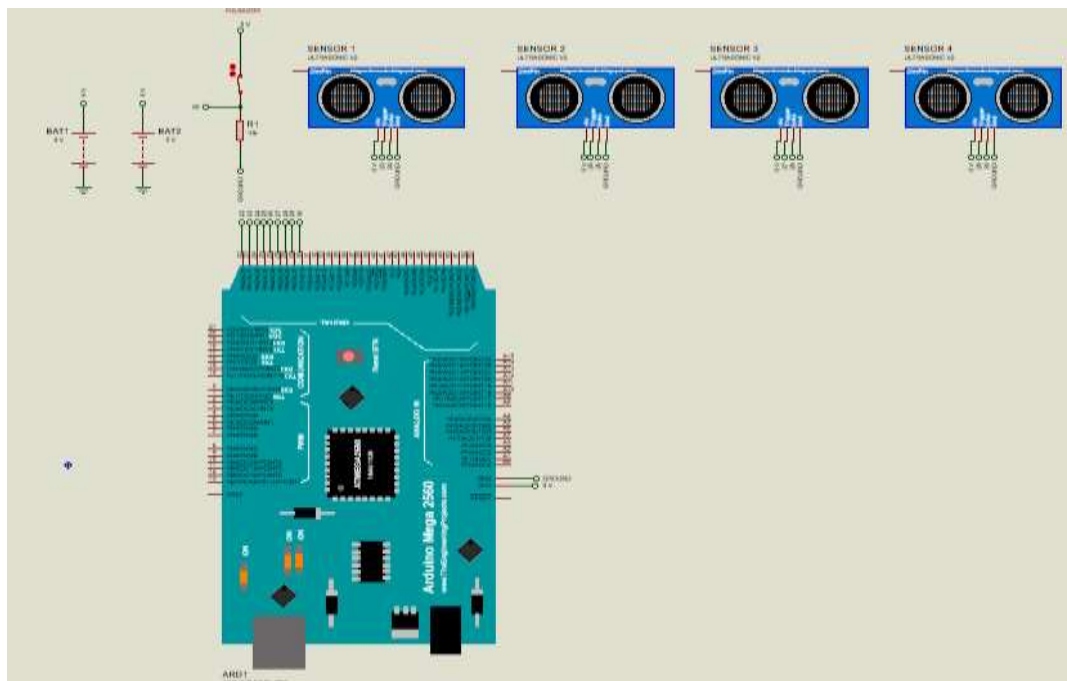


Figura 25- Conexionado sensores

6.1.3 Motor

El motor se alimenta de 12V DC y necesita para su funcionamiento 6,21A con una carga máxima de 50 kg. El funcionamiento del motor se activará con una señal de salida de Arduino. Como muestra el datasheet del actuador, se pueden realizar dos tipos de conexiones:

- Conexión estándar

Este conexionado se deberá realizar para longitudes de la carrera del actuador entre 2,5 cm y 15,2 cm.

| Motor | |
|----------------|----------------|
| 1 | 2 |
| M ⁻ | M ⁺ |

Tabla 15- Referencia pines motor conexión estándar

El actuador, en la posición de inicio, deberá de tener un recorrido de 7'5 cm para estar posicionado a la misma altura que la pata de la mesa. Tras estudiar el pavimento y sus irregularidades, se observa que el desnivel no va a ser mayor a 3,9 cm. El recorrido del actuador en el proyecto irá hasta 11,4 cm, por lo que le corresponde esta conexión.

Arduino tiene pines de entrada y de salida para comunicarse físicamente con su entorno. Los pines de salida pueden dar una pequeña cantidad de energía que sirve por ejemplo para encender un LED, pero no pueden alimentar a un motor.

Para ello necesitamos un circuito que haga de intermediario. Este circuito, controlador o driver, tomará energía de la batería y siguiendo las instrucciones de Arduino hará funcionar el motor.

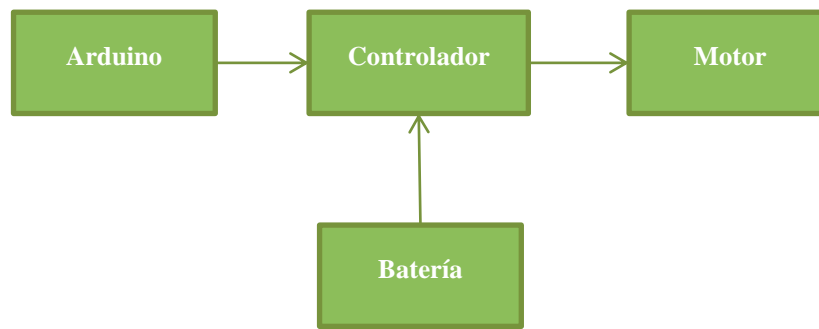


Figura 26- Esquema general controlador motor y arduino

El controlador se selecciona atendiendo a las siguientes características:

- Que permitan invertir el sentido de giro del motor.
- Que permitan graduar la velocidad de giro del motor.
- Características motor, en el proyecto 12V y 6,21A para la carga máxima.

Atendiendo a estas condiciones, se selecciona el módulo XY-15AS PWM MOTOR DC 6-36V. En la Figura 27 se muestra el conexionado del motor 1 con el controlador.

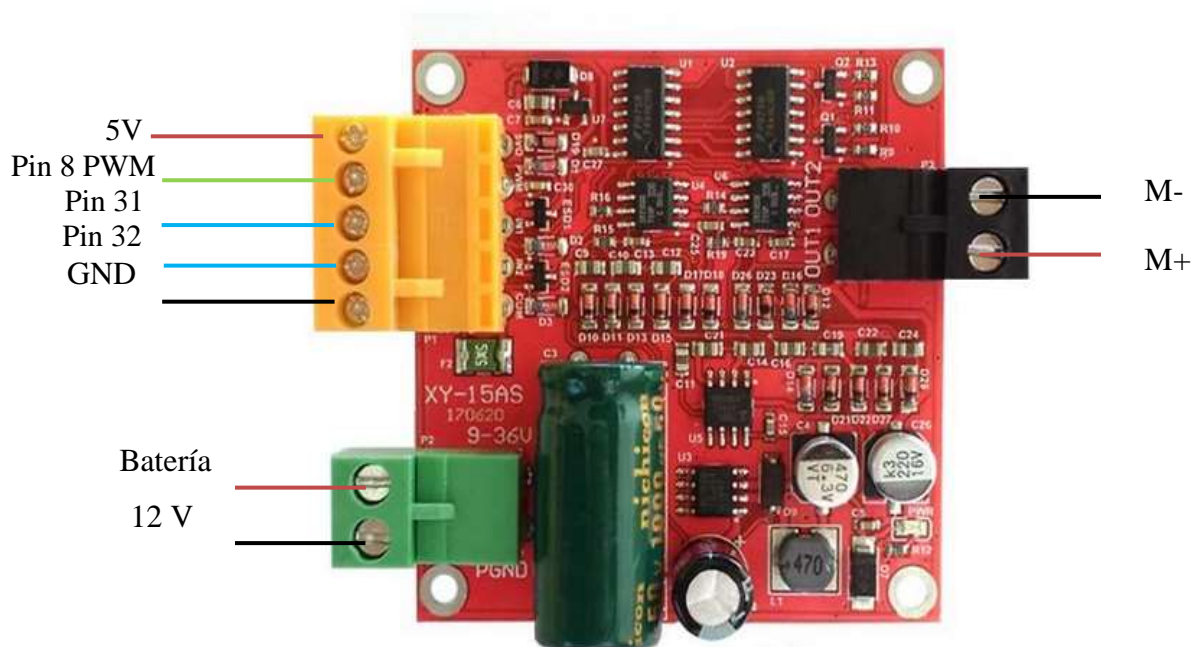


Figura 27- Conexionado del driver motor uno del actuador

Mediante la señal PWM del “pin 8” se determina la velocidad a la que va a funcionar el motor DC del actuador. El giro será invertido mediante los valores de las entradas del “pin 31” y “pin 32”.

| | Motor 1 | Motor 2 | Motor 3 | Motor 4 |
|---------------------------|----------------|----------------|----------------|----------------|
| Pin señal PWM | Pin 8 | Pin 9 | Pin 10 | Pin 11 |
| Pin giro derecha | Pin 31 | Pin 33 | Pin 35 | Pin 37 |
| Pin giro izquierda | Pin 32 | Pin 34 | Pin 36 | Pin 38 |

Tabla 16- Pines conexión motor 1, 2, 3 y 4

- Conexionado efecto hall

Se deberá realizar para longitudes de la carrera del actuador entre 15,2 cm y 101,6 cm.

| Motor | | Sensor Hall | | | |
|----------------|----------------|--------------------|-------|--------------------|---------------------|
| 4 | 5 | 2 | 3 | 1 | 6 |
| M ⁺ | M ⁻ | GND | +5VDC | Señal de extensión | Señal de retracción |

Tabla 17- Referencia pines conexión efecto hall

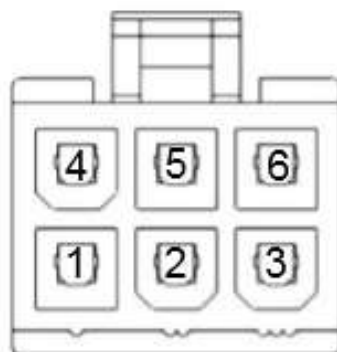


Figura 28- Pines efecto hall

En la Figura 28 se puede observar cómo sin necesidad de un módulo controlador se puede controlar la velocidad y el sentido de giro del motor. En el proyecto planteado no será necesario este segundo conexionado ya que la distancia de recorrido es menor de 15,2cm. Debido a que el recorrido puede ser mayor en diferentes circunstancias, por ejemplo cuando se dé una rugosidad superior del suelo en otro pavimento. Esta conexión será realizada para dichos casos.

6.1.4 Leds para el aviso de errores

Para el funcionamiento del mecanismo se desean observar errores como puede ser una incorrecta lectura de los sensores, así como, que la carga de la batería sea menor a lo necesario para la alimentación del motor.

Los límites de recorrido del actuador, como se ha citado anteriormente, van a ser de 75mm (límite inferior) y 114mm (límite superior). De esta forma, cuando los actuadores se encuentre en el “punto de inicio: equiparación con las patas”, el valor del sensor será de 681 mm, es decir, la longitud de la pata.

El valor de lectura máximo del sensor, será el valor de lectura mínimo más la rugosidad del pavimento máxima. Esto hace un valor de 720mm.

Además, la resolución de 3 mm del sensor, hace que el incremento de medidas leídas por el sensor sea de 3 mm en 3 mm.

En base a lo explicado en el párrafo anterior, cualquier lectura realizada por el sensor que no se encuentre entre 681 mm (límite inferior) y 720 mm (límite superior) será un error de lectura. Dicho error se visualiza con el encendido de un led rojo conectado al pin “39” a través de una resistencia calculada a continuación:

$$R = \frac{V_{cc} - V_d}{I_{nominal}} = \frac{5 - 3,5}{0,01} = 150 \Omega \quad (19)$$

El funcionamiento de los actuadores se verá afectado por la carga de la batería. La comprobación de la carga se llevará a cabo con el sensor de tensión FZ0430. La conexión se muestra en la Figura 34.



Figura 29- Conexión sensor de tensión FZ0430

La tensión máxima que podemos medir será 25V para un procesador de tensión de alimentación Vcc 5V, y 16.5V para un procesador de Vcc 3.3V. A continuación se realiza el cálculo estimando que cuando la batería sea de 11,5V, necesitará ser cargada.

El sensor de tensión tendrá una tensión 5 V en el pin analógico cuando la carga sea de 25V, de esta forma:

$$\frac{11,5 V \cdot 5V}{25 V} = 2,3 V \quad (20)$$

Así, cuando la lectura en el pin "A0" sea igual o menor que 2,3 V se encenderá un led verde conectado al pin "40" de Arduino a través de una resistencia de 150 Ω calculada de la misma forma que para el led rojo.

6.2 Diseño gráfico

El diseño del prototipo de la mesa con los elementos seleccionados se muestra a continuación.

En la Figura 35 se desarrolla el alzado de la mesa con sus correspondientes sensores, actuadores, batería, microcontrolador y pulsador.

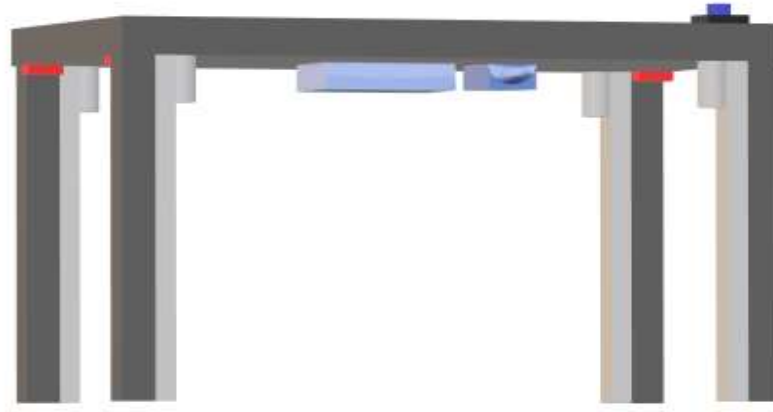


Figura 30- Prototipo mesa

En la Figura 36 se muestra la planta de la mesa en la que se puede apreciar el pulsador que utilizará el cliente para la iniciación de la nivelación de la mesa.



Figura 31- Vista superior prototipo mesa

En la Figura 37 representa la vista inferior, en la cual se pueden apreciar todos los componentes además del cableado.

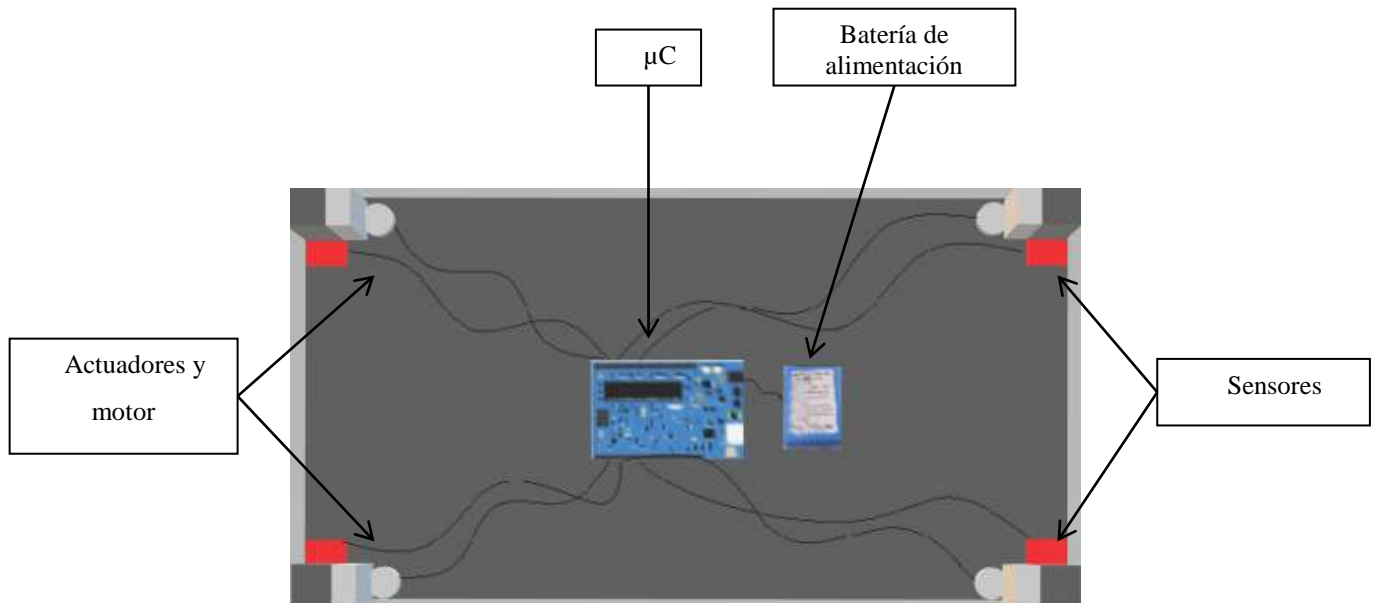


Figura 32- Vista inferior prototipo mesa

6.3 Diagrama de flujo

El diagrama de flujo se realiza como condición primera para proceder a la programación. Para ello, se tuvieron en cuenta los siguientes puntos:

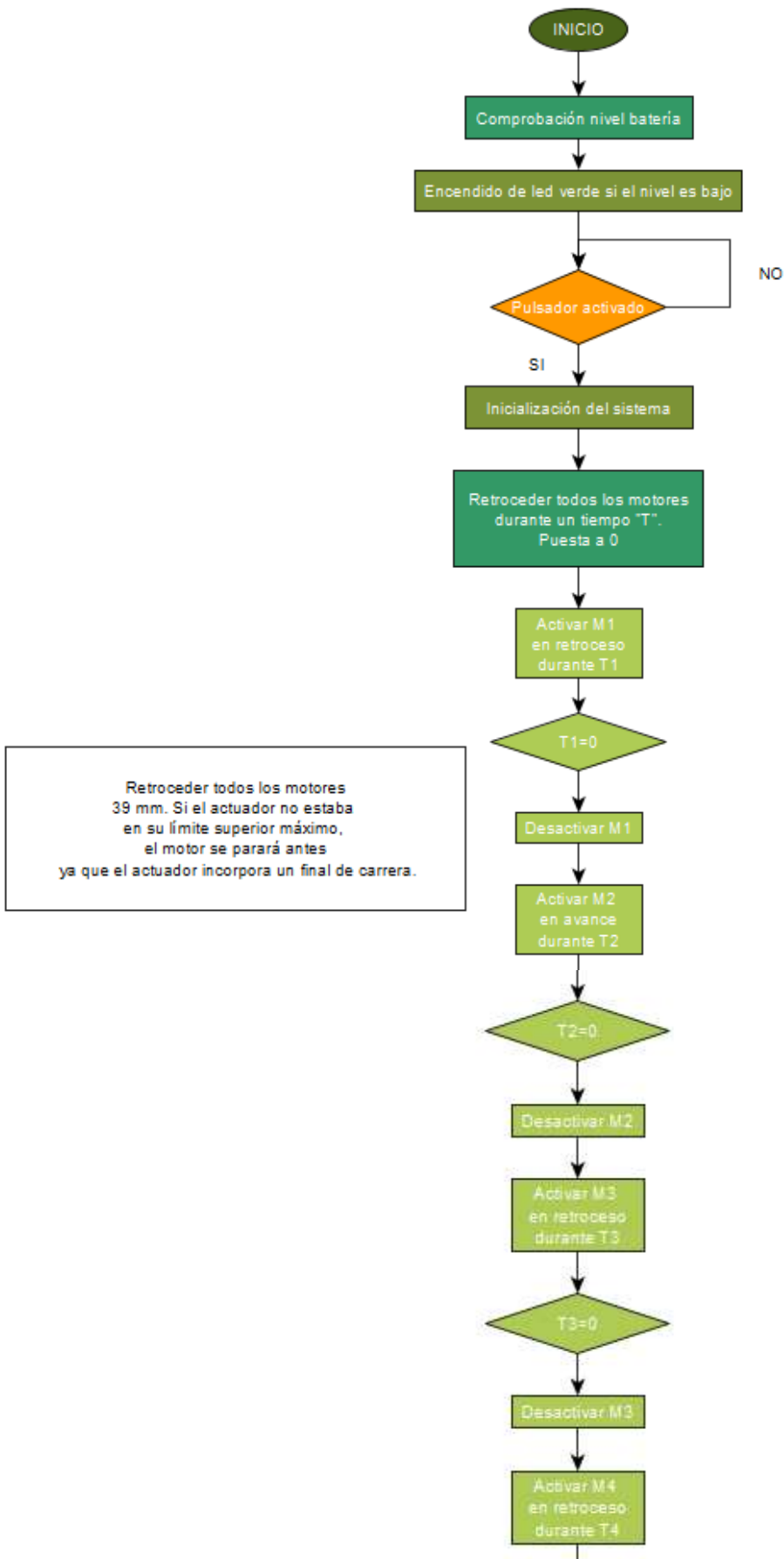
- La inicialización del sistema se realiza retrocediendo los actuadores hasta el máximo, posición de puesta a cero.
- Los actuadores tendrán un límite inferior y superior de recorrido. El límite inferior será de 75 mm debido a que el actuador en esta posición tendrá la misma longitud que la pata de la mesa. Los cálculos de este límite inferior se muestran a continuación:

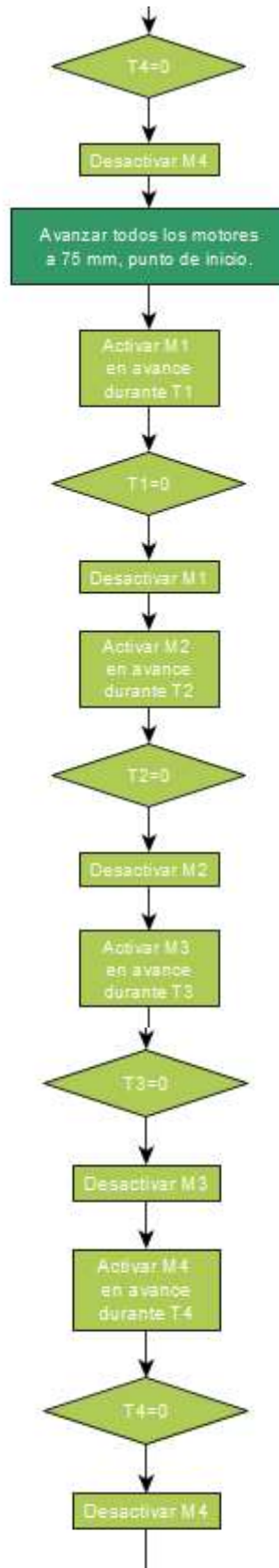
$$\begin{aligned} \text{Altura real sensor al final de la pata} &= \text{Altura mesa} - \text{Anchura base mesa} = \\ &= 720 - 39 = 681 \text{ mm} \end{aligned} \quad (21)$$

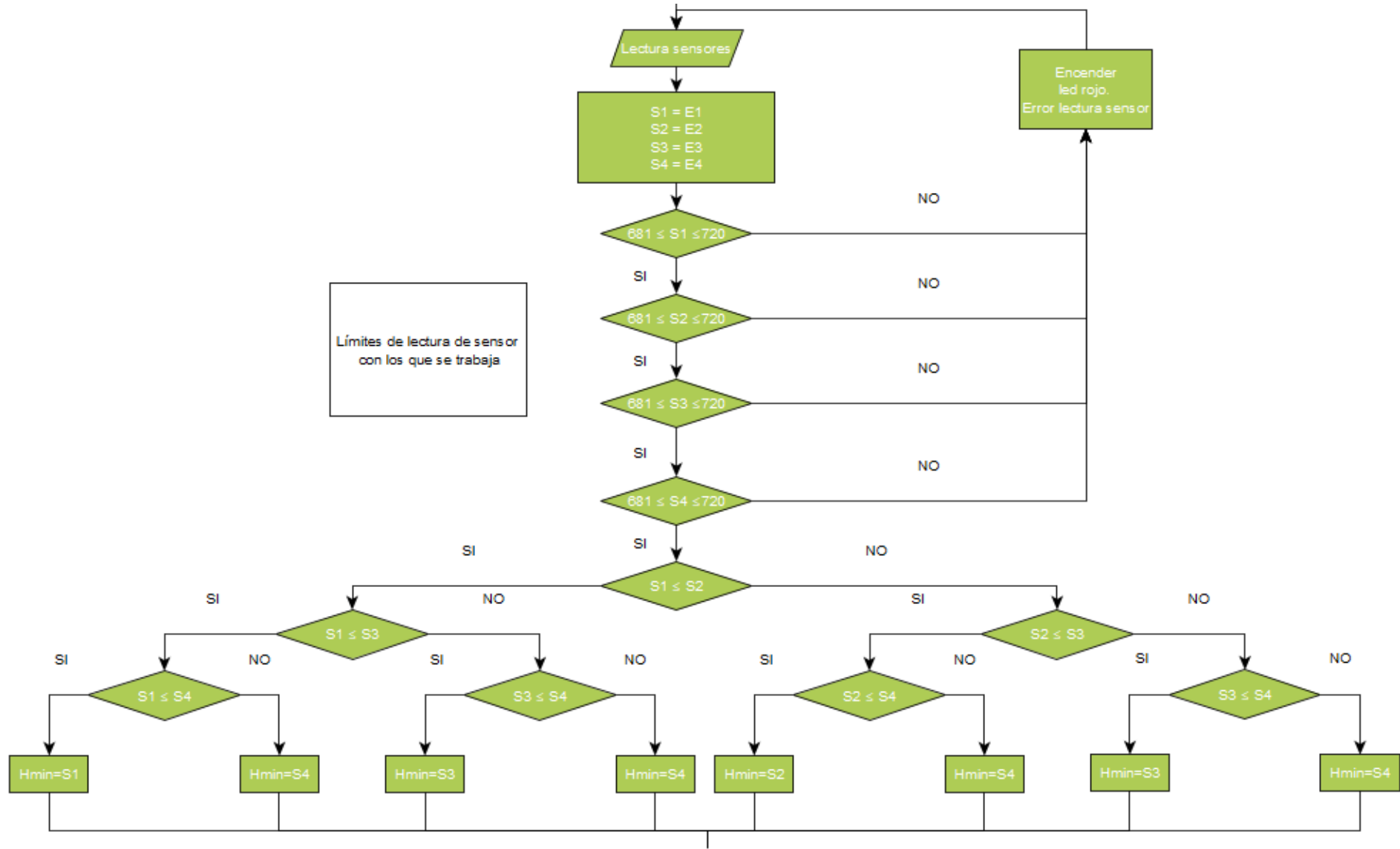
$$\begin{aligned} \text{Posición inicial} &= \text{Altura real sensor} - \text{largura actuador} = \\ &= 681 - 606 = 75 \text{ mm} \end{aligned} \quad (22)$$

- Los valores de la lectura del sensor con la que se trabajan están comprendidos entre 681 mm, punto de inicio en el que el actuador está equiparado a la mesa y 720 mm, como se ha explicado en el apartado del conexionado de leds para el aviso de errores. El led rojo será el que indique este error. El programa permanecerá leyendo los sensores mientras estos den una lectura errónea. Dicha lectura se puede dar cuando el desnivel es mayor de 39 mm, pudiéndose solventar moviendo la mesa a un desnivel menor y reseteando el sistema. También se puede dar dicha situación cuando los motores por una avería o por falta de batería no hayan logrado llegar al punto de inicio de 75 mm.
- Una vez se realiza la autorregulación, y se desea realizar una regulación nueva, los actuadores primero vuelven a la posición de puesta a cero. Tras ello, vuelven a su posición de inicio de 75 mm. De esta manera se controla la posición del actuador.
- Tras presionar el pulsador se realiza la autonivelación. Mientras esté en ejecución este proceso, una nueva pulsación en el pulsador, no será atendida.
- El valor de la carga de la batería puede ser inferior a lo necesario para la alimentación del motor. Para ello, cuando este valor sea menor de 11,5V se encenderá un led verde. Esta ejecución se realiza mediante un sensor de tensión, explicado en el apartado de conexionado de leds para el aviso errores.

En la Figura 38 se muestra el diagrama de flujo completo. Se pueden diferenciar dos partes: La primera de inicialización del sistema, y la segunda de auto-regulación de la mesa:







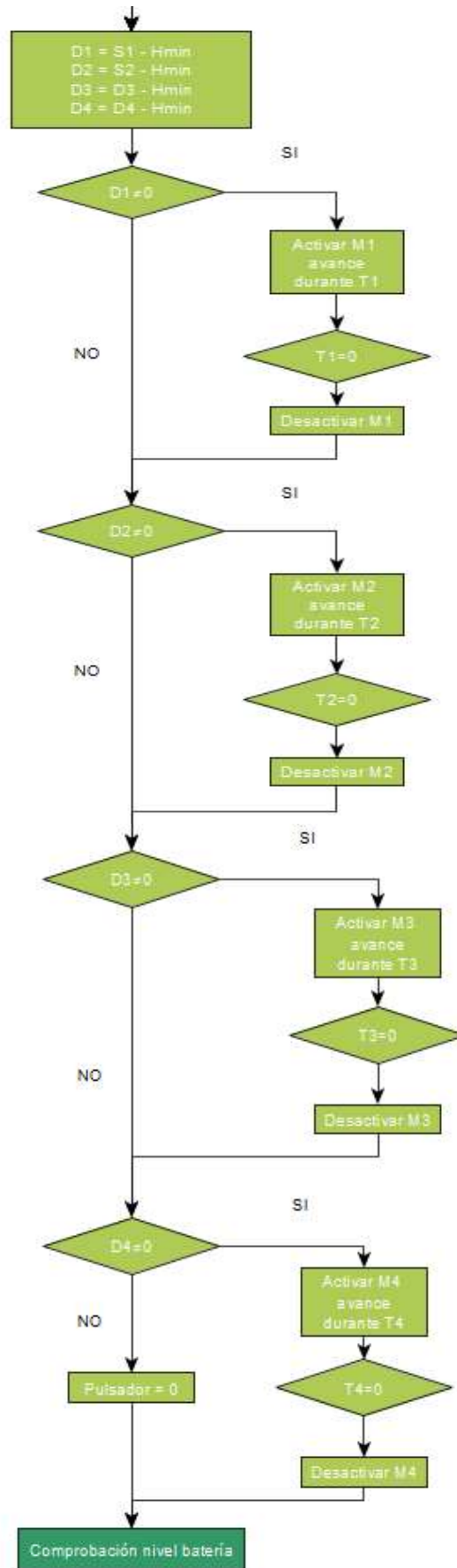


Figura 33- Diagrama de flujo

6.4 Programación

La programación se va a realizar con el software de Arduino. Para el entendimiento del código se deben explicar los siguientes apartados:

- La lectura de los sensores se realizará uno a uno para evitar interferencias en la medición y asegurar una correcta lectura.
- La correcta lectura del pulsador se realiza con un sistema antirebote mediante software.
- La lectura del pin analógico “A0” da un resultado entre los valores 0 y 1023 debido a que dicha entrada es de 10 bits. Para obtener el valor en Voltios, se realiza la siguiente conversión:

$$\frac{\text{Lectura A0} \cdot 5V}{1023} = \text{Resultado en V} \quad (23)$$

- La velocidad a la que el motor realizará su movimiento va a ser fija. El cálculo se realiza atendiendo al datasheet del actuador, observando que la velocidad para este cuando no haya carga será de 24,892 mm/s y con una carga total de 181,437 kg será de 19,812 mm/s. La carga estimada para este proyecto es de 50 kg, por lo que la velocidad fijada del actuador:

$$50 \text{ kg} = 490 \text{ N} \quad (24)$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{24,892 - 19,812}{0 - 181,437} = -0,02799 \quad (25)$$

$$(y - y_1) = m * (x - x_1) \rightarrow y - 19,812 = -0,02799 * (50 - 181,437) \quad (26)$$

$$y = 23,49 \text{ mm/s} \quad (27)$$

Para asegurarse de que el motor puede mover dicho peso, la velocidad deberá de estar entre los valores de 19,812 mm/s y 23,49 mm/s.

- La velocidad del motor se va a fijar con una señal PWM, la cual es proporcionada en Arduino Mega en los pines de 2 al 13 y del 44 al 46. Dichas salidas son de 8bits, por lo que el valor de 256 de la PWM determinará un Duty Cycle del 100% y una velocidad máxima de 24,892 mm/s. La velocidad que se desea fijar es de 20 mm/s.

$$\text{Valor PWM} = \frac{20 \cdot 256}{24,892} = 205,69 \quad (28)$$

El valor de la señal PWM que determina el ciclo de trabajo tiene que ser un número entero, por lo que se realiza una aproximación a 206.

$$\text{Valor PWM} = \frac{206 \cdot 24,892}{256} = 20,0303 \text{ mm/s} \quad (29)$$

- Teniendo en cuenta la velocidad seleccionada de 20,0303 mm/s y las distancias que deberá moverse el motor que provienen de la lectura de los sensores, se determinará el tiempo de funcionamiento del actuador.

Como se ha explicado en el diagrama de flujo, el motor será activado durante un tiempo “x” dependiendo de la distancia que deba recorrer, y tras ello, se parará. Para ello, se utilizará una función denominada “delay” que hace que el procesador espere hasta que pasa el tiempo determinado. El parámetro de entrada para esta función es entero, se debe realizar en milisegundos. Debido a ello, y a que el tiempo que deberemos encender nuestro motor no será un valor entero, se realizará una aproximación de dicho valor. Esto conlleva a un error en la distancia de movimiento debido al redondeo del valor del tiempo, el cual se muestra en la tabla siguiente.

En la tabla que se muestra a continuación se destaca en rojo los valores máximos de error relativo en tanto por ciento, pudiendo concluir que el error de movimiento debido a la aproximación del tiempo será mínimo, teniendo un error relativo máximo de 0,15%.

| Lectura (mm) | Distancia a recorrer (mm) | Aproximación tiempo (ms). Delay | Movimiento real con la aproximación del tiempo | Error absoluto (mm) | Error relativo (%) |
|--------------|---------------------------|---------------------------------|--|---------------------|--------------------|
| 681 | 75 | 3744 | 74,993 | 0,0066 | 0,0088 |
| 684 | 3 | 150 | 3,005 | -0,0045 | -0,1514 |
| 687 | 6 | 300 | 6,009 | -0,0091 | -0,1514 |
| 690 | 9 | 449 | 8,994 | 0,0064 | 0,0712 |
| 693 | 12 | 599 | 11,998 | 0,0019 | 0,0155 |
| 696 | 15 | 749 | 15,003 | -0,0027 | -0,0179 |
| 699 | 18 | 899 | 18,007 | -0,0072 | -0,0401 |
| 702 | 21 | 1048 | 20,992 | 0,0083 | 0,0394 |
| 705 | 24 | 1198 | 23,996 | 0,0037 | 0,0155 |
| 708 | 27 | 1348 | 27,001 | -0,0008 | -0,0030 |
| 711 | 30 | 1498 | 30,005 | -0,0054 | -0,0179 |
| 714 | 33 | 1648 | 33,010 | -0,0099 | -0,0300 |
| 717 | 36 | 1797 | 35,994 | 0,0056 | 0,0155 |
| 720 | 39 | 1947 | 38,999 | 0,0010 | 0,0027 |

Tabla 18- Error por aproximación de tiempo

7. SIMULACIÓN

La simulación ha sido desarrollada con Proteus Design Suite.

En primer lugar, como se ha podido observar en el apartado de conexionado, se desarrollaron los circuitos step-down para el paso de una tensión de 12V a 7V y 9V.

Tras ello, se han realizado diferentes pruebas en la placa de Arduino Mega. La primera trata de una programación simple mediante el encendido de un led como se observa en la Figura 34.

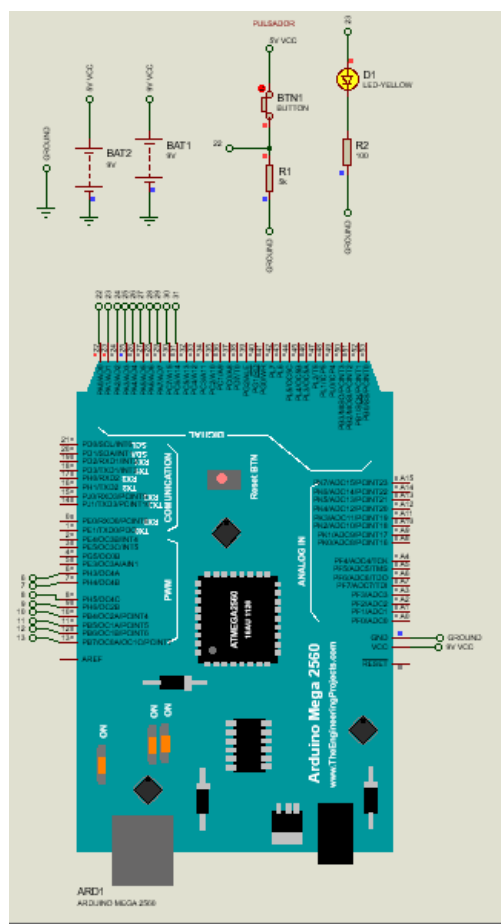


Figura 34- Simulación pulsador y led

La segunda prueba se ha realizado para la programación del sensor ultrasonido Hc-sr04, pudiéndose observar mediante una pantalla LCD las diferentes lecturas variando el valor de un potenciómetro.

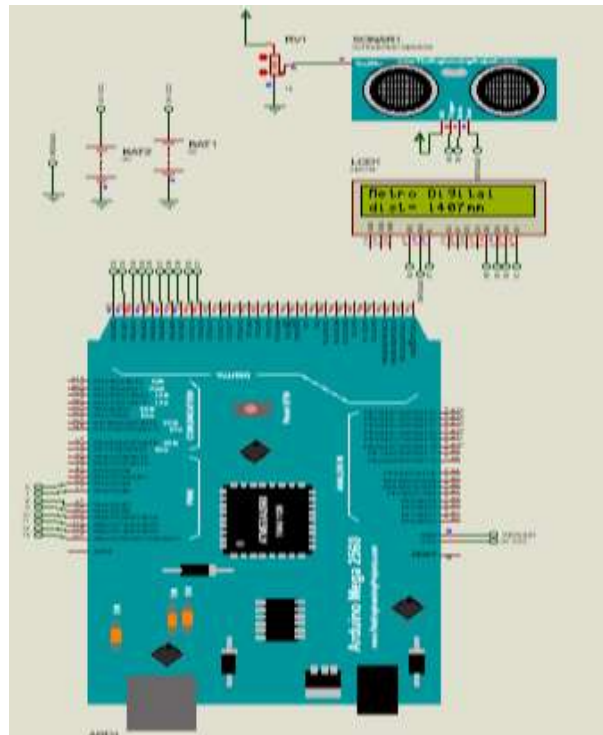


Figura 35- Simulación lectura sensor ultrasonido 140cm

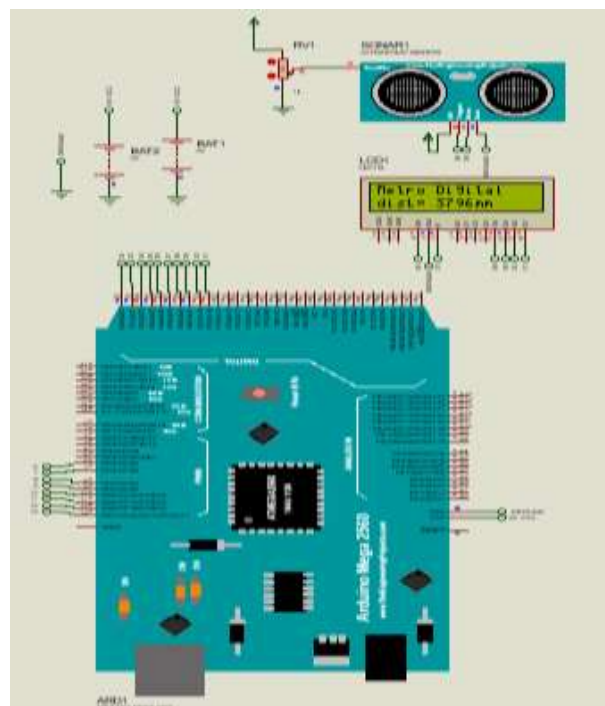


Figura 36- Simulación lectura sensor ultrasonido 379cm

En tercer lugar se desarrolló la programación del motor para ver su funcionamiento. Al no disponer del driver seleccionado para el motor en Proteus, se ha desarrollado la simulación con el driver L298, accionando el movimiento del motor en ambos sentidos, representado en la Figura 37.

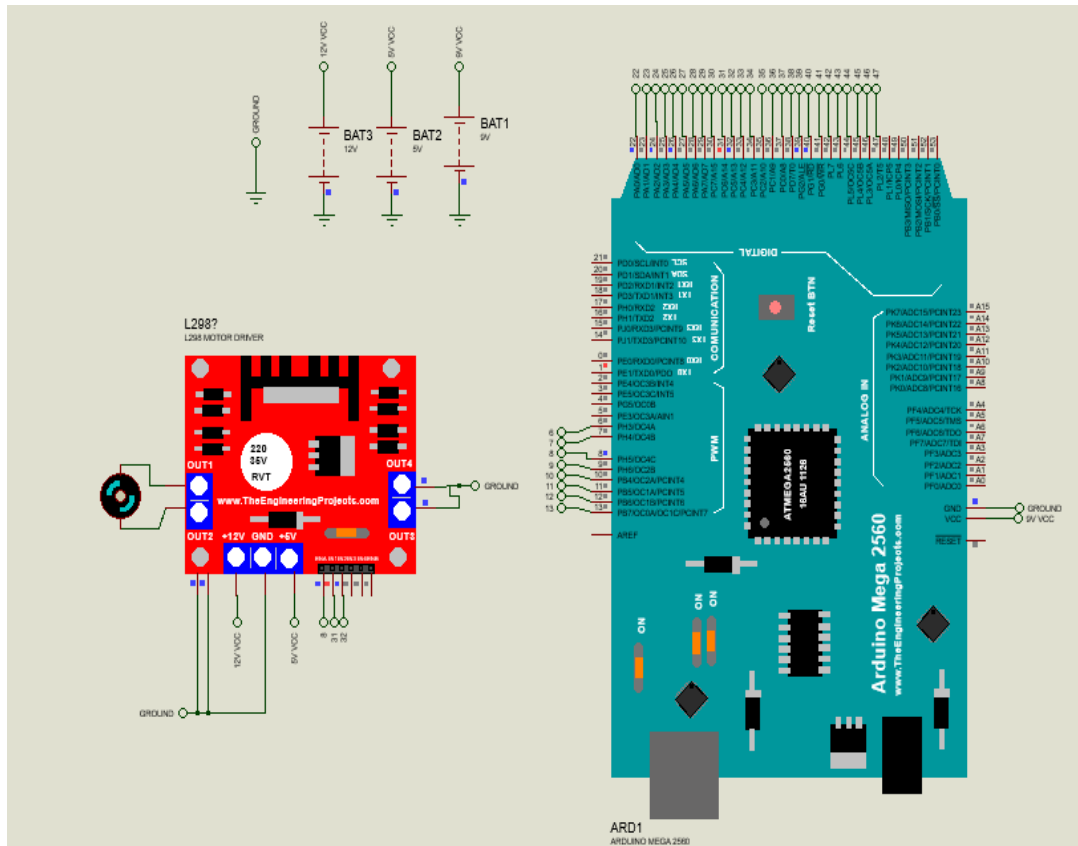


Figura 37- Simulación motor con driver

Por último se realiza la simulación del proyecto, pudiendo observar en la Figura 38 los elementos y el conexionado.

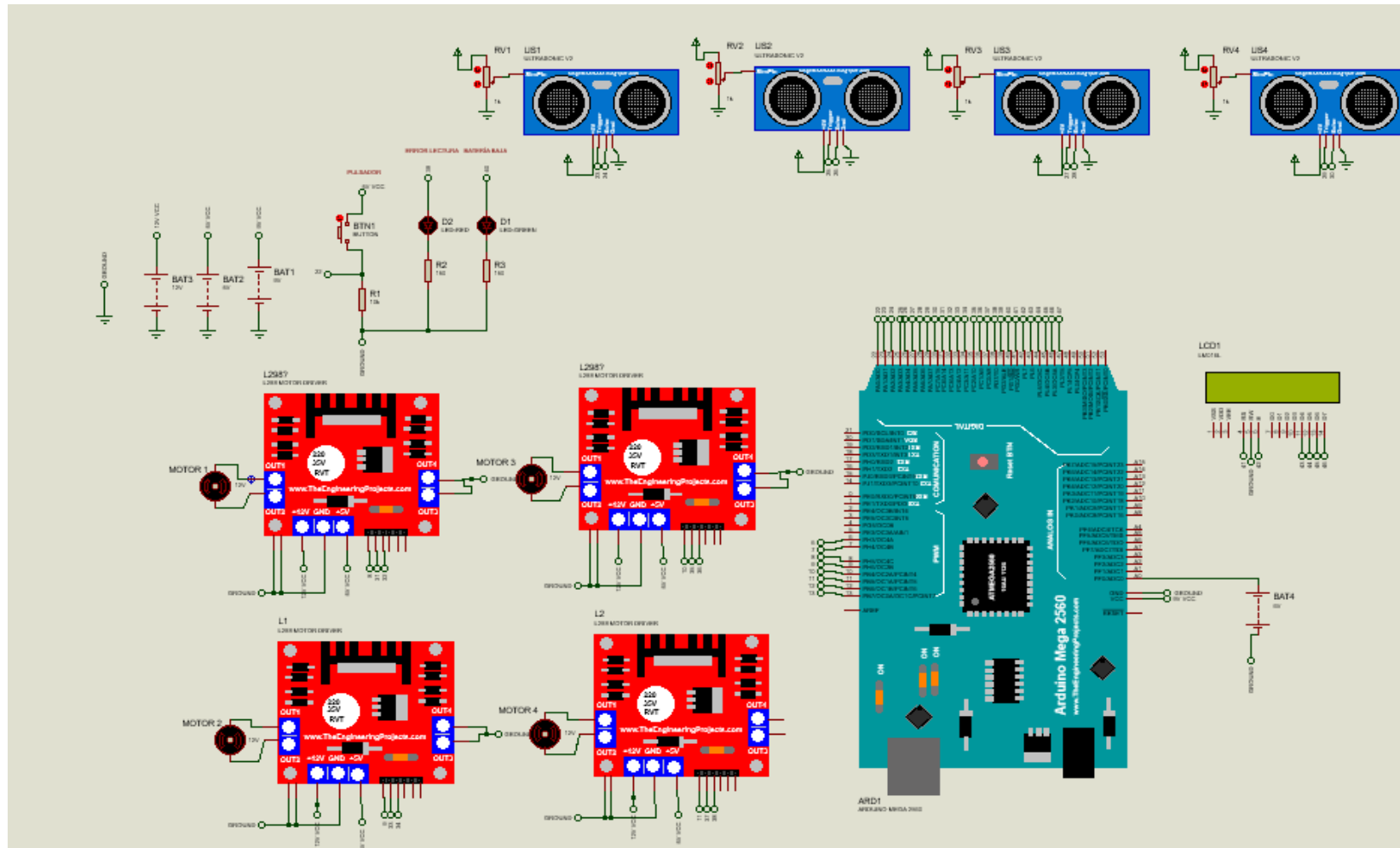


Figura 38- Simulación proyecto

8. PRESUPUESTO

En este apartado se especifican los costes asociados a la realización del proyecto, a su diseño y realización del hardware. Los costes se desglosan en:

- Coste en recursos humanos
- Coste de software asociado
- Coste de material

8.1. Coste en recursos humanos

En este apartado se tiene en cuenta el valor asociado a la mano de obra, dividiéndose éste en la realización por la ingeniera junior y la supervisión por la ingeniera senior.

| Tipo de mano de obra | Concepto | Cantidad (horas) | Coste horario (€/h) | Coste total (€) |
|----------------------|----------------------------------|------------------|---------------------|-----------------|
| Ingeniera junior | Diseño y realización electrónica | 600 | 12 | 7200 |
| Ingeniera senior | Supervisión | 45 | 24 | 1080 |
| Subtotal | | | | 8280 |

Tabla 19- Presupuesto recursos humanos

El coste derivado de recursos humanos suma una cuantía de 8280 €

8.2. Coste de software asociado

A continuación se muestran los costes relacionados con los software utilizados para el desarrollo del presente proyecto y el elemento fundamental para la ejecución de estos.

| Licencia de software | Tipo de licencia | Coste horario (€/h) | Coste total (€) |
|------------------------|------------------|---------------------|-----------------|
| Proteus 8 Professional | Estudiante | 792 | Amortizado |
| Arduino 1.8.10 | Estudiante | 0 | Amortizado |
| Microsoft Office 2016 | Estudiante | 149 | Amortizado |
| Subtotal | | | 0 |

Tabla 20- Coste de software asociado

| Componente | Concepto | Cantidad | Coste unitario (€) | Coste total (€) |
|--------------------|--------------------|----------|--------------------|-----------------|
| HP LAPTOP 5IH88AD6 | Ordenador portátil | 1 | 840 | Amortizado |
| Subtotal | | | | 0 |

Tabla 21- Coste elemento ejecución software

El coste relacionado con el software asociado suma una cuantía de 0€ ya que los equipos y software están amortizados.

8.3. Coste de material

A continuación, se muestran en forma los costes asociados a la adquisición del material necesario para la elaboración del estudio.

| Componente | Referencia | Necesidad | Precio unitario - inc. IVA (€) |
|---|---------------|-----------------|--------------------------------|
| <i>Mesa</i> | | 1 | 67,92 |
| <i>Actuadores lineales</i> | SKU:PA-04 | 4 | 145,25 |
| <i>Sensores distancia</i> | SKU026931 | 4 | 1,95 |
| <i>Microprocesador</i> | AR-MEGA2560 | 1 | 37,95 |
| <i>Pulsador</i> | 8400010066 | 1 | 0,49 |
| <i>Batería</i> | BLR15 | 1 | 78,95 |
| <i>LM7809CT</i> | 512-LM7809CT | 1 | 0,52 |
| <i>HS115-ND</i> | 507302B00000G | 1 | 0,37 |
| <i>Controlador motor TKXEC XY-15AS</i> | | 4 | 8,32 |
| <i>Kit resistencia de 10 Ω a 1 MΩ</i> | | 1 | 3,30 |
| <i>Led verde</i> | | 1 | 0,17 |
| <i>Led rojo</i> | | 1 | 0,16 |
| <i>Sensor de tensión FZ0430</i> | SEN-0093 | 1 | 1,45 |
| Coste total material | | 813,36 € | |

Tabla 22- Coste de material

El cableado está amortizado, por ello, no se tiene en cuenta en la tabla anterior.

Por consiguiente, el coste total será la suma de todos los costes, el cual se muestra en la siguiente tabla:

| Tipo de coste | Coste total (€) |
|--|-----------------|
| Recursos humanos | 8280 |
| <i>Costes de ejecución del proyecto</i> | 0 |
| <i>Costes de realización de proyecto</i> | 813,36 |
| Coste total proyecto | 9093,36 |

Tabla 23- Coste total

9. DIAGRAMA DE GANTT

| | |
|---------|--|
| Tarea 1 | <p><i>Objetivo:</i> Planificación y estructuración del TFG a través de una reunión con la tutora para tratar la idea.</p> <p><i>Entregable:</i> Comienzo del TFG</p> |
| | <hr/> |
| Tarea 2 | <p><i>Objetivo:</i> Búsqueda viabilidad externa del proyecto para la valoración de su realización</p> <p><i>Entregable:</i> Realización del diagrama de flujo</p> |
| | <hr/> |
| Tarea 3 | <p><i>Objetivo:</i> Inicio de la documentación del prototipo</p> <p><i>Entregable:</i> Realización de los esquemas fundamentales para la elaboración inicial</p> |
| | <hr/> |
| | <p><i>Subtarea 3.1</i></p> <p><i>Objetivo:</i> Visión global del proyecto para su funcionamiento</p> <p><i>Entregable:</i> Realización del diagrama de bloques y esquemas generales.</p> |
| | <hr/> |
| | <p><i>Subtarea 3.2</i></p> <p><i>Objetivo:</i> Análisis de objetivos a alcanzar con sus posibilidades</p> <p><i>Entregable:</i> Elaboración del organigrama</p> |
| | <hr/> |
| | <p><i>Subtarea 3.3</i></p> <p><i>Objetivo:</i> Realización de la documentación del TFG</p> <p><i>Entregable:</i> Entrega del 1º borrador</p> |
| | <hr/> |
| Tarea 4 | <p><i>Objetivo:</i> Obtener la parte mecánica que afecta al proyecto</p> <p><i>Entregable:</i> Análisis de cuáles son los mejores elementos</p> |
| | <hr/> |

| | |
|---------|--|
| | <p><i>Objetivo:</i> Realización de la documentación del TFG</p> <p><i>Entregable:</i> Entrega del 2º borrador</p> |
| Tarea 5 | <p><i>Objetivo:</i> Obtener la parte electrónica que afecta al proyecto</p> <p><i>Entregable:</i> Análisis de los diferentes elementos</p> |
| Tarea 6 | <p><i>Objetivo:</i> Realización del conexionado de los componentes</p> <p><i>Entregable:</i> Viabilidad del proyecto económica</p> |
| Tarea 7 | <p><i>Objetivo:</i> Visión 3D del prototipo</p> <p><i>Entregable:</i> Entrega del 3º borrador</p> |
| Tarea 8 | <p><i>Objetivo:</i> Investigación teórica de los elementos electrónicos seleccionados</p> <p><i>Entregable:</i> Entrega del 4º borrador</p> |
| Tarea 9 | <p><i>Objetivo:</i> Realización del documento TFG</p> <p><i>Entregable:</i> Entrega del 5º borrador</p> |
| | <p><i>Objetivo:</i> Organización de las tareas futuras y realizadas</p> <p><i>Entregable:</i> Comienzo en la realización del diagrama de Gantt</p> |

Objetivo: Realización del documento TFG

Entregable: Entrega del 6º borrador

Tarea 10

Subtarea 10.1

Objetivo: Estudio de la reestructuración del conexionado.

Entregable: Desarrollo y cálculos del conexionado de Arduino mediante un step-down.

Subtarea 10.2

Objetivo: Comprobación del funcionamiento del circuito step-down.

Entregable: Simulación con Proteus del circuito

Tarea 11

Objetivo: Mejora de la estructura del documento

Entregable: Entrega del 7º borrador

Tarea 12

Objetivo: Investigación para el correcto conexionado del motor

Entregable: Entrega del 8º borrador

Tarea 13

Objetivo: Realización del documento TFG

Entregable: Entrega del 9º borrador

Subtarea 13.1

Objetivo: Estudio de las posibilidades del conexionado de Arduino.

Entregable: Sustitución del circuito step-down por un regulador lineal

Tarea 14

Objetivo: Realización del documento TFG

Entregable: Entrega del 10º borrador

| | |
|-----------------|--|
| | <p><i>Objetivo:</i> Conexionado del pulsador y sensor en Proteus</p> <p><i>Entregable:</i> Programación Arduino Mega</p> |
| | <p><i>Objetivo:</i> Comprobación del funcionamiento</p> <p><i>Entregable:</i> Ejecución de la simulación de Arduino en Proteus</p> |
| Tarea 15 | <p><i>Objetivo:</i> Mejora del conexionado de los elementos</p> <p><i>Entregable:</i> Entrega del 11º borrador</p> |
| | <p><i>Objetivo:</i> Realización de los puntos del documento TFG</p> <p><i>Entregable:</i> Entrega del 12º borrador</p> |
| Tarea 16 | <p><i>Objetivo:</i> Realización del resumen y conclusión</p> <p><i>Entregable:</i> Mejora del documento TFG</p> |
| | <p><i>Objetivo:</i> Conexionado alternativo del motor</p> <p><i>Entregable:</i> Reducción de pines de conexión</p> |
| Tarea 17 | <p><i>Objetivo:</i> Realización de la programación</p> <p><i>Entregable:</i> Entrega del 13º borrador</p> |
| | <p><i>Objetivo:</i> Mejora del diagrama de flujo</p> <p><i>Entregable:</i> Finalización documento TFG</p> |

En la Figura 39 se presenta el gráfico del diagrama de Gantt con las tareas realizadas a lo largo del proyecto.

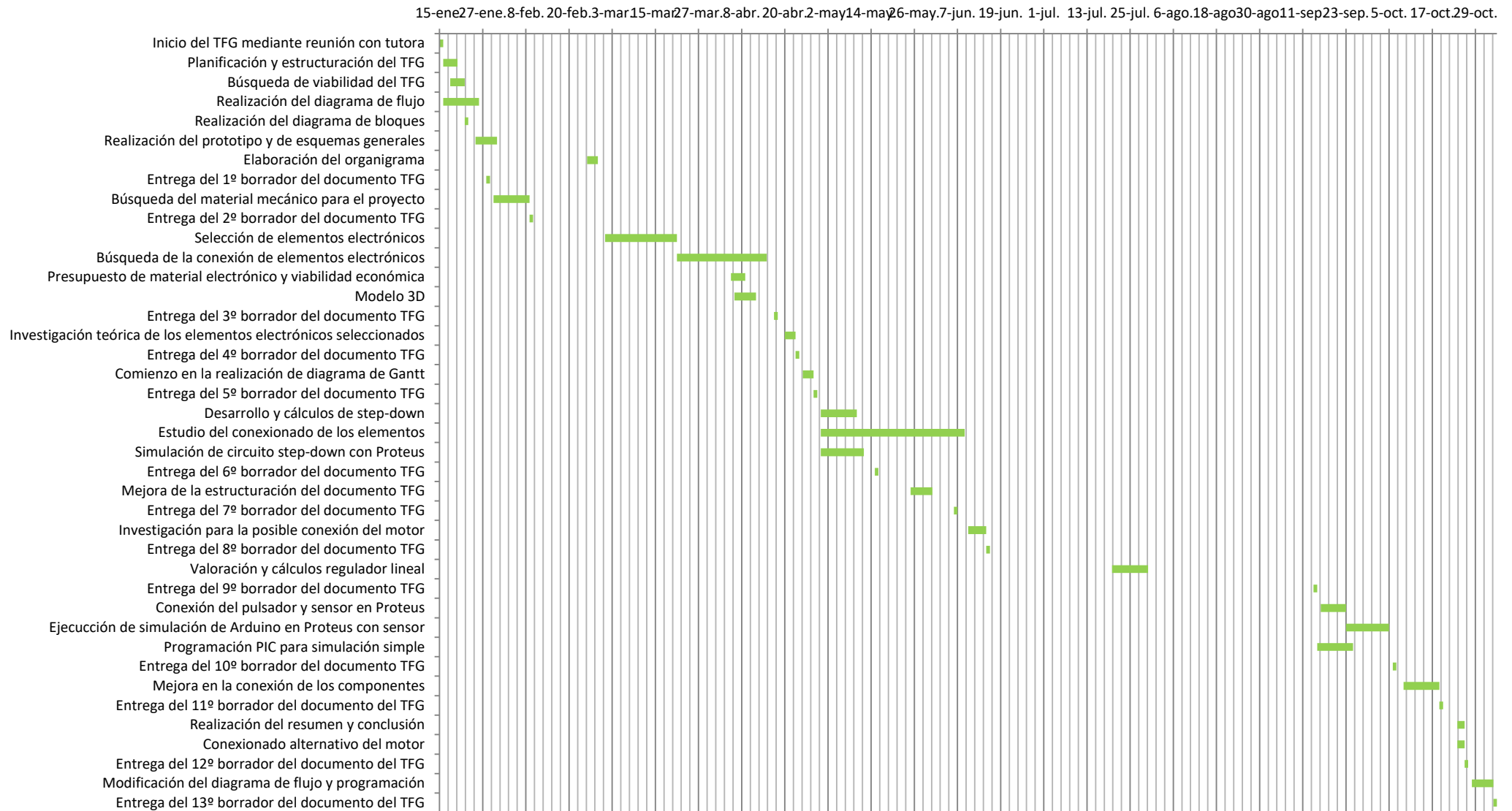


Figura 39- Diagrama de Gantt

10. CONCLUSIONES

El objetivo principal de este proyecto era el diseño, desarrollo y programación de un sistema de control para equilibrar una mesa en función del desnivel del suelo. Para ello, como ha visualizado en el documento, era necesario una mesa equipada con cuatro motores, cuatro sensores y un sistema mecánico de ascenso/descenso, permitiendo así su equilibrio.

El diseño y desarrollo del control para la nivelación automática de la mesa se planteó en primer lugar para realizarlo físicamente. Tras valorar los elementos necesarios para el desarrollo del proyecto y la situación, al no estar físicamente cerca de la Escuela de Ingeniería de Bilbao EHU, se decidió realizar el estudio de los componentes y la programación del microprocesador para finalmente realizar la simulación en Proteus.

La selección de Proteus se debe a la posibilidad de simular la programación de Arduino y observar los posibles errores de ésta. Este programa permitió que sin la realización del proyecto físicamente se pudiera hacer lo más visible posible y valorar el funcionamiento de los elementos electrónicos como los sensores, así como, la comprobación del programa desarrollado en Arduino.

La investigación de los elementos tuvo ciertas complicaciones. El estudio de los sensores a utilizar fue complejo debido a la variedad de sensores de proximidad que existen en el mercado. Además, se tuvo que tener en cuenta el tipo de interferencias que pudiera haber en el entorno en el que se iba a situar la mesa, ya que se trata de un ambiente al aire libre.

Además, los actuadores lineales para el desarrollo del proyecto deben de cumplir ciertas características como son el elevado soporte de peso debido al peso de las bebidas posicionadas en las mesas, y una longitud extensa, ya que la pata de la mesa mide 720 mm y como mínimo el actuador debía tener mayor distancia que éstas soportar sin ningún inconveniente la estimación de 50 kg. Debido a estos dos requisitos, el precio de los actuadores es elevado. Por ello, y debido al ámbito en el que se desea aplicar la mesa autorregulable, en la hostelería, este proyecto no es viable para ser aplicado en

este sector. Sin embargo, tras realizar una búsqueda, se ha observado que este tipo de mesas son utilizadas para espacios de trabajo como oficinas.

Los propósitos personales establecidos en el inicio se han cumplido, ya que se han ampliado conocimientos en electrónica, tanto en el funcionamiento de elementos electrónicos como en las posibilidades existentes en el mercado.

Así mismo, se han reforzado los estudios realizados a lo largo del grado relacionados con la automatización, control, instrumentación y circuitos eléctricos.

11. BIBLIOGRAFÍA

1. Llamas Luis (2016). Guía de selección motores y actuadores para Arduino.
Consultado: 8/03/2020
<https://www.luisllamas.es/guia-seleccion-motores-para-arduino/>
2. González Víctor (2015). Actuadores.
Consultado: 25/03/2020
http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/actuadores.htm#compara_actuadores
3. Llamas Luis (2016). Ejercer grandes fuerzas con actuador lineal y Arduino.
Consultado: 02/04/2020
<https://www.luisllamas.es/arduino-actuador-lineal/>
4. Isaac (2019). VL53L0X: sensor de distancia láser de alta precisión.
Consultado: 15/04/2020
<https://www.hwlibre.com/vl53l0x/>
5. Llamas Luis (2018). Medir distancia con precisión con Arduino y sensor láser VL53L0X y VL6180X.
Consultado: 22/04/2020
<https://www.luisllamas.es/arduino-sensor-distancia-vl53l0x/>
6. Llamas Luis (2015). Medir distancia con Arduino y sensor de ultrasonidos HC-SR04.
Consultado: 29/04/2020
<https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>
7. Prometec (2020). Distancia sensores de distancia infrarrojos.
Consultado: 10/05/2020
<https://www.prometec.net/sensores-distancia-infrarrojos/>
8. Prometec (2020). VL53L0X: Sensor de distancia que mide por la velocidad de la luz (Time-of-Fly).
Consultado: 15/05/2020
<http://robots-argentina.com.ar/didactica/vl53l0x-sensor-de-distancia-que-mide-por-la-velocidad-de-la-luz-time-of-fly/>

-
- 9.** Cinjordiz Cesar (2018). Sensor ultrasonidos HC-SR04.
Consultado: 22/04/2020
<https://www.infootec.net/sensor-ultrasonidos-hc-sr04/>
 - 10.** Colomer Javier (2018). Estudio de los sensores para la detección de obstáculos aplicables a robots móviles.
Consultado: 25/05/2020
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/80846/6/jacobarTFM0618memoria.pdf>
 - 11.** Kevin (2013). Todo lo que tienes que saber sobre: HC-SR04 Sensor Ultrasonico.
Consultado: 05/06/2020
<http://bkargado.blogspot.com/2013/09/todosobrehc-sr04.html>
 - 12.** Crespo Jesús (2020). Alimentación Arduino.
Consultado: 12/06/2020
<https://aprendiendoarduino.wordpress.com/2016/11/09/alimentacion-arduino/>
 - 13.** Villanueva Elena (2019). Tecnología de las baterías.
Consultado: 08/07/2020
<http://www2.elo.utfsm.cl/~elo383/apuntes/PresentacionBaterias.pdf>
 - 14.** Llamas Luis (2014). Leer un pulsador con Arduino.
Consultado: 12/07/2020
<https://www.luisllamas.es/leer-un-pulsador-con-arduino/>
 - 15.** Del Valle Luis (2019). Utilizar pulsadores en Arduino.
Consultado: 15/07/2020
<https://programarfacil.com/blog/utilizar-pulsadores-en-arduino/>
 - 16.** – (2017). Alimentar el Arduino.
Consultado: 15/07/2020
<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/>
 - 17.** Crespo Jesús (2016). Motor DC básico.
Consultado: 18/07/2020
<https://aprendiendoarduino.wordpress.com/tag/motor-dc/>

-
- 18.** Llamas Luis (2016). Controlar grandes cargas con Arduino y transistor mosfet.
Consultado: 22/07/2020
<https://www.luisllamas.es/arduino-transistor-mosfet/>
- 19.** Argos (2015). Convertidores DC-DC buck.
Consultado: 23/07/2020
<https://nomadaselectronicos.wordpress.com/2015/04/12/convertidores-dcdc-buck/>
- 20.** Félix Sánchez (2018). Control de motores con Arduino.
Consultado: 15/09/2020
https://bricolabs.cc/wiki/guias/control_de_motores
- 21.** Luis Llamas (2016). Controlar motores de corriente continua con Arduino y L298N.
Consultado: 20/09/2020
<https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>
- 22.** Juan Antonio Villalpando (2019). Modulación por PWM.
Consultado: 20/09/2020
<http://kio4.com/arduino/6pwm.htm>