

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

FACULTAD DE INGENIERÍA ELÉCTRICA



**CONTROL PARA CAMAS ELÉCTRICAS HOSPITALARIAS CON SMARTPHONE
POR BLUETOOTH.**

TESIS

Tesis para optar el Título de **Ingeniero en Electrónica**, que presenta:

Josue Jozic Zetina Martinez

Asesor

Mc. Samuel Pérez Aguilar

Morelia Mich, Agosto 2019

DEDICATORIA

La presente tesis se la dedico a mis padres, hermanos, suegros, esposa y a mi hijo Yaser, que con mucho cariño les dedico todo mi esfuerzo y dedicación puestos para la realización de esta tesis, ya que me dieron su gran apoyo incondicional en cada momento y depositaron su entera confianza y esperanza en mí. Sin ellos, no lo hubiese podido llegar hasta esta nueva etapa. ¡GRACIAS!

RESUMEN

En esta tesis se presenta la documentación, diseño e implementación de software y hardware para el control inalámbrico de camas eléctricas hospitalarias, por medio de la tecnología bluetooth desde un dispositivo smartphone. Se muestran los fundamentos de los temas para el desarrollo de la aplicación y la tarjeta de control, así como las ventajas, desventajas y actualizaciones que presente este proyecto a futuro.

El propósito de este sistema es actualizar el hardware y una aplicación para el control de camas eléctricas hospitalarias, con los siguientes beneficios:

- Mayor comodidad y movimiento para el paciente.
- Control de nivel de iluminación.
- Diseño adaptado a diferentes dispositivos y tamaños de pantallas.
- Integración con las redes sociales.
- Control de la cama eléctrica.
- Posibilidad que el paciente seleccione las melodías que desea escuchar.
- Control inalámbrico con la tecnología Bluetooth.
- Sistema de compra o renta de la cama eléctrica por medio de la aplicación.
- Aislamiento eléctrico.
- Menor consumo de energía eléctrica.
- Monitoreo de la cama.
- Botón de RCP y alarma.

Además, el sistema está enfocado para el ahorro de energía y reducir costo de reparación, por lo que se han considerado componentes y sensores de bajo consumo de potencia para la longevidad de todos los componentes electrónicos.

ABSTRACT

In this thesis presents the documentation, design and implementation of software and hardware for the wireless control of hospital electric beds, by means of Bluetooth technology from our smartphone device. The fundamentals of the topics for the development of the application and the control card are shown, as well as the advantages, disadvantages and updates that this project presents in the future.

The purpose of this system is to update the hardware that this presented by the hospital electric beds, also it will bring the next benefits:

- Greater comfort and movement.
- Dimmer for LED light.
- Easy use of the application.
- Technical service and customer service.
- Total control of the electric bed.
- Music specially selected for the patient.
- Wireless control with Bluetooth technology.
- System of buy or sale of the electric bed by means of the application.
- Electrical insulation.
- Easy hardware maintenance and repair.
- Less electricity consumption.
- Monitoring of the bed.
- CPR button and alarm.

In addition, it's important mentioning that the system is focused on energy saving and its low cost, so it has been considered essentially low power consumption components and sensors for the longevity of all electronics devices.

PALABRAS CLAVE

Arduino Nano, comunicación serial, UART, Sistemas autómatas, Sistemas digitales, MP3, Bluetooth, lenguaje C, APK (Android Application Package), IDE (Entorno de Desarrollo Integrado), resucitación cardiopulmonar (CPR).

ÍNDICE GENERAL

DEDICATORIA.....	3
RESUMEN.....	4
ABSTRACT.....	5
PALABRAS CLAVE.....	6
ÍNDICE GENERAL	7
ÍNDICE DE FIGURAS	10
ÍNDICE DE TABLAS	11
CAPÍTULO I. INTRODUCCIÓN.....	13
1.1. Introducción.....	13
1.2. Reseña Histórica	14
1.3. Características de la cama eléctrica	16
1.4. Objetivos.....	17
1.5. Motivaciones	18
1.6. Justificación	18
1.7. Descripción de capítulos.....	19
CAPÍTULO II. MARCO TEÓRICO	21
2.1. La cama eléctrica hospitalaria.....	21
2.2. Posiciones de la cama	22
2.3. Descripción y ventajas sobre las posiciones de la cama	25
2.4. Normas para camas eléctricas hospitalarias	28
2.5. Liberación de CPR	30
2.6. Los micro interruptores de paro	31
2.7. El motor para el movimiento de las articulaciones.....	33
CAPÍTULO III. HERRAMIENTAS DE HARDWARE Y SOFTWARE	37
3.1. Arduino	37
3.1.1. Historia del Arduino	38
3.1.2. Características técnicas.....	39
3.1.3. Alimentación de la placa Arduino Nano	40
3.1.4. Pinout Arduino Nano.....	41
3.1.5. Entradas y salidas	42

3.2. Lenguaje de programación Arduino	43
3.2.1. Variables y operadores	44
3.3. IDE Arduino.....	47
3.3.1. Estructura de un Sketch	49
3.4. Módulo Bluetooth HC-05	51
3.4.1. Características.....	53
3.4.2. Funcionamiento.....	54
3.5. Dfplayer TF-16P	56
3.5.1. Características.....	56
3.5.2. Aplicaciones.....	58
3.5.3. Pinout del módulo Dfplayer TF-16P	58
3.5.4. Funcionamiento.....	59
3.6. Tecnologías de Software utilizadas	60
CAPÍTULO IV. CONSTRUCCIÓN DEL MODELO DEL SISTEMA.....	61
4.1. Introducción.....	61
4.2. Modelo del sistema.....	61
4.2.1. Materiales.....	62
4.2.2. Fuente de poder	63
4.2.3. Entradas	64
4.2.4. Proceso.....	65
4.2.5. Actuadores	65
4.2.6. Conexión del módulo Bluetooth HC - 05	65
4.2.7. Conexión del módulo Dfplayer.....	67
4.2.8. Conexión del amplificador de audio LM386.....	67
4.2.9. Conexión del sistema de alarma y la intensidad de luz	69
4.2.10. Conexión del sistema de Motores de AC	70
4.3. Software de la aplicación	71
4.3.1. Interfaz para la activación del producto	72
4.3.2. Interfaz de bienvenida.....	73
4.3.3. Interfaz de inicio	74
4.3.4. Interfaz del reproductor de audio	76
4.3.5. Interfaz de contacto.....	77
4.3.6. Interfaz de cama	78

CAPÍTULO V. CONCLUSIONES Y TRABAJOS FUTUROS.....	79
5.1. Conclusiones.....	79
5.2. Trabajos futuros	80
REFERENCIAS	81
ANEXOS.....	83
ANEXO A: Código de desarrollo de la Placa Arduino.....	83
ANEXO B: PCB del proyecto (Cooper Botton).....	96
ANEXO C: PCB del proyecto (Cooper Top).	97
ANEXO D: Diagrama electrónico.	98

ÍNDICE DE FIGURAS

Figura 1.1. Cama eléctrica hospitalaria Advanced Hill-Rom Bed.....	13
Figura 1.2. Cama con respaldo.....	14
Figura 1.3. Cama en forma de caja de madera.....	15
Figura 1.4. Cama confortable de máxima suavidad del siglo XXI.....	16
Figura 1.5. Características de una cama eléctrica hospitalaria.....	17
Figura 2.1. Posiciones de las articulaciones.....	22
Figura 2.2. Posición de altura máxima y mínima.....	23
Figura 2.3. Posición de silla.....	24
Figura 2.4. Posición de Trendelemburg.....	25
Figura 2.5. Botón de CPR para posicionamiento horizontal.....	31
Figura 2.6. Micro switch acoplado en un riel con desplazamiento alto y bajo.....	32
Figura 2.7. Motor de corriente alterna con capacitor de arranque.....	33
Figura 2.8. Motor de corriente alterna sin capacitor de arranque.....	34
Figura 2.9. Motores para las articulaciones de la cabeza y elevación de la cama.....	35
Figura 2.10. Motores para las articulaciones de los pies.....	36
Figura 3.1. Placa Arduino nano.....	38
Figura 3.2. Diagrama Pinout de la placa Arduino nano.....	41
Figura 3.3. IDE de Arduino.....	48
Figura 3.4. Sketch de Arduino para el encendido y apagado de un led.....	50
Figura 3.5. Elección de la placa y del puerto de conexión para Arduino nano.....	51
Figura 3.6. Módulo HC-05.....	52
Figura 3.7. Módulo Dfplayer TF-16P.....	58
Figura 4.1. Diagrama de bloques del software y hardware empleado.....	62
Figura 4.2. Diagrama electrónico de la fuente de poder.....	63
Figura 4.3. Botones para el movimiento de las articulaciones.....	64
Figura 4.4. Conexión del módulo Bluetooth HC -05 con Arduino.....	66
Figura 4.5. Conexión del módulo Dfplayer TF-16P con Arduino nano.....	67
Figura 4.6. Conexión del sistema de audio.....	68
Figura 4.7. Conexión del sistema de alarma e iluminación.....	69
Figura 4.8. Conexión del sistema a un motor de AC.....	70
Figura 4.9. Editor de bloques de App Inventor 2.....	72
Figura 4.10. Pantalla para activación del producto.....	73
Figura 4.11. Pantalla de bienvenida.....	74
Figura 4.12. Pantalla de inicio.....	75
Figura 4.13. Interfaz de audio.....	76
Figura 4.14. Pantalla de contacto.....	77
Figura 4.15. Pantalla de posiciones.....	78

ÍNDICE DE TABLAS

Tabla 2.1. Normas para el uso de camas eléctricas hospitalarias.....	29
Tabla 3.1. Características técnicas de la placa Arduino nano.....	40
Tabla 3.2. Tipos de variables en Arduino.....	45
Tabla 3.3. Tipos de operadores matemáticos en Arduino.....	46
Tabla 3.4. Tipos de operadores compuestos en Arduino.....	46
Tabla 3.5. Jerarquía de operadores en Arduino.	47
Tabla 3.6. Características técnicas del módulo HC-05.....	53
Tabla 3.7. Versiones de los módulos Bluetooth.....	54
Tabla 3.8. Características técnicas del módulo Dfplayer TF-16P.....	57
Tabla 3.9. Aplicaciones más utilizadas para del módulo Dfplayer TF-16P.....	58
Tabla 3.10. Descripción del Pinout del módulo TF-16P.....	59
Tabla 3.11. Comandos para el módulo Dfplayer TF-16P.....	60
Tabla 4.1. Material utilizado para el diseño del sistema.	62
Tabla 4.2. Configuración personalizada para el módulo bluetooth.....	66
Tabla 4.3. Configuración de los pines de entrada y salida utilizados para el control de los motores de corriente alterna.	71

CAPÍTULO I. INTRODUCCIÓN

1.1. Introducción

En todo hospital la disponibilidad y oferta del servicio de camas eléctricas es vital ya que contribuyen con la pronta recuperación del paciente, facilitando a los doctores y enfermeras la movilidad y el posicionamiento del mismo, de modo que utilizar estas camas traerá las siguientes ventajas para el paciente:

- Aumento de la movilidad.
- Mayor independencia.
- Apoyo durante las comidas.
- Facilidad para cambiar de lugar dentro de la casa u hospital.
- Mejora la circulación sanguínea.
- Menor esfuerzo.
- Seguridad.

Por ello existen en el mercado una gran variedad de camas eléctricas hospitalarias para todo tipo de necesidades. En este documento nos enfocamos en la actualización del hardware de la cama eléctrica hospitalaria Hill-Rom Advance Bed, que se muestra en la Figura 1.1. [1]



Figura 1.1. Cama eléctrica hospitalaria Advanced Hill-Rom Bed. [1]

1.2. Reseña Histórica

La cama posee la misma forma desde el antiguo Egipto que consistía en un bastidor rectangular alargado de madera o metal, que se muestra en la Figura 1.2. Ya tenía respaldo decorado con figuras. Sin embargo, no solo se utilizaba para dormir sino también para comer. El cabecero no llegó hasta la Grecia clásica. [2]



Figura 1.2. Cama con respaldo. [2]

Fueron los romanos quienes empezaron a utilizar ropa de cama. Se cubrían de sábanas de lino y dormían sobre una especie de alfombra hecha de trenzas de lana o pelo. Podría decirse que durante esta época se crearon los primeros colchones, que consistían en un simple saco de paja. Los más acaudalados los rellenaban de lana y plumas. [2]

En la Edad Media las camas eran, sencillamente, cajas de madera que más tarde cubrieron con colchones como se muestra en la Figura 1.3.



Figura 1.3. Cama en forma de caja de madera. [2]

Como es de esperar cada rango social disponía de una ropa de cama de mejor o menor calidad. Los más ricos dormían sobre colchones y sábanas fabricadas con lino rellenos de plumón de ganso o pato en camas hechas con madera. Mientras que los más pobres dormían sobre literas realizadas con heno o sobre colchones rellenos de paja colocados en el suelo. [2]

Básicamente, hasta la primera Revolución Industrial, el lino era el material primordial de las sábanas.

Gracias a los avances técnicos en la industria textil se empieza a comercializar la ropa de cama de algodón, mucho más barata y práctica. Asimismo, también se usaba para los colchones, lo que los hacía mucho más cómodos. Durante esta época se introdujeron las fundas de edredones y los nórdicos rellenos de plumas. [2]

A partir del siglo XX las fibras sintéticas toman el poder y ocasionan una cadena de producción colosal que se expande de manera global hasta el día de hoy como se muestra en la Figura 1.4. [2]



Figura 1.4. Cama confortable de máxima suavidad del siglo XXI. [2]

1.3. Características de la cama eléctrica

La cama es el lugar donde el paciente pasa la mayor parte del tiempo durante la estancia hospitalaria, es un elemento muy importante a la hora de la prevención y recuperación del estado de salud de las personas ingresadas.

Es también el lugar donde ocurren otras dolencias que pueden retrasar el alta del paciente como: las caídas, las infecciones, etc.... Puede evitarse con un adecuado manejo e higiene de la cama. Entre las características que hacen una cama hospitalaria diferente a otro tipo de cama son:

- El colchón ha de ser más semirrígido para soportar adecuadamente el peso del paciente sin llegar a comprometer la integridad de la piel y la circulación, originando nuevos problemas que se añadan a las patologías ya presentes.
- Son de muelles y de una sola pieza.
- Están compuestas de tubos huecos que se pueden lavar fácilmente.
- Llevan ruedas móviles en cada pata (cuatro) y un sistema de freno que permite su bloqueo, lo que facilita el desplazamiento y seguridad del paciente.

- El somier puede estar dividido en una, dos o tres partes (segmentos) que pueden ser cambiados de posición en el hospital manualmente mediante una manivela o con el control remoto.
- En cuanto a las dimensiones de las camas son: 90 – 105 cm de ancho y 190 – 200 cm de largo.
- La ropa de cama ha de ser amplia y resistente a lavados frecuentes, pero sin llega a ser áspera. El objetivo es sujetarla correctamente al colchón. [3]

En la Figura 1.5 se muestran las características de las camas eléctricas hospitalarias importantes a considerar.



Figura 1.5. Características de una cama eléctrica hospitalaria. [3]

1.4. Objetivos

Diseñar una tarjeta electrónica para el control de camas eléctricas hospitalarias por medio de la tecnología bluetooth para smartphone con sistema operativo Android.

- 1) Objetivo 1 Proveer al paciente una mejor comodidad controlando la cama eléctrica desde su dispositivo smartphone.
- 2) Objetivo 2 Actualizar el hardware de la cama eléctrica hospitalaria.
- 3) Objetivo 3 Diseñar una aplicación Android.

1.5. Motivaciones

Debido a que el smartphone hoy en día es un dispositivo electrónico muy utilizado, se vio la oportunidad de diseñar una aplicación para el control inalámbrico de camas eléctricas hospitalarias y monitoreo de tiempo desde nuestro dispositivo inteligente con sistema operativo Android.

Además de poder diseñar una tarjeta electrónica, con todo el procedimiento que implica de diseño para ponerla en función.

1.6. Justificación

Este proyecto de investigación contribuye a la innovación y tecnología de camas eléctricas hospitalarias. Esto puede ser útil para los hospitales y casa habitación, debido a que las camas eléctricas permiten al paciente una pronta mejora de su estado físico. Debido a la importancia de uso de camas eléctricas en el área médica, este proyecto busca proveer al paciente un mejor confort y control de la cama eléctrica mediante comunicación inalámbrica por medio de la tecnología Bluetooth desde su dispositivo smartphone con sistema operativo Android 3.0 en adelante.

1.7. Descripción de capítulos

En el Capítulo 1 se presenta una breve reseña de la importancia del uso de las camas eléctricas hospitalarias y como estas han cambiado en el tiempo para cubrir las necesidades del paciente. Se dan a conocer los objetivos, motivación y justificación para realizar este proyecto.

En el Capítulo 2 se describe la cama eléctrica hospitalaria Advanced 2000 Hill-Rom Bed, también las posiciones que puede optar por medio de sus articulaciones y sus beneficios que traen hacia el paciente, además se describen las normas para la seguridad y cuidado del paciente. Por último, en este tema se describe el motor especial de corriente alterna de la marca Hill-Rom Bed para el control de las articulaciones.

En el Capítulo 3 se describen todas las herramientas de hardware y software que se utilizaron para el diseño del control inalámbrico; se relata una breve historia de Arduino, así como la descripción de sus características técnicas y como utilizar el IDE (Entorno de Desarrollo Integrado) de Arduino y por último se describen los módulos empleados para este sistema tales como: El Módulo Bluetooth HC-05, Modulo Dfplayer TF-16P y el microcontrolador Atmega 328p.

En el Capítulo 4 se describe el software y hardware por medio de dos diagramas a bloques, analizando los aspectos del diseño tales como: la lista de materiales de componentes electrónicos para su diseño, las conexiones de cada módulo, el diseño y programación del software para el smartphone con sistema operativo Android y todas sus interfases para el usuario.

En el Capítulo 5 se dan las conclusiones del proyecto y algunos trabajos futuros que pueden realizarse tanto para el hardware como al software.

CAPÍTULO II. MARCO TEÓRICO

2.1. La cama eléctrica hospitalaria

Una cama eléctrica es un tipo de cama articulada que incorpora al menos un motor eléctrico para conferirle movimiento. La cama eléctrica es un mueble que se utiliza para conseguir diferentes posiciones de descanso sin necesidad de manipularla de forma manual como se muestra en la Figura 2.1. Es un modelo habitual en los hospitales que se usa para colocar a los enfermos en las posiciones tanto para comer como para ver la televisión o conversar. [4]

En primer lugar, si hablamos de las camas eléctricas, hemos de señalar que añaden, a las articuladas comunes, el accionamiento mediante motores como se muestra en la figura 2.7. Estos motores sirven para colocar la cama en la posición más conveniente, ya que facilita los distintos movimientos entre sus tramos de articulación (generalmente, cuatro o cinco). De esta manera, se facilita la higiene postural idónea para cada necesidad del usuario.

Además, se puede controlar mediante un mando a distancia, lo que supone una gran comodidad y funcionalidad, este tema se describirá en el capítulo IV. Por ejemplo, será posible adoptar la postura más conveniente para evitar las úlceras por presión permitiendo el descanso de las piernas. Asimismo, también se pueden elegir las posiciones que contribuyan a comer, leer o ver la televisión más cómodamente, lo cual vale tanto para las personas con movilidad reducida como para las que no tienen problemas de salud.

La relajación muscular y los cambios de posturas ayudarán a mejorar la circulación sanguínea. Sin duda, este bienestar se traduce, a medio plazo, en una mejoría del sistema inmunológico. [4]

2.2. Posiciones de la cama

La cama eléctrica hospitalaria Advanced 2000 Hill-Rom Bed, consta con cuatro posiciones: secciones de cabeza, asiento, rodilla y pie. Las posiciones de la cama se muestran en la Figura 2.1.

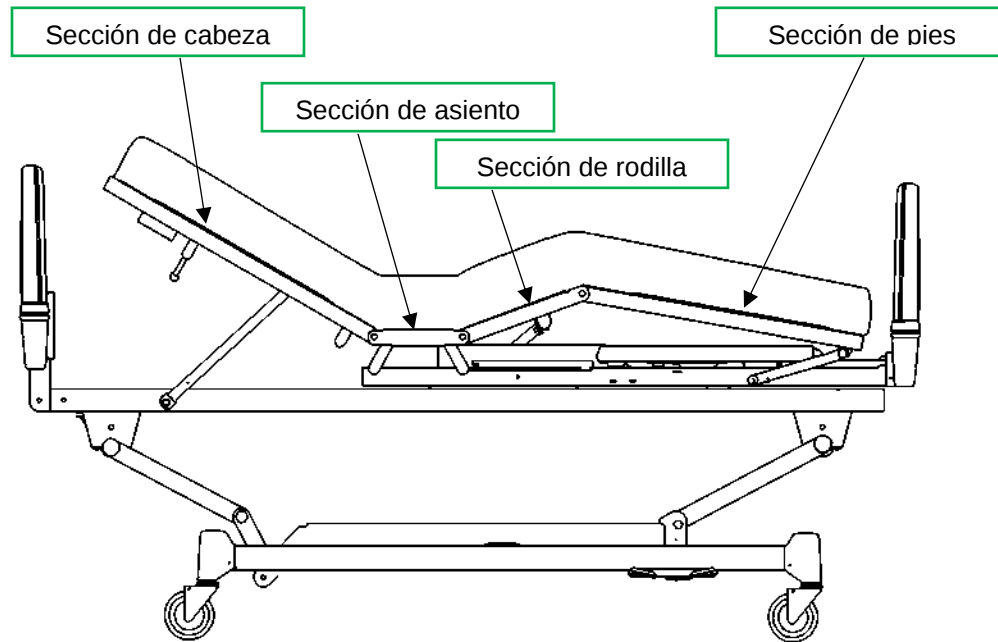


Figura 2.1. Posiciones de las articulaciones. [5]

A continuación, se mostrarán todas las posiciones de la cama Advanced 2000 Hill-Rom Bed, que puede articular por medio de 4 motores, los cuales giran en sentido horario y anti horario para subir o bajar las posiciones de las articulaciones de la cama y por medio de rieles en los extremos de estos están instalado dos micros switch para detener el giro del motor. Más adelante se hablará sobre los motores y el micro switch de paro.

En la Figura 2.2 se muestra la altura máxima (a) que puede alcanzar esta cama y también la altura mínima (b).

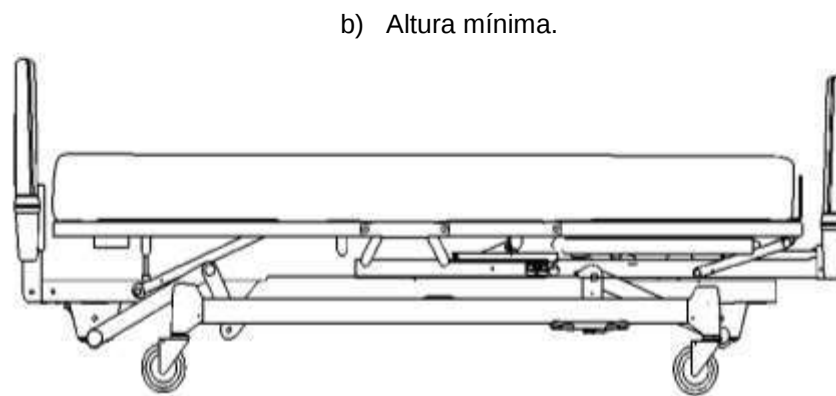
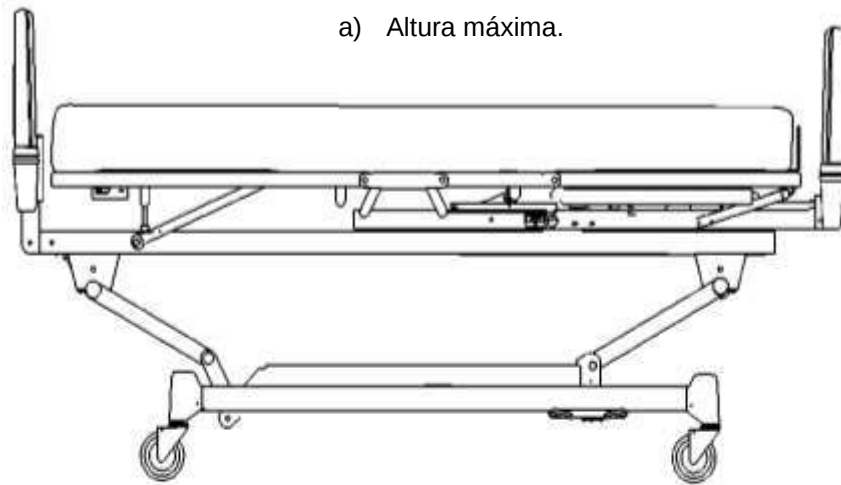
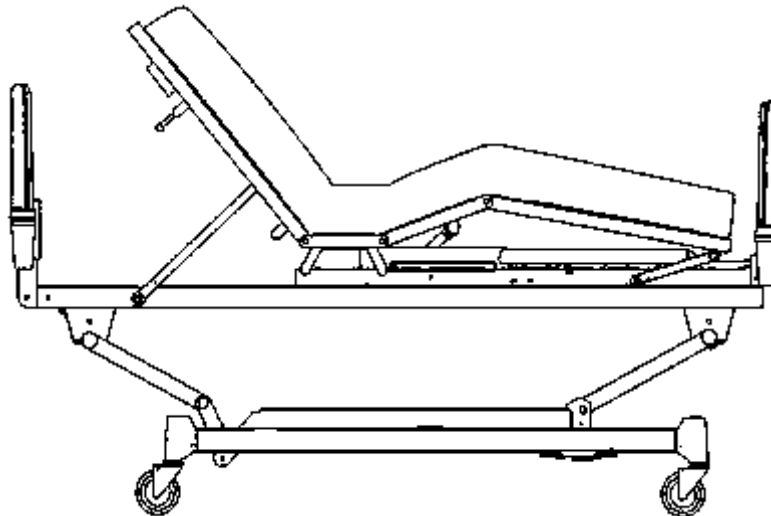


Figura 2.2. Posición de altura máxima y mínima. [5]

El control de esta posición es útil para situaciones donde se necesita ingresar un nuevo paciente debido a que podemos subir o bajar la cama. En esta posición por lo común ayuda al paciente a estar cómodo, cuando las rodillas están flexionadas para aflojar la pared abdominal.

En la Figura 2.3 se muestra la posición de silla (a) y posición silla con inclinación (b).

a) Posición de silla.



b) Posición de silla con inclinación.

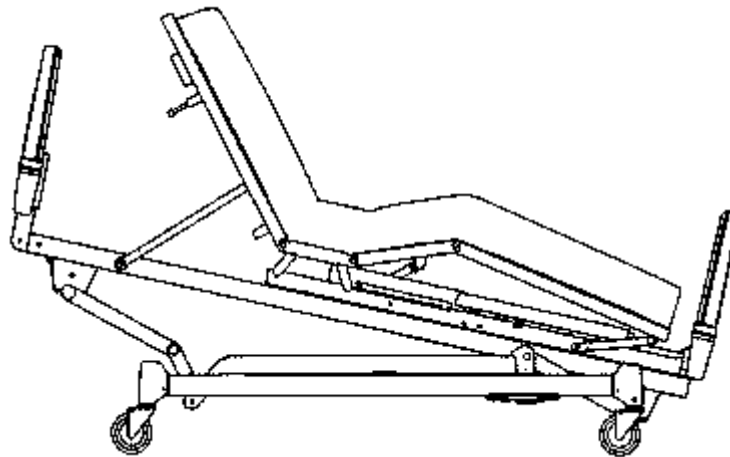
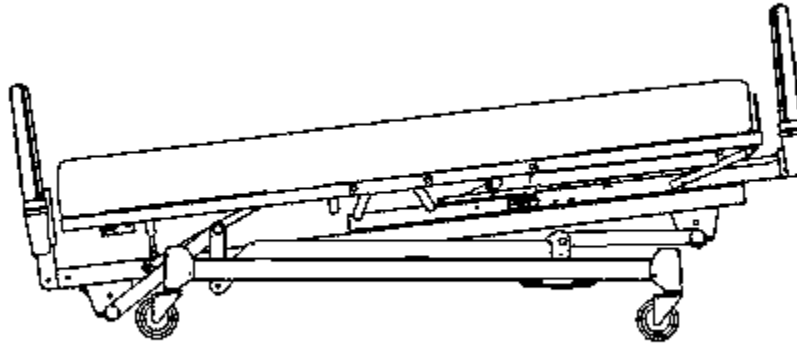


Figura 2.3. Posición de silla. [5]

El control de esta posición nos permite sentar al paciente para poder tomar sus alimentos o para descansar la espalda y evitar llagas.

En la Figura 2.4 se muestra la posición de Trendelenburg (a) y la posición de Trendelenburg inverso (b).

a) Posición de Trendelemburg.



b) Posición de Trendelemburg inverso.

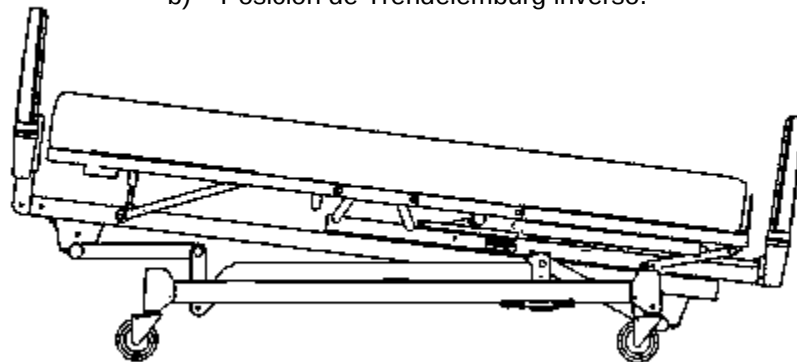


Figura 2.4. Posición de Trendelemburg. [5]

El control de esta posición permite que se facilite, por el efecto de la gravedad, el retorno de un gran volumen de sangre desde el sistema de la vena cava inferior al corazón. [6]

2.3. Descripción y ventajas sobre las posiciones de la cama

La cama eléctrica hospitalaria Hill-Rom Advance Bed presenta múltiples posiciones por medio de sus articulaciones y a su vez tiene barandales

que permiten al paciente poder sujetarse y mantenerlo dentro de la cama, está diseñada con mecanismo eléctrico para ajustar la altura por medio de sus posiciones articuladas. Es una cama con sistema eléctrico para su operación y por ello es una de las mejores camas eléctricas hospitalarias en el mercado por su durabilidad, permitiendo realizar las siguientes posiciones:

Posición horizontal o decúbito supino

Consiste en bajar el área de los pies y la cabeza de modo que el paciente sea tumbado sobre la espalda con los brazos y las piernas en extensión y cerca del cuerpo como se muestra en la Figura 2.2. El paciente se encuentra tendido boca arriba, en posición horizontal, siendo el eje del cuerpo paralelo al suelo. Es la posición más común que adopta el paciente en la cama y también es la más frecuente en operaciones quirúrgicas. Permite una expansión pulmonar y facilita la alineación de los distintos segmentos corporales. Posición adecuada para el examen del tórax, abdomen, miembros superiores e inferiores, postoperatorios, estancia en cama y cambios posturales, para la realización de R.C.P. (Reanimación cardiopulmonar). [7]

Posición semi sentado o flowler

El paciente se encuentra semisentado, con el cabecero de la cama elevado 45° y las rodillas semiflexionadas. Existen variantes de la posición Fowler: a) Semi-Fowler: la elevación del cabecero es de 30°; b) Fowler-alta: la elevación del cabecero es de 90°, como se muestra en la Figura 2.3. Colocando una almohada en la espalda apoyando la zona lumbar, otra en la cabeza y hombros, otra pequeña bajo los muslos y otra bajo los tobillos. Es la posición adecuada para pacientes con problemas respiratorios como: asma, EPOC (Enfermedad pulmonar obstructiva crónica), etc. y cardíacos, ya que permite la expansión máxima del tórax y un mejor aporte de aire a los pulmones. También para pacientes con hernia de hiato, cambios posturales, para dar de comer al paciente que no

puede hacerlo por sí mismos; para realizar exploraciones de cabeza, cuello, ojos, oídos, nariz, garganta y pecho, favorecer el drenaje después de operaciones abdominales, administración de oxígeno. [7]

Trendelemburg

El paciente se encuentra en decúbito supino inclinado 45° respecto al plano del suelo, con la cabeza más baja que los pies como se muestra en la Figura 2.4. Posición indicada para intervenciones de la zona inferior del abdomen o de la pelvis; cirugía de vejiga y colon; exploraciones radiográficas; pacientes con problemas respiratorios; facilita el drenaje de secreciones bronquiales. Es la posición correcta para trasladar una embarazada con hemorragia vaginal. Se le llama la Posición Anti-Shock, ya que mejora la circulación cerebral, es la posición más idónea para lipotimias o síncope, conmoción o shock, hipotensión arterial severa. [7]

Morestin o Antitrendelemburg

Posición contraria a la de Trendelemburg, el paciente se encuentra en posición de decúbito supino con el plano inclinado 45°, estando la cabeza más alta que los pies. También llamada Trendelemburg Inversa (b) como se muestra en la Figura 2.4. Es la posición más adecuada para intervenciones de cuello (tiroides), cara y cráneo con el fin de disminuir el riego sanguíneo y evitar el estancamiento de sangre en la zona que se está interviniendo. Se utiliza también para cirugía de la zona del abdomen y del diafragma. [7]

2.4. Normas para camas eléctricas hospitalarias

Para iniciar con este subtema es necesario comprender el concepto de norma e I.S.O (Organización Internacional para la Estandarización) debido a su gran importancia de seguridad y cuidado del paciente.

Las siglas I.S.O representan a la Organización Internacional para la Estandarización; organismo responsable de regular un conjunto de normas para la fabricación, comercio y comunicación en todas las industrias y comercios del mundo. Esta organización surge en 1947, una vez finalizada la segunda guerra mundial, transformándose en una entidad dedicada a fomentar la creación de normas y regulaciones de carácter internacional para la elaboración de todos los productos, a excepción de aquellos pertenecientes al área de la electrónica y la electricidad. De esta manera se garantiza calidad en todos los productos aunado al respeto por las políticas de protección ambiental.

El objetivo de estas normas es poder estandarizar el proceso productivo de los productos que se fabriquen en cada uno de los países a nivel mundial, de tal manera que cada uno pueda compararse entre sí.

Para las industrias, el poder contar con una certificación ISO, le permite mostrar que están acatando con lo establecido por los acuerdos contractuales, mientras que para los consumidores esta certificación, les permite reconocer cuales son los productos o servicios que cumplen con los requisitos y cuáles son los proveedores más confiables. [8]

La Tabla 2.1 nos muestra las normas FDA, CE o JIS. ISO 13485 para camas eléctricas hospitalarias de origen extranjero y nacional:

Tabla 2.1. Normas para el uso de camas eléctricas hospitalarias. [9]

CAMA HOSPITALARIA ELÉCTRICA DE MÚLTIPLES POSICIONES		
Descripción	1.- Cama hospitalaria eléctrica de múltiples posiciones.	
	2.- Que soporte un peso de 200 Kg como mínimo.	
	3.- Controles eléctricos inter construidos o integrados en barandales, con seguro de bloqueo para paciente.	
	4.- Que permita dar las siguientes posiciones en forma eléctrica:	4.1.- Trendelenburg de 12° como mínimo.
		4.2.- Trendelenburg inverso de 12° como mínimo.
		4.3.- Sección de espalda o fowler con auto contorno, que cubra el rango de 0 a 60° como mínimo.
		4.4.- Sección de rodilla que cubra el rango de 0 – 23° como mínimo.
		4.5.- Altura y descenso ajustable que cubra el rango de 47 cm. A 74 cm. (medido de la plataforma de la cama al piso, sin colchón), como mínimo.
		4.6.- Posición de reanimación cardio pulmonar (RCP).
		4.7.- Posición vascular o elevación de pies.
		4.8.- posición de silla.
	5.- Sistema de bloqueo de los movimientos eléctricos.	
	6.- Indicador del ángulo de la cabeza.	
	7.- Superficie de la cama rígida.	
	8.- Dimensiones de la superficie del paciente 203 cm. De largo x 88 cm. De ancho, como mínimo.	
	9.- Dimensiones de la cama $\pm$ 5%: longitud total 230 cm. Ancho total 105 cm.	
	10.- Cabecera y piecero desmontables de material de alta resistencia a golpes y solventes.	
	11.- Barandales laterales abatibles que permitan la transferencia segura del paciente, dos barandales en sección de cabeza y dos en sección de pies.	
	12.- Con protectores o parachoques en las cuatro esquinas.	
	13.- Colchón de poliuretano de 15 cm. ($\pm$ 5%) de espesor, de alta densidad con diseño para reducción de presión,	

	antiestático, recubrimiento de material lavable, repelente a líquidos, retardante al fuego y con funda removible para lavado.
	14.- Con ruedas antiestáticas o conductivas de 12.5 cm como mínimo.
	15.- Ganchos para bolsas de soluciones en ambos lados de la cama.
	16.- Que cuente con batería de seguridad en caso de falla eléctrica para todos los movimientos.
	17.- Que los barandales puedan expandirse o contraerse, por medio de una palanca con seguro.
INSTALACIÓN:	Corriente eléctrica 120V/60 Hz.
OPERACIÓN:	Por personal especializado y de acuerdo al manual de operación.
MANTENIMIENTO:	Preventivo y correctivo por personal calificado.
NORMAS – CERTIFICADOS:	Para producto de origen nacional: certificado de buenas prácticas de fabricación expedido por la COFEPRIS. ISO 13485.
	Para producto de origen extranjero: que cumpla con alguna de las siguientes normas: FDA, CE o JIS. ISO 13485.

2.5. Liberación de CPR

La reanimación cardiopulmonar (CPR) se administra cuando la respiración o pulso de alguien se detiene. Si ambos se detienen, entonces ha ocurrido muerte súbita.

A pesar de que algunas de las causas de muerte repentina incluyen envenenamiento, ahogamiento, asfixia, sofocación, electrocución o inhalación de humo, la causa más común es de un ataque al corazón por ello es que la cama eléctrica consta con este sistema de emergencia para situaciones donde la vida del paciente corre peligro.

La cama eléctrica consta con un botón de liberación de CPR que se encuentra en la sección de la cabeza, por lo que está mecanizada para bajar de emergencia rápidamente posicionando a la cama horizontalmente

en poco tiempo. El mecanismo de liberación requiere ser presionado para la liberación. El mecanismo de la cama automáticamente baja la sección de la rodilla y de la cabeza para poder realizar la resucitación cardiovascular en caso de una emergencia.

En la figura 2.5 se muestran la ubicación de los botones para activar el posicionamiento horizontal de la cama.



Figura 2.5. Botón de CPR para posicionamiento horizontal. [5]

2.6. Los micro interruptores de paro

El micro switch o micro interruptor. Es un interruptor eléctrico que se acciona por muy poca fuerza física, a través del uso de un mecanismo de punto de inflexión, a veces llamado un mecanismo de "sobre centro".

Son muy comunes debido a su bajo costo y durabilidad, mayor que 1 millón de ciclos y hasta 10 millones de ciclos para los modelos de servicio pesado. Esta durabilidad es una consecuencia natural del diseño.

La característica definitoria de micro interruptores es que un movimiento relativamente pequeño en el botón de accionamiento produce un movimiento relativamente grande en los contactos eléctricos, que se produce a alta velocidad (independientemente de la velocidad de accionamiento).

Por medio de un riel donde la articulación puede desplazarse en ambos sentidos debido al motor, se encuentran dos micro switch respectivamente como se muestra en la Figura 2.6(C), permitiendo así poder interrumpir la señal de activación del motor, ya que esta señal entra al sistema de lógica binaria haciendo que el microcontrolador desactive o active (dependiendo del caso) los relevadores que energizan o apagan al motor. Esta misma configuración es para los motores que activan la articulación de los pies, cabeza, elevación de la cama y trendelemburg.

En la Figura 2.6 se muestra la ubicación de los micro interruptores que permiten la detección del límite superior e inferior para el paro de los motores de las articulaciones de los pies, cabeza, elevación de la cama y trendelemburg. [10]

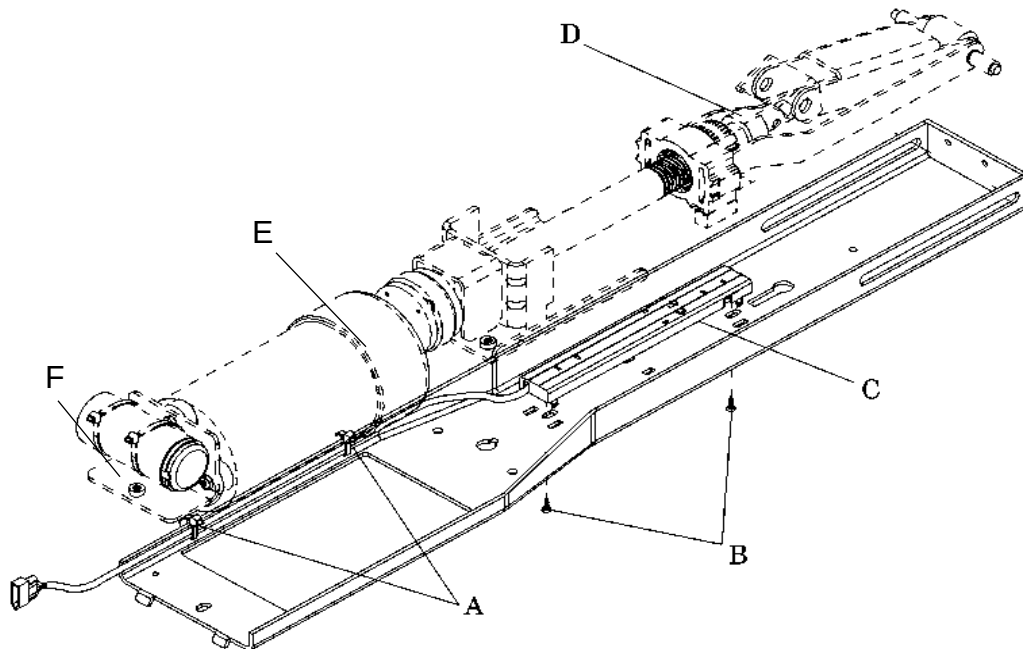


Figura 2 6. Micro switch acoplado en un riel con desplazamiento alto y bajo. [5]

A) Sujetadores de la señal del micro switch ON – OFF.

B) Tornillos para sujetar el riel de los micro switch.

C) Micro switch de límite alto para el paro total del motor.

- D) Soporte radial para el tornillo que sujeta la articulación.
- E) Motor de VAC para girar la articulación permitiendo subir o bajar.
- F) Capacitor de arranque para el motor de VAC.

Nota: VAC (Voltaje de corriente alterna).

2.7. El motor para el movimiento de las articulaciones

Es necesario saber que es un motor de VAC y su funcionamiento para poder comprender su utilidad en las camas eléctricas hospitalarias.

Se denomina motor de CA a aquellos motores eléctricos que funcionan con corriente alterna. Un motor es una máquina eléctrica, esto es, un aparato que convierte una forma determinada de energía en energía mecánica de rotación o par, generando un torque para posteriormente ser utilizado para el movimiento mecánico donde esté montado, los motores eléctricos se componen en dos partes una fija llamada estator y una móvil llamada rotor.

Un motor eléctrico convierte la energía eléctrica en fuerzas de giro por medio de la acción mutua de los campos magnéticos. Como se muestra en la Figura 2.7.



Figura 2.7. Motor de corriente alterna con capacitor de arranque. [11]

El motor que se utiliza para la cama eléctrica hospitalaria Hill-Rom Advance Bed, es un motor especial debido a que éste consta con tres pines de entrada de color blanco, rojo y azul donde: el pin de color blanco (Es el neutro), el pin de color rojo (Es la fase) y el pin de color azul (Es la fase), es decir que este motor puede girar en ambos sentidos sin necesidad de diseñar un driver para hacerlo, por ejemplo si conectamos el pin de color blanco y el pin de color azul a neutro y fase o viceversa respectivamente, el motor girará en sentido horario a la manecillas del reloj, de lo contrario si conectamos el pin de color blanco y el pin de color rojo a neutro y fase o viceversa respectivamente, el motor girará en sentido anti horario con respecto a las manecillas del reloj.

En las Figuras 2.7 y 2.8 se muestra el motor de corriente alterna que utiliza la cama eléctrica hospitalaria Hill-Rom Advance Bed.

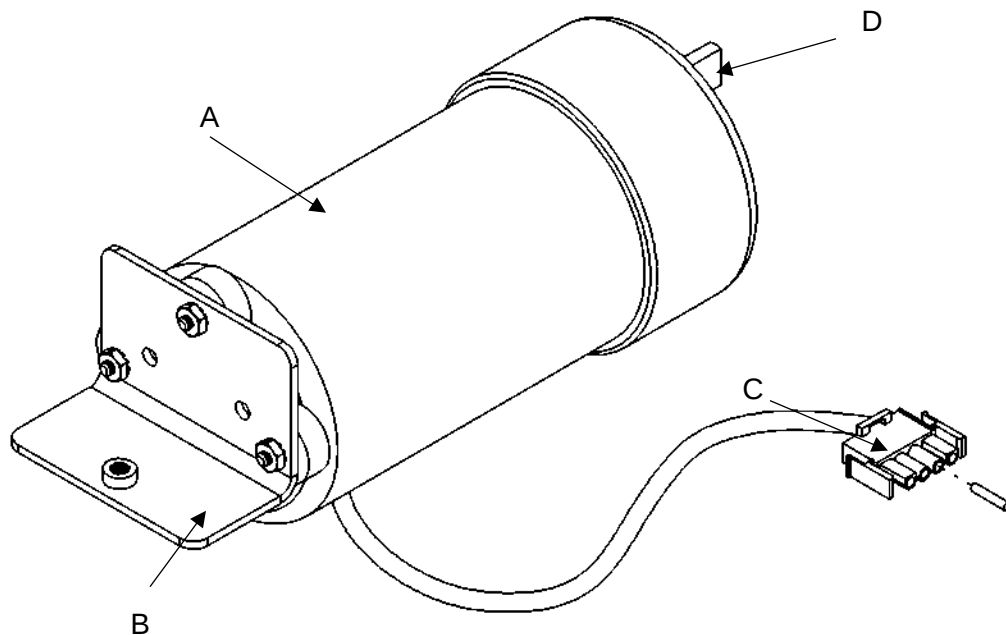


Figura 2.8. Motor de corriente alterna sin capacitor de arranque. [5]

A. Motor de corriente alterna.

B. Base para el capacitor de VCA.

C. Conector tipo hembra con seguro.

D. Rotor.

Siguiendo el mismo principio del funcionamiento del motor (A), todas las articulaciones de la cama eléctrica tienen el mismo motor con su propio conector (C) para elevar o bajar las articulaciones para las posiciones del movimiento de la cabeza, elevación de la cama, pies y trendelemburg tal como se muestra en las Figuras 2.9 y 2.10. Todo motor tiene un base para colocar el capacitor de arranque (B) y un rotor (D) para girar el engrane.

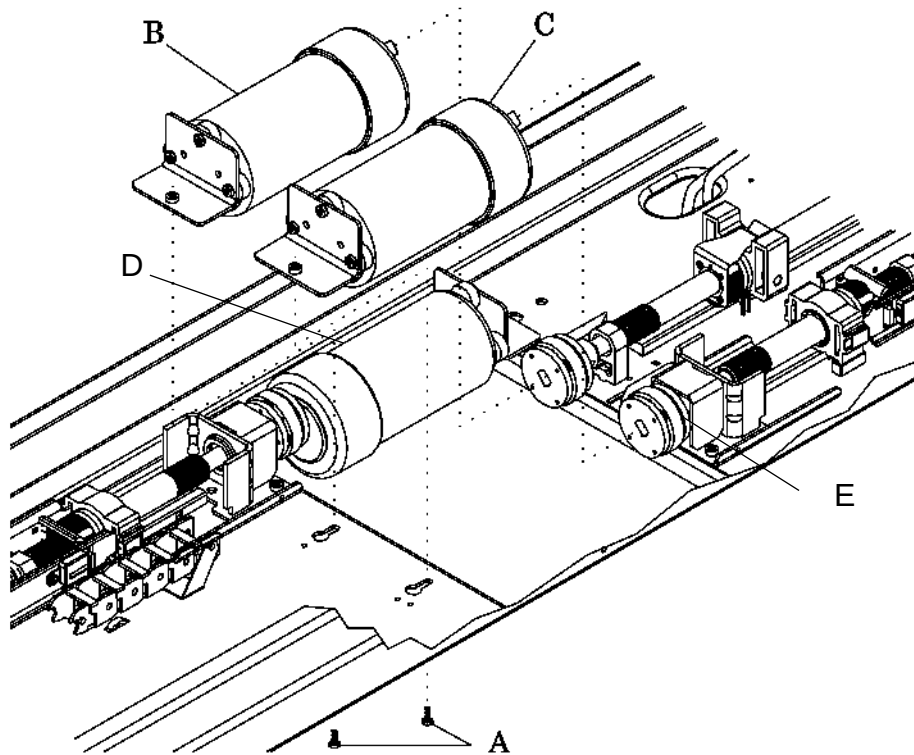


Figura 2.9. Motores para las articulaciones de la cabeza y elevación de la cama. [5]

A) Tornillos para sujetar los motores.

B) Motor para elevar o bajar la articulación de la cabeza.

C) Motor para inclinación del trendelemburg.

D) Motor para inclinación inversa del trendelemburg.

E) Acoplamiento mecánico

En la Figura 2.9 se muestra el diagrama mecánico del motor para el trendelemburg (C y D) y el motor de la articulación para la cabeza (B) sujetos mediante tornillos (A), permitiendo que los motores no se desacoplen (E) evitando accidentes.

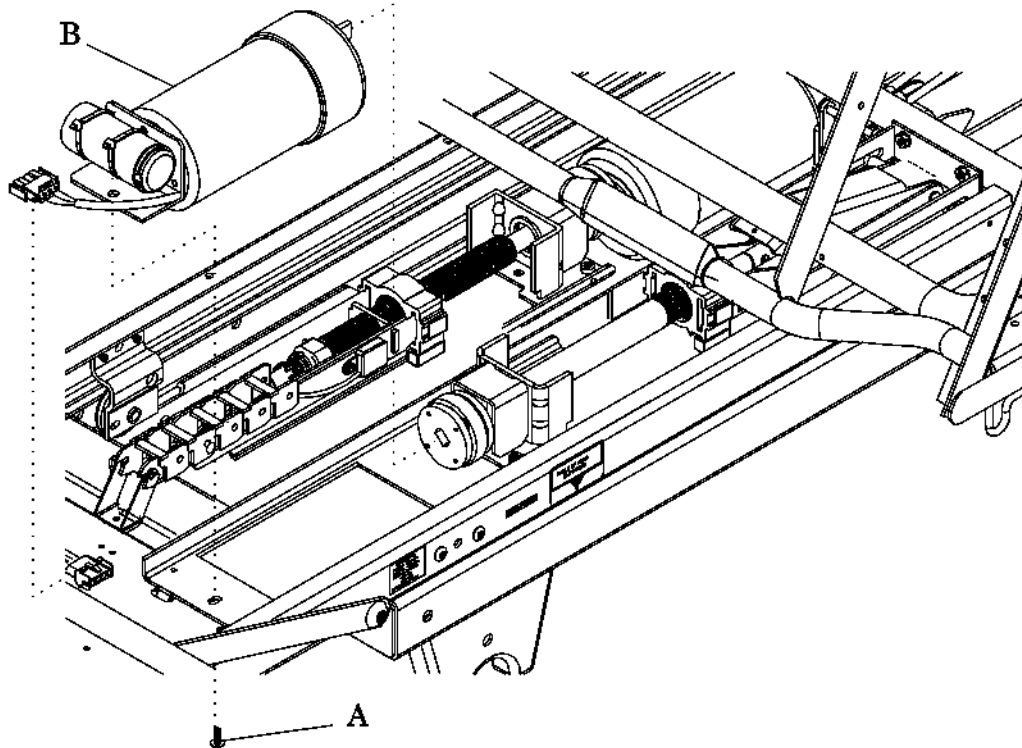


Figura 2.10. Motores para las articulaciones de los pies. [5]

A) Tornillo para sujetar el motor.

B) Motor para elevar o bajar la articulación de los pies.

El rotor de los motores se conecta a un sistema de engranes por medio de una rosca, permitiendo que cuando gire el motor(B) y éste se detenga por medio de la rosca, la articulación pueda quedar fija y no se mueva, de esta manera se asegura al paciente que la cama eléctrica es fiable en cuanto a todas sus articulaciones y por medio del tornillo (A) sujetar el motor.

CAPÍTULO III. HERRAMIENTAS DE HARDWARE Y SOFTWARE

3.1. Arduino

Arduino es una plataforma de hardware libre basada en una placa con un microcontrolador. Con software y hardware flexibles, Arduino ha sido diseñado para adaptarse a las necesidades de todo tipo de público, desde aficionados, hasta expertos en robótica o equipos electrónicos. [12]

Ante todo, y sobre todo es un microcontrolador, es una mini computadora completa integrada en un chip, con su CPU, memoria de programa, memoria de datos y circuitos para el control de periféricos.

El microcontrolador necesita para su correcto funcionamiento, de algunos circuitos auxiliares y complementos tales como:

- 1-. Entrada de alimentación.
- 2-. El oscilador de trabajo.
- 3-. Circuito de RESET.
- 4-. La conexión USB.
- 5-. Los accesos a las líneas de entrada y salida, etc.

También consta de un simple, pero completo, entorno de desarrollo, que nos permite interactuar con la plataforma de manera muy sencilla. Se puede definir como una sencilla herramienta de contribución a la creación de prototipos, entornos, u objetos interactivos destinados a proyectos multidisciplinarios. [12]

En la Figura 3.1 se muestra la placa Arduino, en este caso más específicamente la versión 3.0 del Arduino nano.

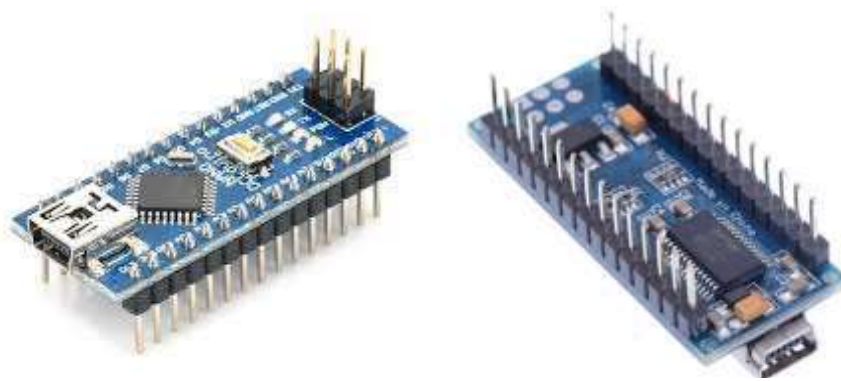


Figura 3.1. Placa Arduino nano. [12]

3.1.1. Historia del Arduino

De forma estricta, el proyecto Arduino se inició en el año 2005 como un proyecto enfocado a estudiantes en el Instituto IVREA (IDII), en Ivrea Italia.

En aquellos años, los estudiantes usaban el microcontrolador BASIC Stamp, cuyo costo era de \$100USD, un costo considerablemente alto para un estudiante promedio. Antes del año 2005, específicamente durante el año 2003, Hernando Barragán había creado la plataforma de desarrollo *Wiring* como resultado de su proyecto de tesis en la maestría en el IDII, bajo la supervisión de *Massimo Banzí* y Casey Reas, quienes eran conocidos por haber trabajado en el lenguaje *Processing* y daban clases en el IDII.

El objetivo del proyecto era crear herramientas simples y de bajo costo para la creación de proyectos digitales por parte de personas sin altos conocimientos técnicos o sin un perfil de ingeniería.

El proyecto *Wiring* era una placa de desarrollo de hardware que constaba de una placa de circuito impreso (PCB) con un microcontrolador ATmega168, un Ambiente de Desarrollo Integrado (IDE) basado en funciones de procesamiento y una biblioteca de funciones para programar fácilmente el microcontrolador.

Regresando al año 2005, Massimo Banzi junto con David Mellis (otro estudiante del IDII) y David Cuartielles, agregaron soporte a Wiring para el microcontrolador *ATmega8*, más económico que el inicial (*Atmega168*). Pero en lugar de continuar el desarrollo en Wiring, se separaron del proyecto y lo renombraron Arduino.

Arduino es una compañía de fuente abierta y hardware abierto, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

Los productos que vende la compañía son distribuidos como Hardware y Software Libre, bajo la Licencia Pública General de GNU (GPL) y la Licencia Pública General Reducida de GNU (LGPL), permitiendo la manufactura de las placas Arduino y distribución del software por cualquier individuo.

3.1.2. Características técnicas

El Arduino Nano es una pequeña y completa placa basada en el ATmega328. Es muy compacto y puedes ser utilizado fácilmente en un protoboard.

El microcontrolador de alto desempeño Atmega328P de 8 bits, con arquitectura RISC. El Atmega328P en formato DIP es utilizado en el Arduino Uno R3. Posee características como: 32 KB de memoria flash; con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de propósito general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART (Transmisor-Receptor Síncrono/Asíncrono Universal), una

interface serial orientada a byte de 2 cables, SPI e I2C, 6 canales 10-bit Conversor A/D, "watchdog timer" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software.

En la Tabla 3.1 se muestra un resumen de las características técnicas más relevantes del Arduino nano para cumplir con los requisitos al momento de diseñar un proyecto, y así poder evitar problemas de mal funcionamiento o en otros casos dañar la placa o el microcontrolador.

Tabla 3.1. Características técnicas de la placa Arduino nano. [12]

Elemento	Información
Microcontrolador	ATmega328p.
Tensión de Operación (nivel lógico)	5V
Tensión de Entrada (recomendado)	7-12V
Tensión de Entrada (límites)	6-20V
Pines E/S Digitales	14 (de los cuales 6 proveen de salida PWM.
Entradas Analógicas	8 pines
Corriente máx por cada PIN de E/S	40mA
Memoria Flash	32 KB (ATmega328) de los cuales 2KB son usados por el bootloader.
SRAM	2 KB (ATmega328).
EEPROM	1 KB (ATmega328).
Frecuencia de reloj	16MHz
Dimensiones	18.5mm x 43.2mm.

3.1.3. Alimentación de la placa Arduino Nano

El Arduino Nano puede ser alimentado usando el cable USB Mini B, con una fuente externa no regulada de 6-20V (pin 30), o con una fuente externa regulada de 5V (pin 27). La fuente de alimentación es seleccionada automáticamente a aquella con mayor tensión.

El chip FTDI FT232RL que posee el Nano solo es alimentado si la placa está siendo alimentada usando el cable USB. Cuando se utiliza una fuente

externa (no USB), la salida de 3.3V (la cual es proporcionada por el chip FTDI) no está disponible y los pines D1 y D0 parpadearán si los pines digitales 0 o 1 están a nivel alto.

3.1.4. Pinout Arduino Nano

El Pinout, es uno de los documentos de referencia más útiles y que frecuentemente consultaremos a la hora de realizar un montaje. El caso de Arduino no es una excepción, teniendo en cuenta la gran cantidad de pines y modelos de placas disponibles.

A medida que la complejidad de nuestros circuitos aumenta y utilizamos funciones más complejas, como comunicación entre dispositivos o interrupciones, resulta más conveniente disponer a mano de estos esquemas, para poder consultar rápidamente los datos de cada pin en las distintas placas existentes.

En la Figura 3.2 se muestra el Pinout del Arduino nano, este diagrama nos muestra el funcionamiento y configuración de todos los pines internos del micro controlador.

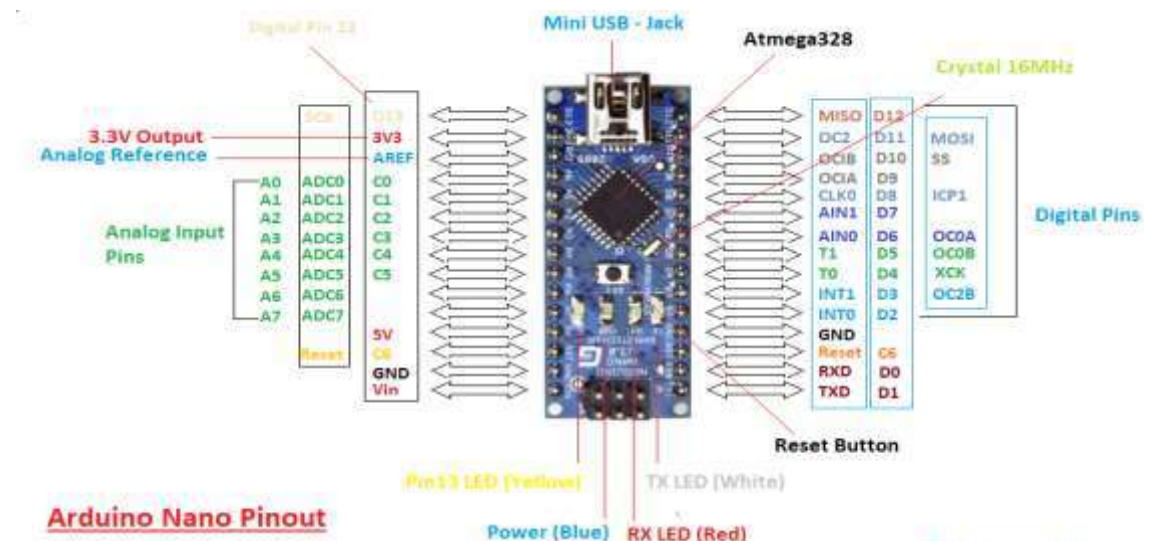


Figura 3.2. Diagrama Pinout de la placa Arduino nano. [19]

3.1.5. Entradas y salidas

Cada uno de los 14 pines digitales del Nano puede ser usado como entrada o salida, usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Operan a 5 voltios. Cada pin puede proveer o recibir un máximo de 40mA y poseen una resistencia de pull-up (desconectada por defecto) de 20 a 50 kOhms.

El Nano posee 8 entradas analógicas, cada una de ellas proveen de 10 bits de resolución (1024 valores diferentes). Por defecto miden entre 5V y 0V. También.

Terminales Digitales

Las terminales digitales de una placa Arduino pueden ser utilizadas para entradas o salidas de propósito general a través de los comandos de programación `pinMode()`, `digitalRead()`, y `digitalWrite()`. Cada terminal tiene una resistencia pull-up que puede activarse o desactivarse utilizando `digitalWrite()` (con un valor de HIGH o LOW, respectivamente) cuando el pin está configurado como entrada.

Los pines D0 (RX) y D1 (TX) son utilizado para recibir (RX) y transmitir (TX) datos serie de niveles TTL normalmente para establecer una comunicación serial.

Interrupciones externas: pines D2 y D3. Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor.

Los pines D3, D5, D6, D9, D10, y D11 pueden, proporcionan salidas PWM de 8 bit con la función `analogWrite()`.

El Reset: Está conectado a la línea de reset del módulo.

SPI: D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK). Estas terminales soportan comunicación SPI (Bus serial de interfaz de periféricos). Aunque esta funcionalidad esta proporcionada por el hardware.

Pines Analógicos

Los pines de entrada analógicos soportan conversiones analógico-digital (ADC) de 10 bit utilizando la función *analogRead()*.

Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14 hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Arduino Mini) no pueden ser utilizadas como pines digitales.

I2C: D4 SDA (Señal de Datos) y D5 SCL (Señal de Reloj). Soportan comunicaciones I2C TWI (Interfaz por dos cables) utilizando la librería Wire.

Otros Pines

AREF. Referencia de voltaje para las entradas analógicas. Utilizada con la función *analogReference()*.

Reset

Poner esta línea a LOW para resetear el microcontrolador. Utilizada típicamente para añadir un botón de reset a shields que bloquean el de la placa principal. [13]

3.2. Lenguaje de programación Arduino

La programación de Arduino es la programación de un microcontrolador.

Programar Arduino consiste en traducir a líneas de código las tareas automatizadas que queremos hacer leyendo de los sensores y en función de las condiciones del entorno programar la interacción con el mundo exterior mediante unos actuadores.

Arduino proporciona un entorno de programación sencillo y potente para programar, pero además incluye las herramientas necesarias para compilar el programa y programar el programa compilado en la memoria

flash del microcontrolador. Además, el IDE nos ofrece un sistema de gestión de una biblioteca y placas muy práctico.

Para programar un Arduino, el lenguaje estándar es C++, aunque es posible programarlo en otros lenguajes. No es un C++ puro, sino que es una adaptación que proviene de avr-libc que provee de una biblioteca de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel.

GCC es un conjunto de compiladores que se considera el estándar para los Sistemas Operativos derivados de UNIX y requiere de un conjunto de aplicaciones conocidas como binutils que son unas herramientas de programación para la manipulación de código de objeto.

Cuando GCC está construido para ejecutarse en un sistema como Linux, Windows o mac OS y generar código para un microcontrolador AVR, entonces se denomina avr-gcc. Avr-gcc es el compilador que usa el IDE de Arduino para convertir el sketch (programa de Arduino) en C++ a un fichero binario (.hex) que es el que se carga en la memoria flash del MCU y que ejecuta el programa. [12]

3.2.1. Variables y operadores

Una variable es un elemento de nuestro sketch que guarda un determinado contenido. Ese contenido (el valor de la variable) se podrá modificar en cualquier momento de la ejecución del sketch.

En Arduino antes de poder utilizar una variable debemos crearla, a esto se llama declarar una variable, y podemos asignarle un valor inicial, y a esto le llamamos inicializar la variable. A continuación, unos ejemplos de cómo podemos hacerlo.

```
int nombreVariable1, nombreVariable2, nombreVariable3; //Crea 3 variables de tipo int
long nombreVariable; //Crea una variable de tipo long
```

`float nombreVariable = 2.1; //Crea una variable tipo float con el valor inicial de 2.1`

Tipos de Variables

Hay muchos tipos de variables en Arduino. El tipo de cada una lo definiremos de acuerdo a los datos que vamos a guardar en la variable, como se muestra en la Tabla 3.2. Además, las variables son muy utilizadas al momento de programar, debido a que podemos guardar o cambiar sus valores y también para recurrir a ellos cuando los necesitemos en nuestro programa.

Tabla 3.2. Tipos de variables en Arduino. [14]

Nombre	Tamaño	Rango	Ejemplo
boolean	1 bit	1 o 0	<code>boolean bit Encendido=0;</code>
byte	8 bit	0 a 255	<code>byte valor=20;</code>
char	8 bit	-128 a 127	<code>char código='B';</code>
int	16 bit	-32,768 a 32,767	<code>int val1=30000;</code>
long	32 bit	-2,147,483,648 a 2,147,483,647	<code>long suma=100000;</code>
float	32 bit	-3.4028235E+38 a 3.4028235E+38	<code>float constanteA = 5.235;</code>
unsigned char	8 bit	0 a 255	<code>unsigned char letra1='A';</code>
unsigned int	16 bit	0 a 65,535	<code>unsigned int sensor=50000;</code>
unsigned long	32 bit	0 a 4,294,967,295	<code>unsigned long conteo10=1;</code>

Constantes

Las constantes se pueden entender como variables de solo lectura, cuando intentemos modificarlas el compilador dará un error. Lo único que hay que hacer para definirla es preceder la declaración de la variable con la palabra clave “const”. Por ejemplo:

`const int valor1=125;`

Existen otras constantes definidas en el lenguaje Arduino. Las constantes true y false son unas de ellas, false es la más fácil de definir y es igual a 0, y true, aunque puede interpretarse como 1. [12]

Operadores aritméticos

Los operadores aritméticos empleados en lenguaje Arduino se muestran en la Tabla 3.3.

Tabla 3.3. Tipos de operadores matemáticos en Arduino. [14]

Símbolo	Descripción
=	Operador de asignación.
+	Operador suma.
-	Operador resta.
*	Operador multiplicación.
/	Operador división.
%	Operador módulo.

Operadores compuestos

Son aquellos que nos permiten abreviar código y los usados en Arduino tal como se muestran en la Tabla 3.4.

Tabla 3.4. Tipos de operadores compuestos en Arduino. [14]

Símbolo	Ejemplo y descripción
++	x++; // incrementa x en uno y retorna el valor anterior de x
--	x--; // decrementa x en uno y retorna el valor anterior de x
+=	x += y; // equivalente a la expresión x = x + y;
-=	x -= y; // equivalente a la expresión x = x - y;
*=	x *= y; // equivalente a la expresión x = x * y;
/=	x /= y; // equivalente a la expresión x = x / y;
%=	x %= y; // equivalente a la expresión x = x % y;

Precedencia de operadores

Es necesario saber el orden de ejecución de los operadores como se muestran en la Tabla 3.5 del mayor a menor prioridad, para evitar errores de código.

Tabla 3.5. Jerarquía de operadores en Arduino. [14]

Símbolo	Descripción	Ejemplos
()	Paréntesis	(x+y)*z
++ —	Incremento, decremento	x++, y–
* / %	Multipliación, división, modulo	x * 1.5
+ –	Adición, sustracción	y + 10
< > <= >=	Comparaciones menores que, mayor que	if (x < 255)
== !=	Igual o diferente	if (x == HIGH)
&&	AND lógico	if (x == HIGH && x > y)
	OR lógico	if (x == HIGH x > y)
= += -= *= /=	Asignación y asignación compuesta	x = y

3.3. IDE Arduino

Dado que el Arduino es como una mini computadora que ejecuta una serie de instrucciones, necesitaremos un programa para desarrollar. Este programa se llama IDE, que significa "Entorno de Desarrollo Integrado".

En él escribimos el programa que Arduino ejecute. Una vez escrito, lo cargaremos a través del USB y el micro controlador inicia la ejecución del programa.

Verifica si el programa está bien escrito, el código se compila correctamente, de lo contrario si el código tiene un error de sintaxis, detiene el proceso de compilación y muestra el error detectado en la consola que se encuentra en la parte inferior del IDE.

En la Figura 3.3 se muestra el IDE de Arduino.

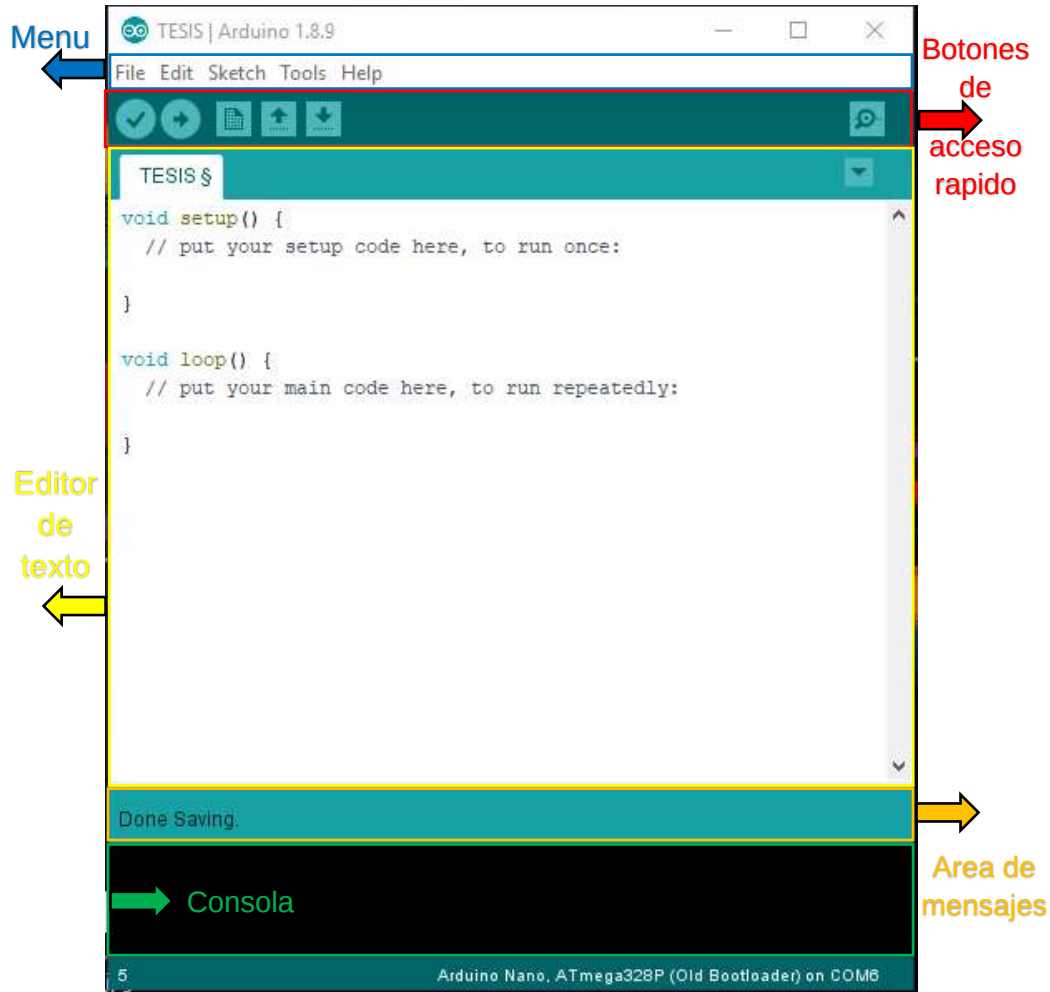


Figura 3 3. IDE de Arduino. [12]

En los botones de acceso rápido tenemos los siguientes iconos:

-  Verifica si tu programa está bien escrito y puede funcionar.
-  Carga el programa a la placa de Arduino tras compilarlo.
-  Crea un programa nuevo.
-  Abre un programa.
-  Guarda el programa en el disco duro del ordenador.

-  (En la parte derecha de la barra de herramientas se encuentra el Monitor Serial) abre una ventana de comunicación con la placa Arduino en la que podemos ver las respuestas que nuestro Arduino nos está dando, siempre que tengamos el USB conectado del microcontrolador.

3.3.1. Estructura de un Sketch

Un programa de Arduino se denomina sketch o proyecto y tiene la extensión .ino. Importante: para que funcione el sketch, el nombre del fichero debe estar en un directorio con el mismo nombre que el sketch.

No es necesario que un sketch esté en un único fichero, pero si es imprescindible que todos los ficheros estén dentro del mismo directorio que el fichero principal.

La estructura básica de un sketch de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes son obligatorias y encierran bloques que contienen declaraciones, estamentos o instrucciones. *setup()* y *loop()*

Setup() es la parte encargada de recoger la configuración y *loop()* es la que contiene el programa que se ejecuta cíclicamente (de ahí el término loop –bucle–). Ambas funciones son necesarias para que el programa trabaje.

A continuación, se muestra en la Figura 3.4 un Sketch de un ejemplo simple del encendido y apagado del led que se encuentra conectado en el pin 13 de la placa Arduino nano.

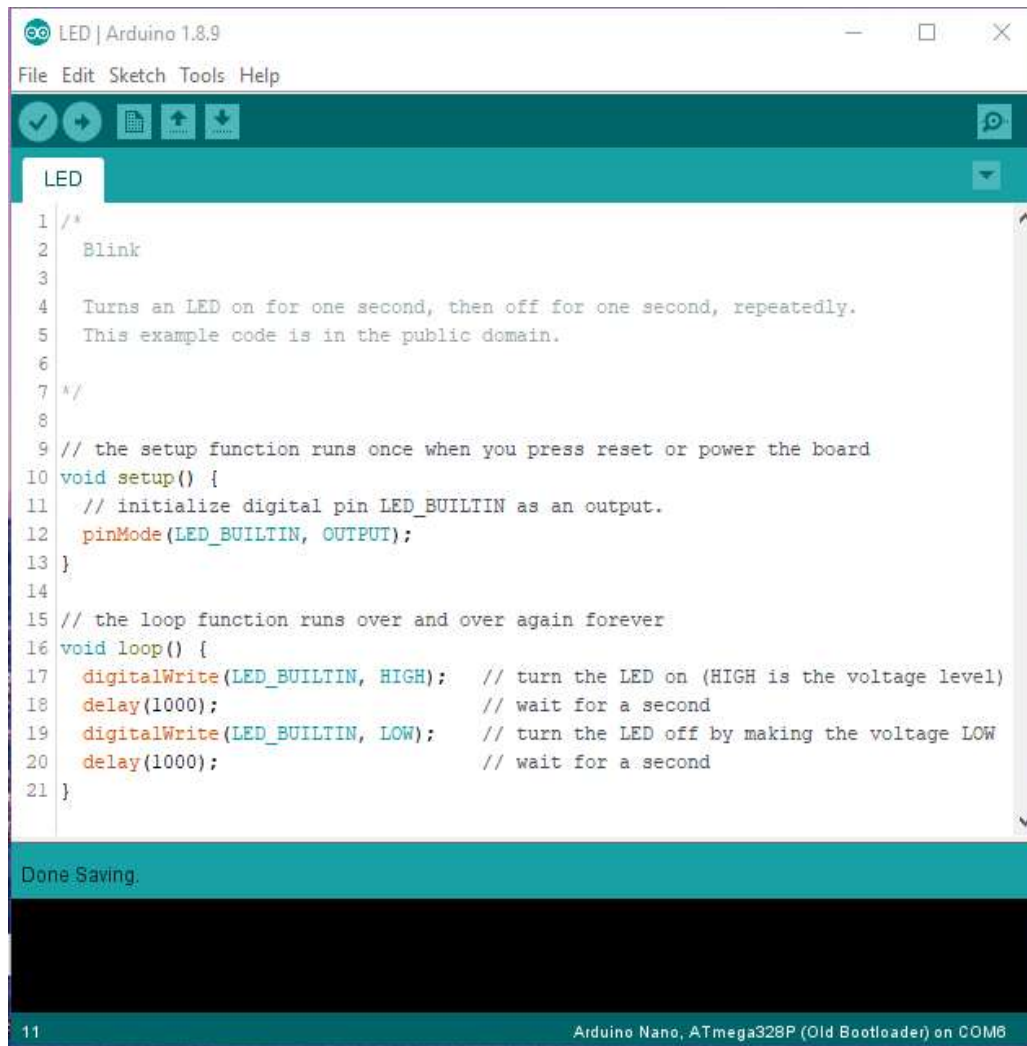


Figura 3.4. Sketch de Arduino para el encendido y apagado de un led.

Para cargar el primer sketch debe tener instalado el IDE de Arduino en su computadora. Una vez instalado el software, diríjase a la pestaña: Archivo ->Ejemplos-> Basics Deberá elegir Blink Posteriormente deberá elegir la placa en: Herramientas-> Placa -> Arduino Nano Conecte su placa y a continuación vaya a: Herramientas->Puerto Elija el puerto COM que su PC asigno a su placa, como se muestra en la Figura 3.5

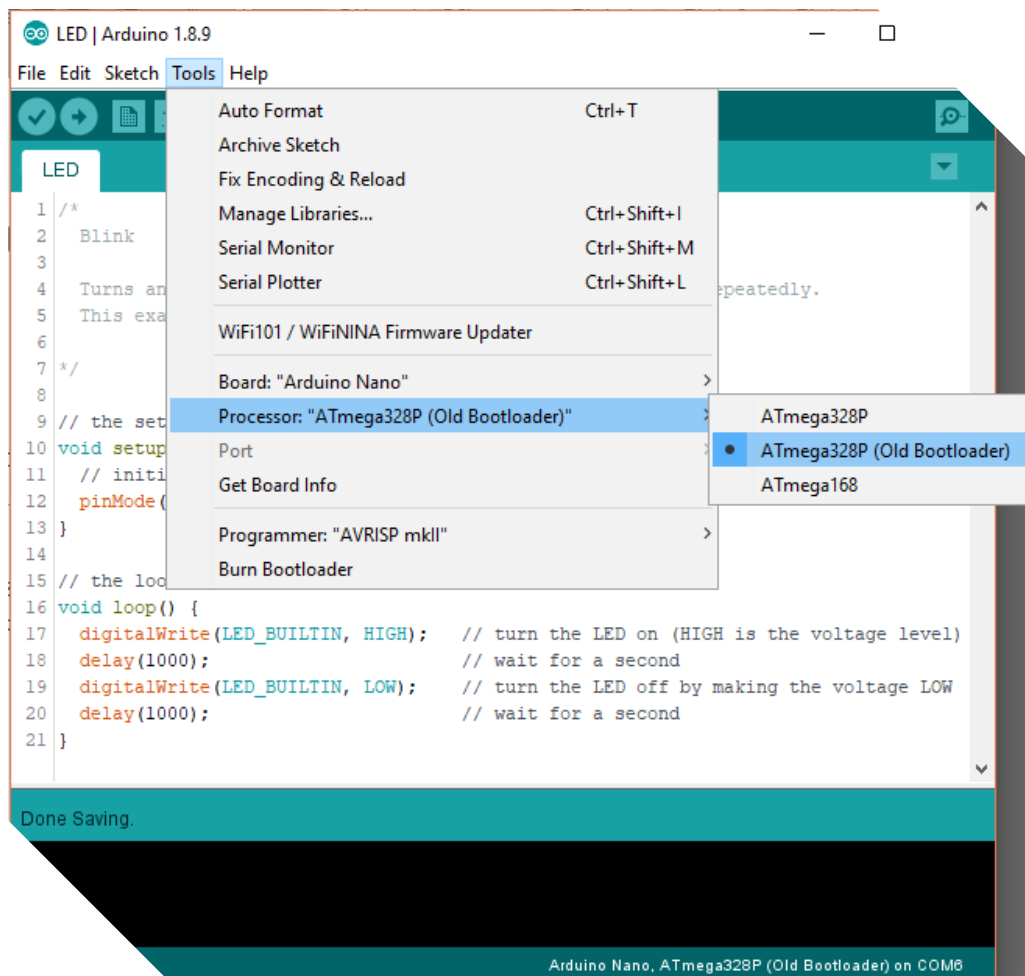


Figura 3.5. Elección de la placa y del puerto de conexión para Arduino nano.

3.4. Módulo Bluetooth HC-05

El módulo Bluetooth HC-05 es ideal para utilizar en todo tipo de proyectos donde necesites una conexión inalámbrica fiable y sencilla de utilizar. Se configura mediante comandos AT y tiene la posibilidad de hacerlo funcionar tanto en modo maestro como esclavo. Eso quiere decir que puedes conectar dos módulos juntos, conectar tu robot al móvil o incluso hacer una pequeña red de sensores comunicados entre ellos con un maestro y varios esclavos.

El módulo Bluetooth HC-05 puede alimentarse con una tensión de entre 3.3 y 6V (normalmente 5V), pero los pines TX y RX utilizan niveles de 3,3V por lo que no se puede conectar directamente a placas de 5V.

Debes utilizar dos pequeñas resistencias como divisor de tensión para que el módulo no se dañe.

Tiene un LED incorporado que indica el estado de la conexión y si está emparejado o no en función de la velocidad del parpadeo. En la Figura 3.5 se muestra el aspecto físico del módulo bluetooth HC-05 y sus pines de entrada y salida. [15]

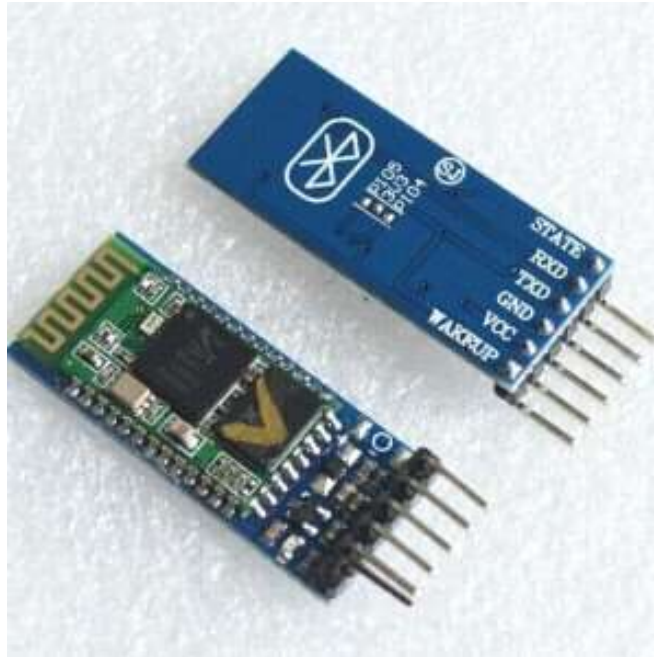


Figura 3.6. Modulo HC-05. [15]

3.4.1. Características

En la Tabla 3.6 se muestran las características técnicas de modulo bluetooth HC-05.

Tabla 3.6. Características técnicas del módulo HC-05. [15]

Nombre	Descripción
Protocolo Bluetooth	v1.1 / 2.0.
Frecuencia	Banda ISM de 2,4 GHz.
Modulación	GFSK
Potencia de transmisión	Menos de 4dB.
Sensibilidad	Menos de -84dBm en el 0,1% BER.
Ratio asíncronos	2.1Mbps (Max) / 160 kbps.
Síncrono	1Mbps / 1Mbps.
Perfiles de la ayuda	puerto serie Bluetooth (maestro y esclavo).
Fuente de alimentación	+ 3.3VDC 50mA. (soporta de 3.3 a 6V)
Temperatura de trabajo	-5 ° C a 45 ° C.

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo.

Los dispositivos que incorporan este protocolo pueden comunicarse entre sí cuando se encuentran dentro de su alcance.

Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente.

Estos dispositivos se clasifican como “Clase 1”, “Clase 2” o “Clase 3” en referencia a su potencia de transmisión.

Los dispositivos con Bluetooth también pueden clasificarse según su capacidad de canal como se muestra en la Tabla 3.7

Tabla 3.7. Versiones de los módulos Bluetooth. [15]

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	32 Mbit/s

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores). Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez.

3.4.2. Funcionamiento

El módulo Bluetooth HC-05 viene configurado de fábrica como "Esclavo" (slave), pero se puede cambiar para que trabaje como "maestro" (master), además al igual que el HC-06, se puede cambiar el nombre, código de vinculación, velocidad y otros parámetros más.

Definamos primero que es un dispositivo bluetooth maestro y dispositivo esclavo:

Módulo bluetooth HC-05 como esclavo

Cuando está configurado de esta forma, se comporta similar a un HC-06, espera que un dispositivo bluetooth maestro se conecte a este, generalmente se utiliza cuando se necesita comunicarse con una PC o Celular, pues estos se comportan como dispositivos maestros.

Módulo bluetooth HC-05 como Maestro

En este modo, EL HC-05 es el que inicia la conexión. Un dispositivo maestro solo se puede conectarse con un dispositivo esclavo. Generalmente se utiliza este modo para comunicarse entre módulos bluetooth. Pero es necesario antes especificar con que dispositivo se tiene que comunicar, esto se explicará más adelante.

El módulo HC-05 viene por defecto configurado de la siguiente forma:

- Modo o role: Esclavo.
- Nombre por defecto: HC-05.
- Código de emparejamiento por defecto: 1234.
- La velocidad por defecto (baud rate): 9600.

Modos de trabajo del HC-05

EL Módulo HC-05 tiene 4 estados los cuales es importante conocer:

Estado Desconectado:

- Entra a este estado tan pronto alimentas el módulo, y cuando no se ha establecido una conexión bluetooth con ningún otro dispositivo.
- EL LED del módulo en este estado parpadea rápidamente.
- En este estado a diferencia del HC-06, el HC-05 no puede interpretar los comandos AT.

Estado Conectado o de comunicación:

- Entra a este estado cuando se establece una conexión con otro dispositivo bluetooth.
- El LED hace un doble parpadeo.
- Todos los datos que se ingresen al HC-05 por el Pin RX se transmiten por bluetooth al dispositivo conectado, y los datos

recibidos se devuelven por el pin TX. La comunicación es transparente.

Modo AT 1

- Para entrar a este estado después de conectar y alimentar el módulo es necesario presionar el botón del HC-05.
- En este estado, podemos enviar comandos AT, pero a la misma velocidad con el que está configurado.
- EL LED del módulo en este estado parpadea rápidamente igual que en el estado desconectado.

Modo AT 2

- Para entrar a este estado es necesario tener presionado el botón al momento de alimentar el módulo, después el módulo debe encender con el mismo botón presionado, después de haber encendido se puede soltar y permanecerá en el estado donde el led estará oscilando entre 2 segundos.
- En este estado, para enviar comandos AT es necesario hacerlo a la velocidad de 38400 baudios, esta velocidad se selecciona en el menú del IDE de Arduino -> herramienta -> monitor serial. [15]

3.5. Dfplayer TF-16P

El DFPlayer TF-16P es un módulo reproductor MP3, es pequeño y de bajo precio. El módulo se puede usar como un módulo independiente con batería, altavoz y botones adjuntos o se puede usar en combinación con un Arduino nano o cualquier otro microcontrolador con capacidades RX / TX.

3.5.1. Características

DFPlayer Mini es capaz de reproducir formatos de fichero MP3, WMA y WAV.

Dispone de un lector micro SD, con una capacidad máxima de 32GB. Soporta hasta 100 carpetas y puede acceder hasta 255 canciones.

Proporciona velocidades de muestreo de 8, 11.025 12 16 22.05 24 32 44.1 y 48 kHz, y salida con DAC de 24 bits. Dispone de 30 niveles de volumen ajustable, ecualizador de 6 niveles, y una relación señal ruido (SNR) de 85dB.

Recibe comandos desde Arduino por puerto serie, esta conexión se muestra en la Figura 4.5. Dispone de funciones para reproducir, detener, pausar, avanzar, retroceder entre canciones. La salida se realiza directamente al altavoz, a través de los pines correspondientes. [16]

En la Tabla 3.8 se muestran las características técnicas del módulo Dfplayer TF-16P.

Tabla 3.8. Características técnicas del módulo Dfplayer TF-16P. [16]

Descripción
Frecuencias de muestreo admitidas (kHz): 8 / 11.025 / 12/16 / 22.05 / 24/32 / 44.1 / 48.
Salida DAC de 24 bits, soporte para rango dinámico 90dB, soporte SNR 85dB.
Soporte completo FAT16, sistema de archivos FAT32, soporte máximo 32G de la tarjeta TF, soporte 32G del disco U, NORFLASH de 64M bytes. Una variedad de modos de control, modo de control de E / S, modo de serie, modo de control de botón AD.
Función de espera de sonido de publicidad, la música se puede suspender. Cuando la publicidad termina, la música continúa reproduciéndose.
Los datos de audio ordenados por carpeta, admiten hasta 100 carpetas, cada carpeta puede contener hasta 255 canciones.
Volumen ajustable de 30 niveles, ecualizador de 6 niveles ajustable.

3.5.2. Aplicaciones

En la Tabla 3.9 se muestran las aplicaciones más comunes para el uso y programación del módulo Dfplayer TF-16P.

Tabla 3.9. Aplicaciones más utilizadas para del módulo Dfplayer TF-16P. [16]

Descripción
Transmisión de voz de navegación del automóvil.
Inspectores de transporte vial, estaciones de peaje de voz.
Estación de tren, inspección de seguridad de autobuses mensajes de voz
Electricidad, comunicaciones, avisos de voz en las salas de negocios financieros.
El vehículo que entra y sale del canal verifica que el mensaje de voz aparezca.
Los mensajes de voz del canal de control de fronteras de seguridad pública.
Indicadores de voz de alarma de incendio.
El equipo de transmisión automática, emisión regular.
La voz de conducción segura del coche eléctrico avisa.
Alarma de falla de equipo electromecánico.
Alarma de voz multicanal o guía de funcionamiento del equipo de voz.

3.5.3. Pinout del módulo Dfplayer TF-16P

En la Figura 3.7 se muestra el nombre de los pines que conforman el módulo Dfplayer TF-16P.

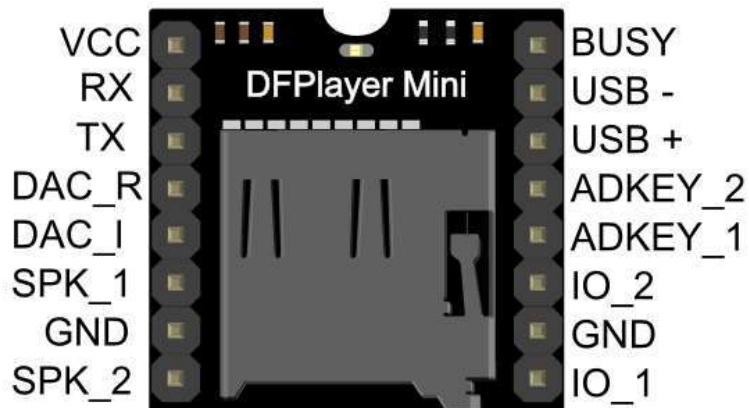


Figura 3.7. Módulo Dfplayer TF-16P. [16]

En la Tabla 3.10 se muestra la descripción de todos los pines del módulo Dfplayer TF-16P.

Tabla 3.10. Descripción del Pinout del módulo TF-16P. [16]

Pines	Descripción	Nota
VCC	Entrada de voltaje	DC 3.2 ~ 5.0v; Típico 4.2v
RX	Entrada serial UART	
TX	Salida serial UART	
DAC_R	Salida de audio derecha	Amplificado para audífono
DAC_L	Salida de audio izquierda	Amplificado para audífono
SPK2	Bocina-	Salida con menos de 3w
GND	Tierra	
SPK1	Bocina+	Salida con menos de 3w
IO1	Trigger puerto 1	Pulso corto para anterior canción (pulso largo para bajar volumen)
GND	Tierra	
IO1	Trigger puerto 2	Pulso corto para siguiente canción (pulso largo para subir volumen)
ADKEY1	AD puerto1	Pulso para reproducir la primera canción
ADKEY2	AD puerto 2	Pulso para reproducir la quinta canción
USB+	USB+ DP	Puerto USB
USB-	USB- DM	Puerto USB
BUSY	Estado de reproducción	0v Reproduciendo música / 5v Sin reproducir música

3.5.4. Funcionamiento

Para tener el buen funcionamiento de modulo Dfplayer, primero debemos de establecer una comunicación serial UART con una velocidad de 9600 baudios.

Además, puede controlarse con cualquier microcontrolador que soporte comunicación UART, esto le permite controlar de manera digital características como: play, pause, stop, next, prev, volumen, etc. En la Tabla 3.11 se muestra los comandos digitales para el control del módulo Dfplayer mediante la placa Arduino nano.

Tabla 3.11. Comandos para el módulo Dfplayer TF-16P. [16]

CMD	Descripción de la función	Parámetros(16bits)
0x01	Siguiente	
0x02	Anterior	
0x03	Track específico (NUM)	0-2999
0x04	Subir volumen	
0x05	Bajar volumen	
0x06	Volumen específico	0-30
0x07	Equalización (0/1/2/3/4/5)	Normal/Pop/Rock/Jazz/Classic/Base
0x08	Specify playback mode (0/1/2/3)	Repetir/Repetir carpeta/Repetir canción/Random
0x09	Specify playback source(0/1/2/3/4)	U/TF/AUX/SLEEP/FLASH
0x0A	Enter into standby – low power loss	
0x0B	Normal working	
0x0C	Reiniciar módulo	
0x0D	Reproducir	
0x0E	Pausa	
0x0F	Specify folder to playback	1~10(need to set by user)
0x10	Volume adjust set	{DH=1:Open volume adjust } {DL: set volume gain 0~31}
0x11	Repeat play	{1:start repeat play} {0:stop play}

3.6. Tecnologías de Software utilizadas

IDE Arduino: Lenguaje de programación propio de Arduino Versión. 1.8.9 de 64 bits para programar las instrucciones en el microcontrolador.

Fritzing: Es un programa libre de automatización de diseño electrónico que busca ayudar a diseñadores y artistas para que puedan pasar de prototipos (usando, por ejemplo, placas de pruebas) a productos finales.

Ni Multisim: Es software Ni Multisim Design Suite 14, este nos permite diseñar el PCB (Placa de Circuito Impreso) y la simulación de todo el esquema del circuito a realizar haciendo pruebas de corrida.

App Inventor 2: El App Inventor 2 es una aplicación de Google Labs para crear aplicaciones Android, estas aplicaciones pueden crearse desde su sitio web y es la que nos permite realizar la interfaz para el usuario para controlar los componentes del sistema de hardware por medio de un Smartphone con sistema operativo Android.

CAPÍTULO IV. CONSTRUCCIÓN DEL MODELO DEL SISTEMA.

4.1. Introducción

En el presente capítulo se muestra el desarrollo y la construcción del controlador para camas eléctricas hospitalarias. Se realiza la descripción de los sensores, actuadores y sistemas de comunicación.

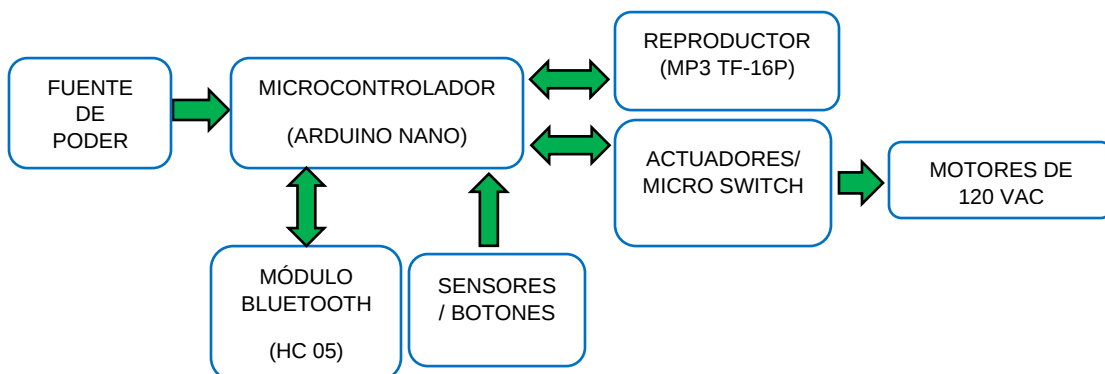
4.2. Modelo del sistema

El proyecto consiste en el desarrollo de software y hardware para el control de camas eléctricas hospitalarias controladas con nuestro Smartphone con sistema operativo Android por medio de la tecnología bluetooth.

El software tiene un temporizador donde el propietario ajusta el tiempo de uso de la cama, permitiendo el uso de todas las funciones de la cama los botones de sus barandales, botones virtuales, el reproductor de música, la alarma y el control de intensidad de luz y al finalizar el tiemer, todas las funciones se deshabilitarán.

En la Figura 4.1 se muestra un diagrama de bloques para el funcionamiento del software y hardware.

HARDWARE



SOFTWARE

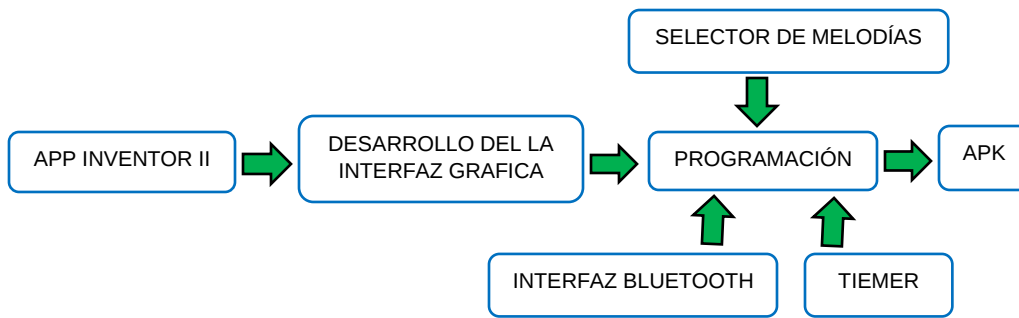


Figura 4.1. Diagrama de bloques del software y hardware empleado.

4.2.1. Materiales

La Tabla 4.1 detalla la lista de los materiales necesarios para la construcción del sistema. El principal elemento es el microcontrolador Atmega 328p.

La mayoría de los componentes complementarios como los leds, resistencias, potenciómetros, bocinas, transistores, etc.... se pueden conseguir en cualquier tienda de electrónica local.

Tabla 4.1. Material utilizado para el diseño del sistema.

Nombre del dispositivo	Cantidad
Relevador de 12v	8
Transistor NPN 2SD882	8
Diodo 1N4007	8
Borneras de 3 entradas	3
Resistencias de 330Ω	8
Memoria micro SD 1GB/2GB	1
Modulo MP3-TF-16P	1
Resistencia de 1KΩ	5
Amplificador de audio LM386	1
Capacitor electrolítico 250μF	2
Capacitor cerámico 0.05 μF	1
Resistencia de 10Ω	1
Preset de 10KΩ	1
Capacitor electrolítico 10μF	1
Arduino nano	1

Modulo Bluetooth HC - 05	1
Buzzer	1
IRLZ44	1
Borneras de 2 entradas	5
Porta fusible	1
Rectificador KBL410	1
Capacitor electrolítico 2200 μ F	2
Regulador LM7812	1
LEDS color verde	2
Resistencia de 10k Ω	1
Transformador reductor de 24VAC con tap central	1
Placa de cobre 15 x 20cm	1

4.2.2. Fuente de poder

En electrónica, una fuente de alimentación es un dispositivo electrónico que convierte la tensión alterna de la red de suministro, en una o varias tensiones, prácticamente continuas, que alimentan los distintos circuitos del equipo electrónico al que se conecta, en este caso al hardware del sistema.

Para la elaboración de este proyecto se diseñó una fuente de CD de 12vcd a 1.2A con protección contra corrientes inversas y cortos circuitos, por medio de diodos y fusibles respectivamente como se muestra en la figura 4.2.

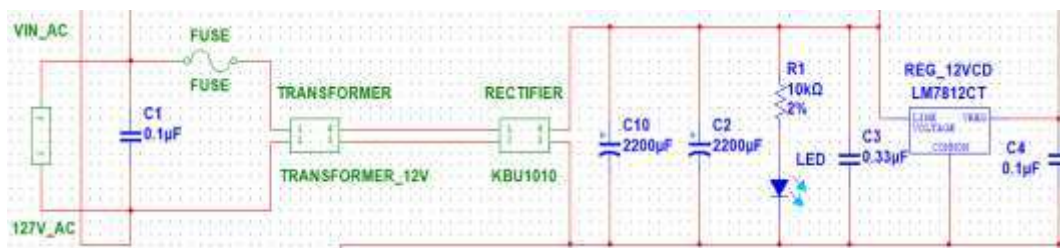


Figura 4.2. Diagrama electrónico de la fuente de poder.

El voltaje de salida de 12vcd a 1.2Amp alimenta todo el sistema, el regulador LM7812 posee un disipador de calor, para evitar

sobrecalentamientos en el sistema y daños posteriores además la fuente está equipada con protección en la entrada y de salida.

4.2.3. Entradas

La cama eléctrica tiene botones en los barandales para elevar o bajar las articulaciones y los micro switch para el paro de los motores, cuando la articulación llega a su máximo o mínimo desplazamiento.

En la Figura 4.3 se muestran los botones pulsadores para el movimiento de las articulaciones.



Figura 4.3. Botones para el movimiento de las articulaciones. [5]

En la programación del microcontrolador se consideran los botones de los barandales funcionando en sincronía con los botones virtuales de la aplicación. Por medio del bluetooth de nuestro Smartphone, el módulo HC-05 recibe una señal de entrada, y con la comunicación UART (Transmisor-Receptor Asíncrono Universal) entre el módulo HC-05 y el microcontrolador Atmega 328p podemos controlar las articulaciones de manera inalámbrica.

4.2.4. Proceso

El microcontrolador Atmega 328p se encarga de la captura de información y procesar todas las entradas digitales que llegan por medio de los sensores, botones y por el módulo Bluetooth, para realizar las acciones programadas.

Este sistema está organizado de tal forma que el microcontrolador es el “eje central” del sistema para protección del microcontrolador se agregaron protecciones en las entradas como en las salidas.

4.2.5. Actuadores

Con base a las señales de entrada, el sistema está programado para efectuar una o varias salidas, este diseño tiene dos tipos de salidas; la primera es una señal de lógica de 0v -5v (Off – On respectivamente) y por medio de un acondicionamiento de señal los transistores NPN activan los relevadores para que éstos energicen los motores de corriente alterna y puedan desplazar las articulaciones; el segundo tipo de salida es una señal audible por medio del módulo MP3 Dfplayer que es reproducida desde una tarjeta micro SD con la música seleccionada.

4.2.6. Conexión del módulo Bluetooth HC - 05

En la Figura 4.4 se muestra el esquema de conexión para el módulo Bluetooth HC-05 con el Arduino nano, permitiendo la configuración del módulo Bluetooth HC-05 con los comandos AT.

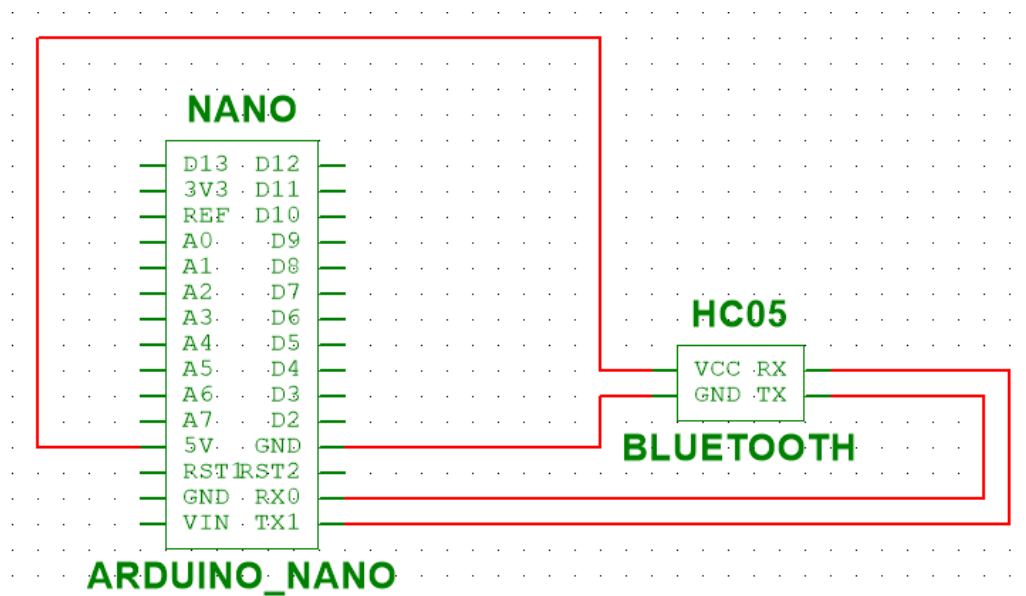


Figura 4.4. Conexión del módulo Bluetooth HC -05 con Arduino.

En la Tabla 4.2 se puede ver cómo queda la configuración final del módulo Bluetooth HC-05 luego de haber sido configurada desde los comandos AT, esta configuración lo que realiza es permitir la modificación de los valores de fábrica del módulo Bluetooth HC-05 y personalizarlos.

Tabla 4.2. Configuración personalizada para el módulo bluetooth.

Nombre	Descripción
Name	BT_BED
Password	123
Role	0 = Slave
UART	9600,0,0 (Baud, StopBit, Parity)
Cmode	1 = Conexión a cualquier dispositivo.

Donde:

Baud equivale a una velocidad, los valores pueden ser: 4800, 9600, 19200, 38400, 57600, 115200, 23400, 460800, 921600 o 1382400.

StopBit es el Bit de parada, puede ser 0 o 1, para 1 bit o 2 bits de parada respectivamente, Para aplicaciones comunes se trabaja con 1 bit por lo que este parámetro normalmente se lo deja en 0.

Parity es la paridad, puede ser 0 (Sin Paridad), 1 (Paridad impar) o 2 (Paridad par). Para aplicaciones comunes no se usa paridad, por lo que se recomienda dejar este parámetro en 0.

4.2.7. Conexión del módulo Dfplayer

En la Figura 4.5 se muestra la conexión que permite la comunicación entre el Arduino nano y el módulo Dfplayer TF-16P para realizar la reproducción de audio mp3.

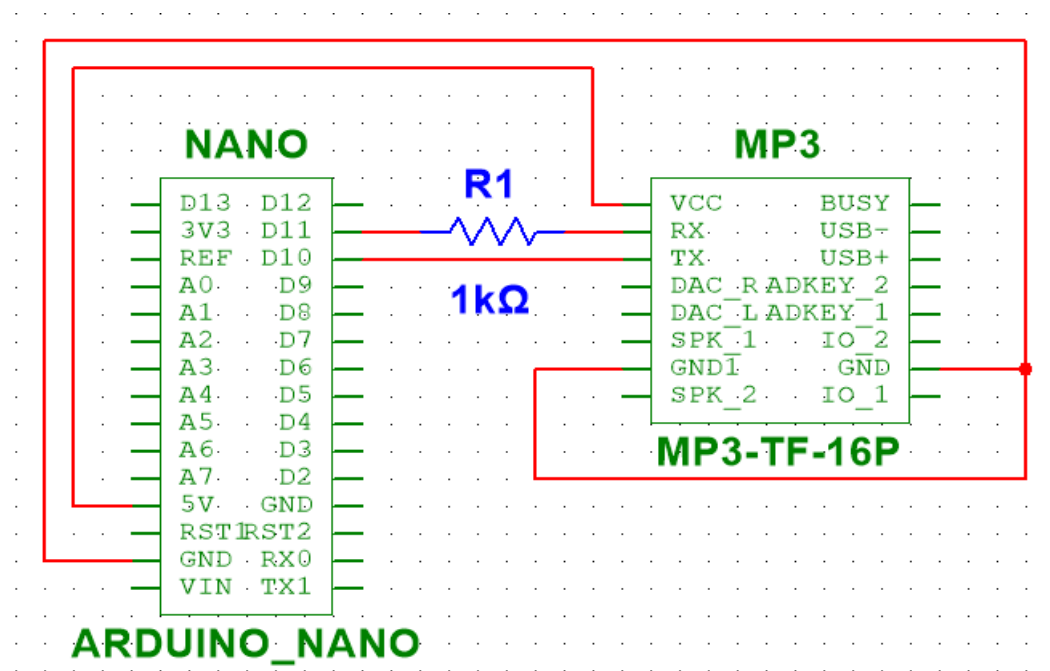


Figura 4.5. Conexión del módulo Dfplayer TF-16P con Arduino nano.

4.2.8. Conexión del amplificador de audio LM386

En la Figura 4.6 se muestra el diagrama de conexión modulo Dfplayer TF-16P y el amplificador LM386, el circuito del amplificador de audio tiene un filtro tipo pasa altas en la entrada, permitiendo que al reproducir la música no se vea afectada por ruido interno.

Cálculo del filtro pasa altas

Diseñar un filtro RC pasa altas permitiendo atenuar una frecuencia de $F_c = 50\text{Hz}$ que provoca la vibración del motor cuando gira, con un valor fijo de $R = 330\Omega$.

$$\text{Ecuación: } F_c = \frac{1}{2\pi RC}$$

$$\text{Calculando: } C_1 = \frac{1}{2\pi R F_c} = \frac{1}{((2\pi)(330\Omega)(50\text{Hz}))} = 9.64\mu F \approx 10\mu F$$

La alimentación del amplificador lm386 proviene de la fuente de poder de 12vcd como se muestra en la Figura 4.2 de la salida del regulador LM7812. El LM7812 es un regulador de voltaje fijo de CD-CD soporta un voltaje en entrada de 31Vcd regulando a 12vcd con una corriente de 1.5Amp. La configuración del amplificador es de una ganancia de 200.

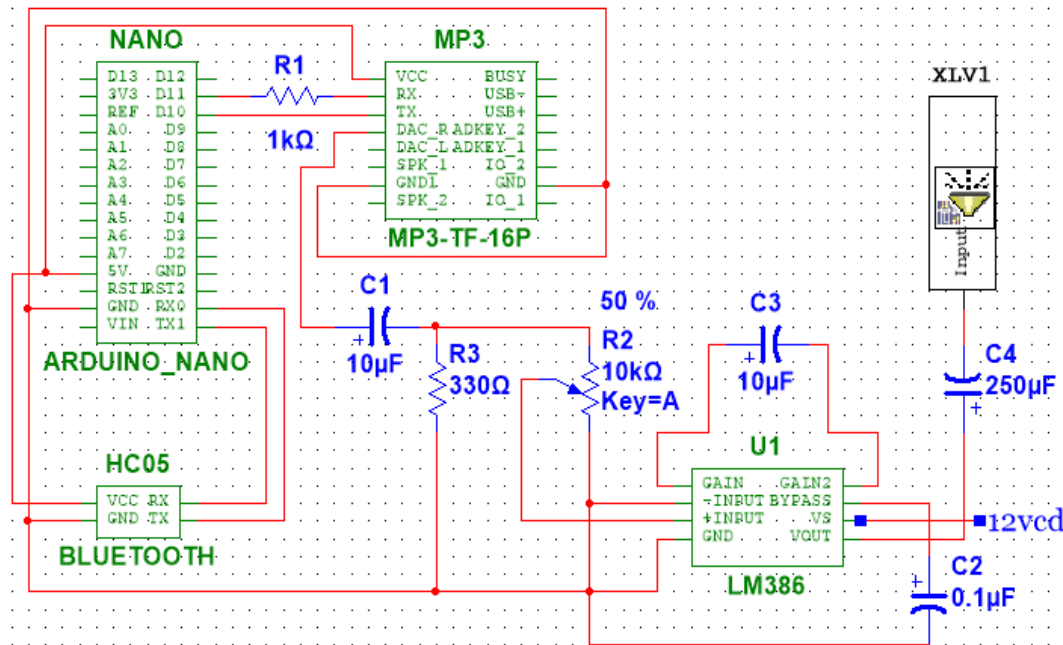


Figura 4.6. Conexión del sistema de audio.

El trimpot R3 nos permite ajustar el volumen del amplificador para dejarlo fijo debido a que en la aplicación podremos variar el volumen de salida.

Todo el sistema esta interconectado y dividido de manera modular, en caso que se presente un problema en algún componente se pueda ubicar fácilmente la falla y repararla o cambiar el componente dañado, por ello todos los componentes se pueden adquirir en una tienda de electrónica.

4.2.9. Conexión del sistema de alarma y la intensidad de luz

En la Figura 4.7 se muestra la conexión entre el Arduino nano y los dos actuadores de salida tales como: el actuador de iluminación led y la activación de la alarma.

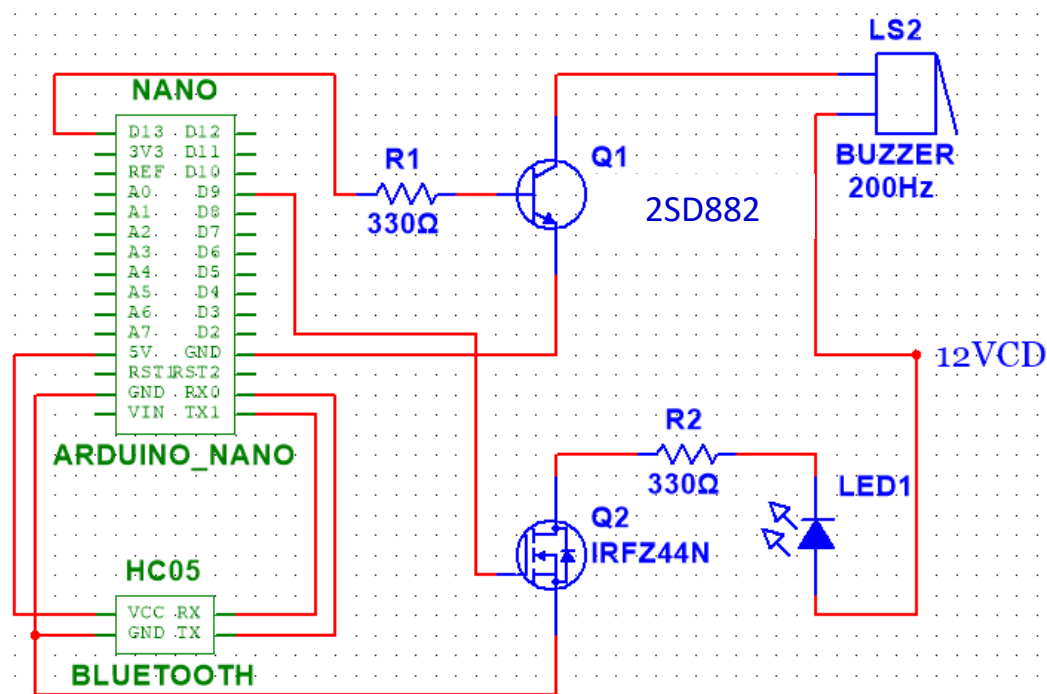


Figura 4.7. Conexión del sistema de alarma e iluminación.

La función de la alarma es avisar al personal que el paciente necesita ayuda urgente, por ellos en la programación se consideró que una vez presionado el botón virtual de la alarma suspenda todas las actividades

que esté realizando el sistema por un periodo de 10 segundos y después seguir operando el sistema de forma normal.

La función del control de iluminación es tener iluminación controlada, es decir que por medio de la aplicación con el mismo botón virtual llamado “luz” obtenga el 20% ,50%, 70% y el 100% de la intensidad de la iluminación; el actuador es un led de potencia que está ubicado por debajo de la cama.

4.2.10. Conexión del sistema de Motores de AC

En la Figura 4.8 se muestra la conexión del motor que permite el desplazamiento de la articulación de la cabeza permitiendo subir o bajar.

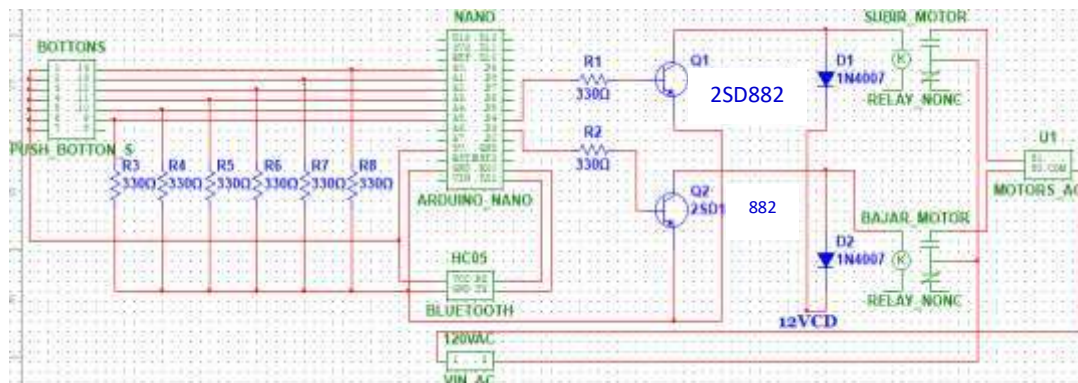


Figura 4.8. Conexión del sistema a un motor de AC.

Solo se muestra el diagrama para la conexión de un motor en este caso el motor de la cabeza, debido a que los demás motores tendrán la misma conexión. En la Tabla 4.3 se muestran los pines que se están utilizando en el Arduino nano tanto como entradas y salidas para controlar el desplazamiento de las articulaciones de la cama eléctrica hospitalaria Hill-Rom Advance Bed.

El conector de color verde que se muestra en la Figura 4.8 alimenta al motor de corriente alterna de la Figura 2.7 donde el pin llamado ‘común’ se conecta al pin de color blanco, el pin llamado ‘subir motor’ se conecta al pin de color rojo y por último el pin llamado ‘bajar motor’ se conecta al pin de color azul del motor.

Tabla 4.3. Configuración de los pines de entrada y salida utilizados para el control de los motores de corriente alterna.

Tipo	Nombre	Pin
Entrada	Botón subir cabeza	A0
Entrada	Botón bajar cabeza	A1
Entrada	Botón subir cama	A2
Entrada	Botón bajar cama	A3
Entrada	Botón subir pies	A4
Entrada	Botón bajar pies	A5
Salida	Motor1 subir cabeza	D3
Salida	Motor1 bajar cabeza	D4
Salida	Motor2 subir cama	D5
Salida	Motor2 bajar cama	D6
Salida	Motor3 subir pies	D7
Salida	Motor3 bajar pies	D8

4.3. Software de la aplicación

Para el desarrollo de esta aplicación se utiliza el software online llamado App Inventor 2 en el cual se diseña cada una de las interfaces de control de los sistemas de alarma, control de iluminación, control de los motores y el reproductor de mp3; en esta aplicación se diseñan los gráficos, botones virtuales, letras, símbolos y otros elementos para el usuario, los requisitos para el desarrollo de esta aplicación son:

- Una computadora.
- Navegador de Internet como Google Chrome.
- Dispositivo Android (Celular, Tablet, etc.).
- Versión de java actualizado en la PC.



Figura 4.9. Editor de bloques de App Inventor 2.

En esta herramienta se definen los componentes de la aplicación con sus respectivas propiedades, así como la interfaz que se utiliza para la comunicación desde el dispositivo móvil por medio de la tecnología bluetooth, posteriormente se da la operación a todos los elementos por medio del editor de bloques que se muestra en la Figura 4.9, donde se define la función de cada uno de los elementos de la aplicación y los datos que se enviarán para el control del sistema inalámbrico.

4.3.1. Interfaz para la activación del producto

En la Figura 4.10 se muestra una pantalla para activar el producto por medio de dos contraseñas, si se digita “123abc” la aplicación se activa como renta o si se digita “abc123” la aplicación se activará como compra. Una vez realizado esta operación, ya no se volverá a mostrar esta pantalla.

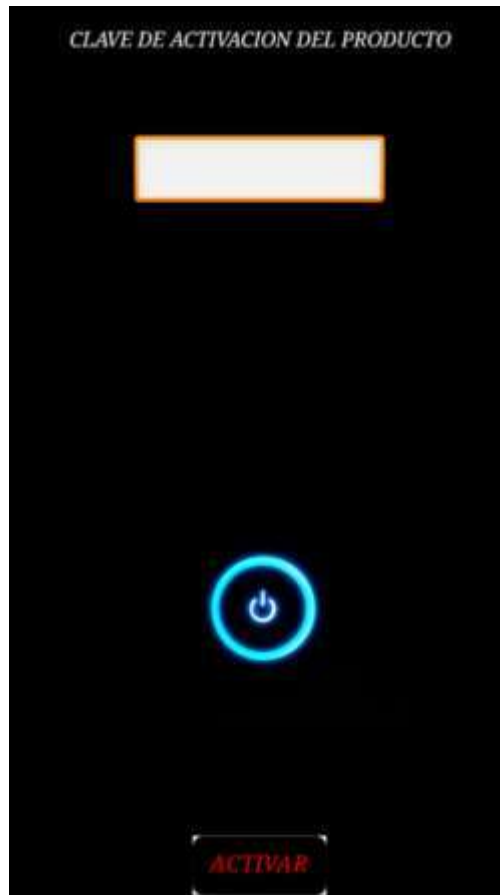


Figura 4.10. Pantalla para activación del producto.

4.3.2. Interfaz de bienvenida

En la Figura 4.11 se ve la pantalla de bienvenida de la aplicación para después iniciar la conexión Bluetooth con nuestro dispositivo móvil.

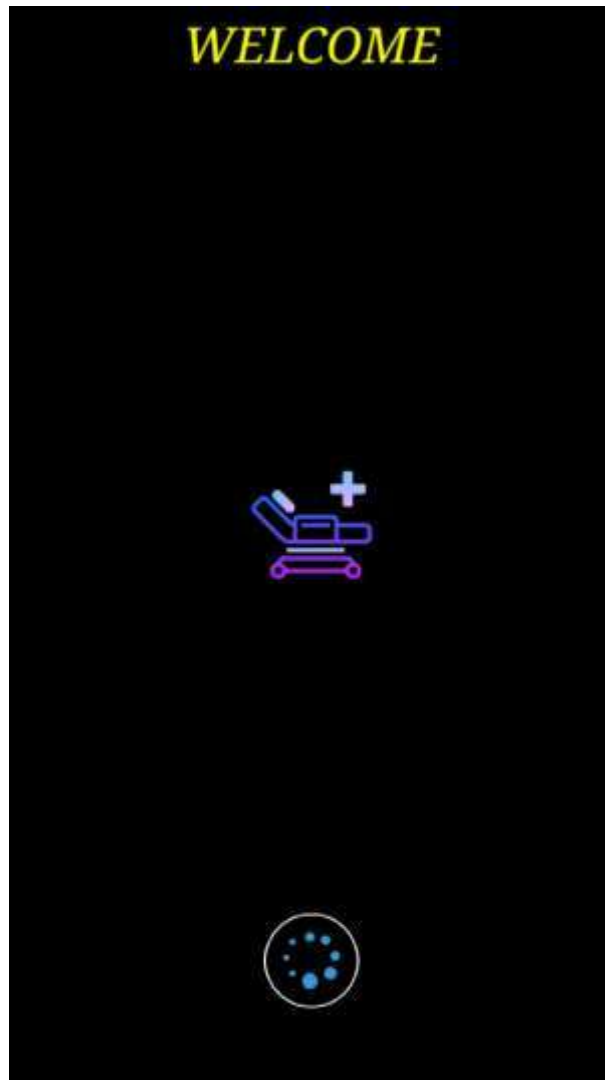


Figura 4.11. Pantalla de bienvenida.

En la pantalla de bienvenida se muestra una imagen de una cama eléctrica y debajo de ella se encuentra una imagen girando, representado que el sistema está cargando durante 4 segundos.

4.3.3. Interfaz de inicio

En la Figura 4.12 se muestra la interfaz de inicio donde se observa la interfaz para el usuario. Con solo darle clic a cualquier icono este abrirá una nueva pantalla permitiendo interactuar con el hardware de la cama.



Figura 4.12. Pantalla de inicio.

Al inicio, el software verifica si está habilitado el bluetooth del smartphone de lo contrario indica en un mensaje de alerta diciendo: “BLUETOOTH DESHABILITADO”.

Después de habilitar el bluetooth tenemos que sincronizarlo con nuestro dispositivo, es decir debemos ir a Configuraciones --> Bluetooth → Buscar dispositivos bluetooth y seleccionar el dispositivo bluetooth llamado “BT_BED” como se muestra en la Tabla 4.2, después solicita una contraseña de emparejamiento la cual es “123” como se muestra es la Tabla 4.2. Por último, volvemos abrir la aplicación y damos clic en el icono “CONECTAR” y seleccionamos en el listado el dispositivo bluetooth

“BT_BED”, después de seleccionarlo el asistente de voz del smartphone nos responde: “CONECTADO CON LA CAMA” y ya podremos utilizar las funciones de la aplicación.

4.3.4. Interfaz del reproductor de audio

Al dar click en el botón música nos muestra la interfaz de la Figura 4.13.



Figura 4.13. Interfaz de audio.

En la interfaz de audio los íconos que se muestran en realidad son carpetas de música donde cada una de ellas consta con 5 canciones mp3 y por medio del reproductor que se encuentra en la parte inferior podremos pausarla, continuar, subir volumen, bajar volumen, siguiente canción y previa canción.

4.3.5. Interfaz de contacto

Cuando le demos clic al ícono llamado contacta nos muestra la interfaz en la Figura 4.14.

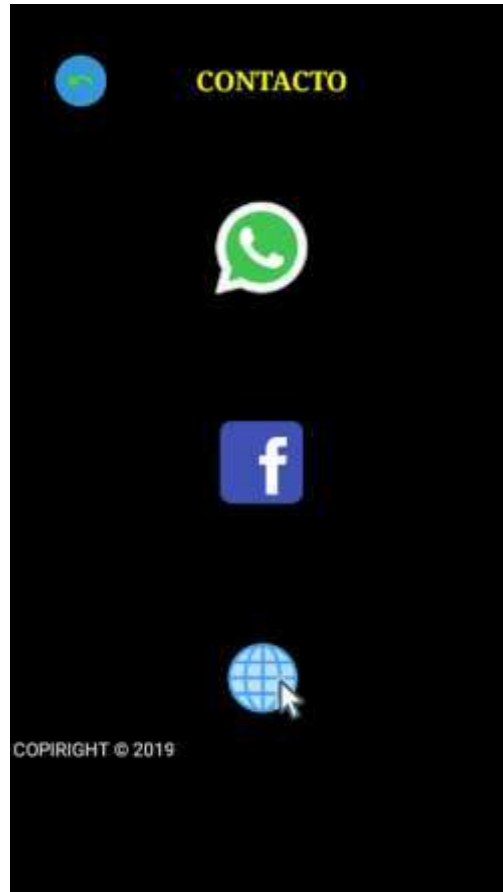


Figura 4.14. Pantalla de contacto.

En esta pantalla podremos ponernos en contacto con el propietario de la cama eléctrica dando clic a cualquier ícono, es decir podremos llamarle, entrar a la página web del propietario o a su Facebook siempre y cuando estemos conectado a internet.

4.3.6. Interfaz de cama

Al dar click al ícono llamado cama nos muestra la interfaz de la Figura 4.15. donde podremos controlar el movimiento de las articulaciones desde nuestro smartphone.

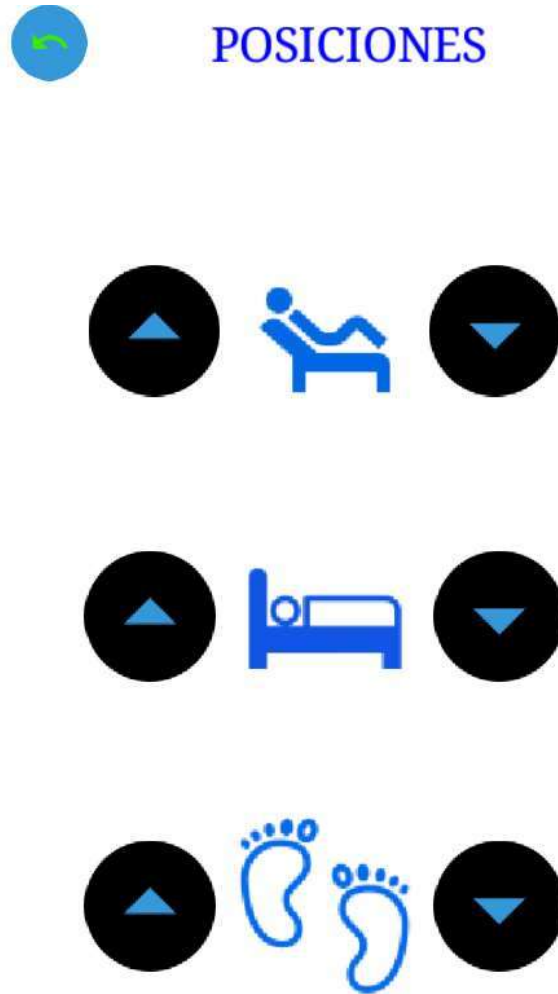


Figura 4.15. Pantalla de posiciones.

En esta interfaz controlamos las posiciones de las articulaciones de la cama por medio de los iconos ya sea para subir o bajar, estos botones virtuales funcionan como push botton, es decir que cuando presionamos cualquier botón, la articulación subirá o bajará y en el momento que soltemos el botón la articulación se detendrá.

CAPÍTULO V. CONCLUSIONES Y TRABAJOS FUTUROS.

5.1. Conclusiones

Para cubrir las expectativas del objetivo general se realizaron pruebas utilizando el prototipo para ir modificando el funcionamiento hasta ser óptimo.

Las pruebas realizadas se realizaron dentro de las instalaciones de la empresa “Equipo Médico”, en una de las camas hospitalarias que tienen en reparación.

Se agregó un diodo 1N4007 en polarización directa en la terminal Vcc del módulo HC-05 debido a que cuando un motor se activaba y después se desactivaba, el módulo se desconectaba debido a una pequeña corriente inversa que circula en la fuente de alimentación y con ello se corrigió el error.

Se logró diseñar e implementar un sistema de control inalámbrico basado en Arduino con una aplicación que por medio de su interfaz se controlan las articulaciones de la cama, la reproducción de música mp3 y cumplir nuestro objetivo general.

En cuanto a los objetivos específicos planteados en el trabajo de investigación, a continuación, se describe el grado de cumplimiento de cada uno de ellos.

- Se desarrolló una aplicación móvil para sistema operativo Android 3.0 en adelante.
- Se controla la intensidad de iluminación, alarma sonora, posicionamiento de las articulaciones, reproductor de música mp3 preseleccionada, todo esto por medio de nuestro teléfono móvil a través de la tecnología bluetooth.

- Se reciben los datos por medio del bluetooth y se procesa a través del microcontrolador Arduino Atmega 328p.
- Se integraron los elementos software, hardware y la aplicación móvil satisfactoriamente.
- Se probó y evaluó el funcionamiento del prototipo del sistema de control inalámbrico.
- Se desarrolló una interfaz de control para el usuario de manera que sea auto suficiente.
- Se brinda el confort que ayuda al usuario a no realizar movimientos bruscos o innecesarios.

5.2. Trabajos futuros

Es posible mejorar la aplicación agregando más funciones como las siguientes: tener una guía de uso rápido cuando el usuario abra por primera vez la aplicación, agregar un icono de información complementaria para que el usuario lea información y comprenda los beneficios de todas las posiciones de la cama y como poder tener un mejor confort y así mejorar la calidad de vida del paciente.

En cuando al hardware, se mejorará el PCB (placa de circuito impreso) con componente SMD (Dispositivo de montaje superficial) para tener una tarjeta pequeña y compacta.

REFERENCIAS

- [1] Wikipedia, «Wikiperia,» 1 Noviembre 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Cama_el%C3%A9ctrica.
- [2] Donurmy, «Donurmy create you habitat,» 21 Enero 2011. [En línea]. Available: <https://www.donurmy.es/blog/historia-de-la-cama/>.
- [3] Universidad de Alicante, «Enfermeria 1014,» 5 Enero 2015. [En línea]. Available: <http://www.enfermeria1014.com/page/Introducci%C3%B3n+y+caracter%C3%ADsticas+de+la+cama>.
- [4] Wikipedia, 20 Diciembre 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Cama_el%C3%A9ctrica.
- [5] Monetmedical, «monetmedical,» 1 Noviembre 2017. [En línea]. Available: <http://www.monetmedical.com/wp-content/uploads/2017/11/Hill-Rom-Advance-Service-Manual.pdf>.
- [6] J. M. Costa, «Enfermeriablog,» 2019. [En línea]. Available: <http://enfermeriablog.com/posicionesdelpaciente/>.
- [7] E. p. Sepulveda, «Procedimientos basicos de enfermeria,» 14 Julio 2007. [En línea]. Available: <http://enfermeriaedwards.blogspot.com/2007/07/mecnica-corporal.html>.
- [8] Venemedia Comunicaciones C.A., «Concepto definición,» 2019. [En línea]. Available: <https://concepto definicion.de/iso/>.
- [9] C. Salud, 26 Octubre 2011. [En línea]. Available: http://www.cenetec.salud.gob.mx/descargas/CETpdf/camas/Cama_Mecanica_Mult_Pos.pdf.
- [10] EcuRed, «Microswitch,» 3 Mayo 2017. [En línea]. Available: <https://www.ecured.cu/Microswitch>.
]
- [11] Mercadolibre, 26 Agosto 2016. [En línea]. Available: https://articulo.mercadolibre.com.ve/MLV-470459413-hill-rom-motor-_JM.
]
- [12] Aprendiendoarduino, «Aprendiendo Arduino,» 23 Enero 2017. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2017/01/23/programacion-arduino-5/>.
]
- [13] TecnoGaming Labs, «Patagoniatec,» 2015. [En línea]. Available: <https://saber.patagoniatec.com/2014/12/arduino-nano-328-arduino-atmega-clon-compatible-arduino-argentina-ptec/>.

- [14] Aprendiendoarduino.wordpress, 30 Junio 2016. [En línea]. Available:
] <https://aprendiendoarduino.wordpress.com/category/operadores/>.
- [15] Naylampmechatronics, «naylampmechatronics.com,» 8 Agosto 2015. [En línea]. Available:
] https://naylampmechatronics.com/blog/24_configuracion-del-modulo-bluetooth-hc-05-usa.html.
- [16] DfRobot, «DFROBOT DRIVE THE FUTURE,» 2019. [En línea]. Available:
] https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299.
- [17] L. G. C. Ramirez, G. S. A. Jimenez y J. M. Carreño, Sensores y actuadores, CDMX: Grupo editorial patria, 2014.
- [18] Wikipedia, «Wikipedia,» 10 Junio 2019. [En línea]. Available:
] <https://es.wikipedia.org/wiki/Arduino>.
- [19] Theengineeringprojects, 16 Noviembre 2018. [En línea]. Available:
] <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>.

ANEXOS

ANEXO A: Código de desarrollo de la Placa Arduino.

```
//ING. JOSUE JOZIC ZETINA MARTINEZ//

#include "Arduino.h"

#include "SoftwareSerial.h"

#include "DFRobotDFPlayerMini.h"

//SERIAL COMUNICATION//

SoftwareSerial mySoftwareSerial(10, 11); // TX, RX

DFRobotDFPlayerMini myDFPlayer;

////////////////////I/O PINS////////////////////

    //MOTOR HEAD UP/DOWNT////

#define HEAD_UP          3

#define HEAD_DOWNT      4

//MOTOR BED UP/DOWNT//

#define BED_UP          5

#define BED_DOWNT      6

//MOTOR FEET//

#define FEET_UP          7

#define FEET_DOWNT      8

//PINES I/O

#define ALARMA          13

#define DIMER           9

int long state = 0; // Variable lectrura dato serial

////////////////////MUSIC////////////////////

boolean STOP = false;

boolean NEXT_SONG;

int volumen = 0;

int count_play = 0;

boolean BUSY;
```

```

void setup(void) {
  ///////////////////////////////////MUSIC MP3////////////////////////////////////
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);
  pinMode (12, INPUT); //PIN BUSY
  if (!myDFPlayer.begin(mySoftwareSerial)) {
    Serial.println("ERROR!!!!");
    while(true);
  }
  Serial.println(F("DFPlayer Mini online."));
  myDFPlayer.volume(15); //Set volume value. From 0 to 30
  ///////////////////////////////////
  //MOTOR1//
  pinMode(BED_UP, OUTPUT);
  pinMode(BED_DOWNT, OUTPUT);
  //MOTOR2//
  pinMode(FEET_UP, OUTPUT);
  pinMode(FEET_DOWNT, OUTPUT);
  //MOTOR3//
  pinMode(HEAD_UP, OUTPUT);
  pinMode(HEAD_DOWNT, OUTPUT);
  Serial.begin(9600);
  //ALARM_AND_LIGHT//
  pinMode(ALARMA, OUTPUT);
  pinMode(DIMER, OUTPUT);
}

```

```

void loop()
{
    if(Serial.available() > 0) {
        state = Serial.read();
    }
    //FOR MOTORS CONTROL//
    switch (state) {
        case 'A':
            delay(100);
            digitalWrite (HEAD_UP, HIGH);
            digitalWrite (HEAD_DOWNT, LOW);
            digitalWrite (BED_UP, LOW);
            digitalWrite (BED_DOWNT, LOW);
            digitalWrite (FEET_UP, LOW);
            digitalWrite (FEET_DOWNT, LOW);
            break;

        case 'B':
            digitalWrite (HEAD_UP, LOW);
            state = 0;
            break;

        case 'C':
            delay(100);
            digitalWrite (HEAD_UP, LOW);
            digitalWrite (HEAD_DOWNT, HIGH);
            digitalWrite (BED_UP, LOW);
            digitalWrite (BED_DOWNT, LOW);
            digitalWrite (FEET_UP, LOW);
            digitalWrite (FEET_DOWNT, LOW);
            break;
    }
}

```

```
case 'D':
digitalWrite (HEAD_DOWNT, LOW);
state = 0;
break;

case 'E':
delay(100);
digitalWrite (HEAD_UP, LOW);
digitalWrite (HEAD_DOWNT, LOW);
digitalWrite (BED_UP, HIGH);
digitalWrite (BED_DOWNT, LOW);
digitalWrite (FEET_UP, LOW);
digitalWrite (FEET_DOWNT, LOW);
break;

case 'F':
digitalWrite (BED_UP, LOW);
state = 0;
break;

case 'G':
delay(100);
digitalWrite (HEAD_UP, LOW);
digitalWrite (HEAD_DOWNT, LOW);
digitalWrite (BED_UP, LOW);
digitalWrite (BED_DOWNT, HIGH);
digitalWrite (FEET_UP, LOW);
digitalWrite (FEET_DOWNT, LOW);
break
```

```

case 'H':
digitalWrite (BED_DOWNT, LOW);
state = 0;
break;

case 'I':
delay(100);
digitalWrite (HEAD_UP, LOW);
digitalWrite (HEAD_DOWNT, LOW);
digitalWrite (BED_UP, LOW);
digitalWrite (BED_DOWNT, LOW);
digitalWrite (FEET_UP, HIGH);
digitalWrite (FEET_DOWNT, LOW);
break;

case 'J':
digitalWrite (FEET_UP, LOW);
state = 0;
break;

case 'K':
delay(100);
digitalWrite (HEAD_UP, LOW);
digitalWrite (HEAD_DOWNT, LOW);
digitalWrite (BED_UP, LOW);
digitalWrite (BED_DOWNT, LOW);
digitalWrite (FEET_UP, LOW);
digitalWrite (FEET_DOWNT, HIGH);
break;

case 'L':
digitalWrite (FEET_DOWNT, LOW);
state = 0;
break;

```

```

//TO ALARM//

case 'M':
myDFPlayer.pause(); //pause the mp3
state = 0;
digitalWrite(ALARMA, HIGH);
delay(1000);
digitalWrite(ALARMA, LOW);
delay(500);
digitalWrite(ALARMA, HIGH);
delay(1000);
digitalWrite(ALARMA, LOW);
delay(500);
digitalWrite(ALARMA, HIGH);
delay(1000);
digitalWrite(ALARMA, LOW);
delay(500);
digitalWrite(ALARMA, HIGH);
delay(1000);
digitalWrite(ALARMA, LOW);
delay(500);
digitalWrite(ALARMA, HIGH);
delay(1000);
digitalWrite(ALARMA, LOW);
delay(500);
digitalWrite(ALARMA, HIGH);
delay(100);
digitalWrite(ALARMA, LOW);
delay(50);
digitalWrite(ALARMA, HIGH);
delay(100);
digitalWrite(ALARMA, LOW);

```

```

if (BUSY == LOW) {
    myDFPlayer.start();
}
state = 0;
break;
//TO LIGHT//
case 'U':
    analogWrite(DIMER, 0);    //OFF
    state = 0;
    break;

case 'V':
    analogWrite(DIMER, 64);    //LOW
    state = 0;
    break;

case 'W':
    analogWrite(DIMER, 128);    //MEDIUM
    state = 0;
    break;

case 'X':
    analogWrite(DIMER, 200);    //MAX_LOW
    state = 0;
    break;

case 'Y':    //100%
    analogWrite(DIMER, 255);    //MAX_MAX
    state = 0;
    break;

```

//////////AUDIO-BOOKS//////////

case 1:

myDFPlayer.playFolder(1, 1); //play specific mp3 in SD:/1/001.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 2:

myDFPlayer.playFolder(1, 2); //play specific mp3 in SD:/1/002.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 3:

myDFPlayer.playFolder(1, 3); //play specific mp3 in SD:/1/003.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 4:

myDFPlayer.playFolder(1, 4); //play specific mp3 in SD:/1/004.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 5:

myDFPlayer.playFolder(1, 5); //play specific mp3 in SD:/1/005.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

/////////////////////////////////TIPS FOR HEALTY////////////////////////////////

case 6:

myDFPlayer.playFolder(2, 1); //play specific mp3 in SD:/2/001.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 7:

myDFPlayer.playFolder(2, 2); //play specific mp3 in SD:/2/002.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 8:

myDFPlayer.playFolder(2, 3); //play specific mp3 in SD:/2/003.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 9:

myDFPlayer.playFolder(2, 4); //play specific mp3 in SD:/2/004.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 10:

myDFPlayer.playFolder(2, 5); //play specific mp3 in SD:/2/005.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

/////////////////INSTRUMENTAL/////////////////

case 11:

myDFPlayer.playFolder(3, 1); //play specific mp3 in SD:/3/001.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 12:

myDFPlayer.playFolder(3, 2); //play specific mp3 in SD:/3/002.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 13:

myDFPlayer.playFolder(3, 3); //play specific mp3 in SD:/3/003.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 14:

myDFPlayer.playFolder(3, 4); //play specific mp3 in SD:/3/004.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 15:

myDFPlayer.playFolder(3, 5); //play specific mp3 in SD:/3/005.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

/////////////////////////////////TERAPIA////////////////////////////////

case 16:

myDFPlayer.playFolder(4, 1); //play specific mp3 in SD:/4/001.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 17:

myDFPlayer.playFolder(4, 2); //play specific mp3 in SD:/4/002.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 18:

myDFPlayer.playFolder(4, 3); //play specific mp3 in SD:/4/003.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 19:

myDFPlayer.playFolder(4, 4); //play specific mp3 in SD:/4/004.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

case 20:

myDFPlayer.playFolder(4, 5); //play specific mp3 in SD:/4/005.mp3; Folder
Name(1~99); File Name(1~255)

state = 0;

break;

}

```

//////////TO PLAY - STOP//////////

if(STOP == true) {
    BUSY = digitalRead(12);    //BUSY MUSIC = 0    NOT MUSIC = 1
    if (BUSY == HIGH) {
        //----Set different EQ----
        myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
        // myDFPlayer.EQ(DFPLAYER_EQ_POP);
        // myDFPlayer.EQ(DFPLAYER_EQ_ROCK);
        // myDFPlayer.EQ(DFPLAYER_EQ_JAZZ);
        // myDFPlayer.EQ(DFPLAYER_EQ_CLASSIC);
        // myDFPlayer.EQ(DFPLAYER_EQ_BASS);
        myDFPlayer.randomAll(); //Random play all the mp3.
    }
}

//////////PLAY/STOP THE MUSIC//////////

if( state == 'O') {
    if(STOP == false) {
        myDFPlayer.start(); //start the mp3 from the pause
        STOP = true;
        state = 0;
    }
}

if (state == 'P') {
    myDFPlayer.pause(); //pause the mp3
    STOP = false;
    state = 0;
}

//////////TO NEXT OT PREVIUS SONG//////////

if(state == 'Q' && STOP == true) {
    myDFPlayer.next(); //Play next mp3
    state = 0;
}

```

```

if(state == 'R' && STOP == true ) {
    myDFPlayer.previous(); //Play previous mp3
    state = 0;
}

//////////TO VOLUMEN UP-DOWNT MAX = 30////////
if(state == 'S' && STOP == true ) {
    for(volumen = 30-15; volumen<= 30; volumen++) {
        myDFPlayer.volumeUp(); //Volume Up
        delay(200);
        state = 0;
        break;
    }
}

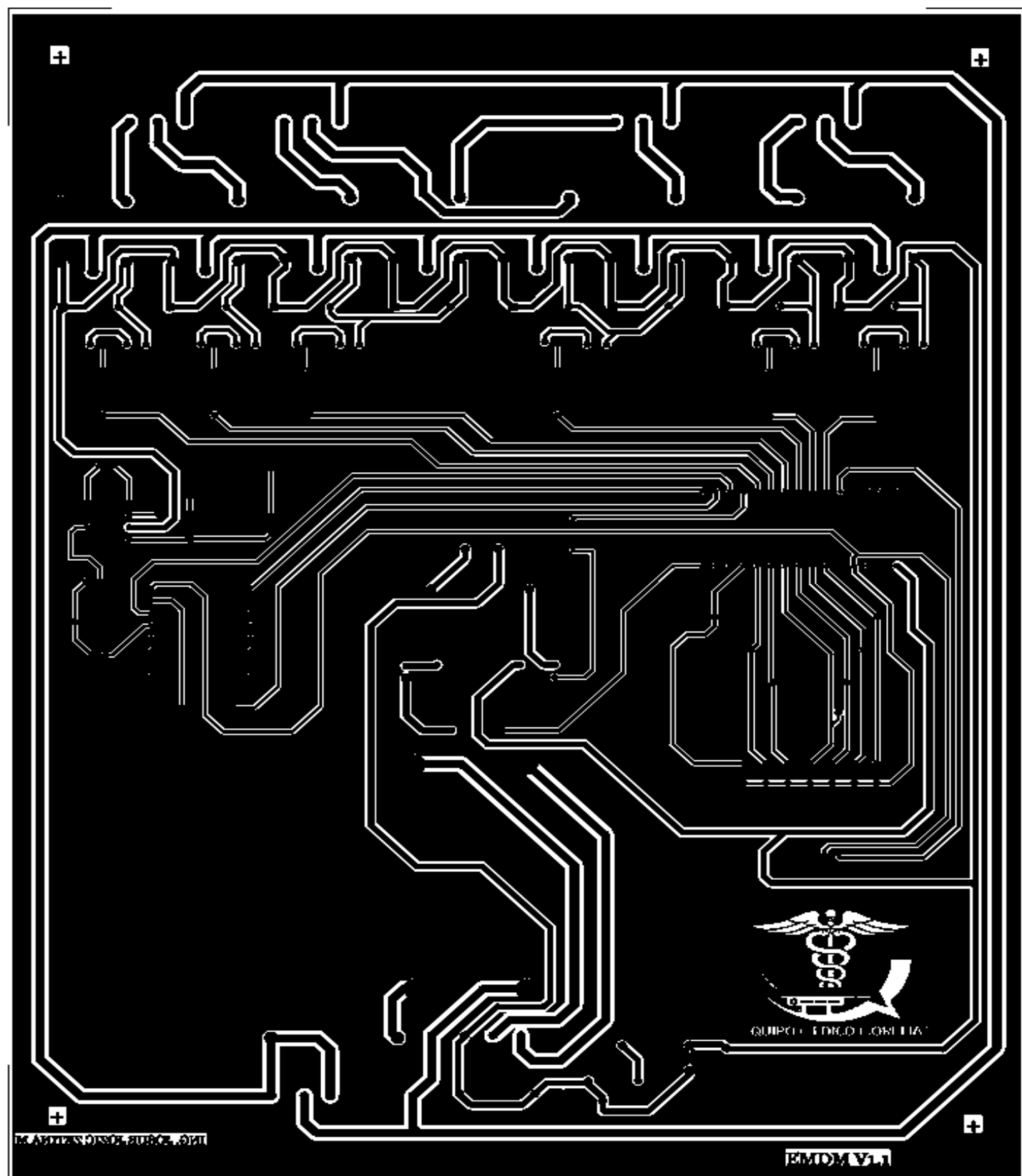
if(state == 'T' && STOP == true ) {
    for(volumen = 30-15; volumen>= 0; volumen--) {
        myDFPlayer.volumeDown(); //Volume Downt
        delay(200);
        state = 0;
        break;
    }
}

delay(50);

} //FIN DEL CODIGO LOOP

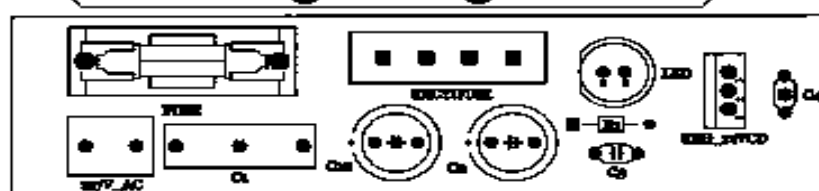
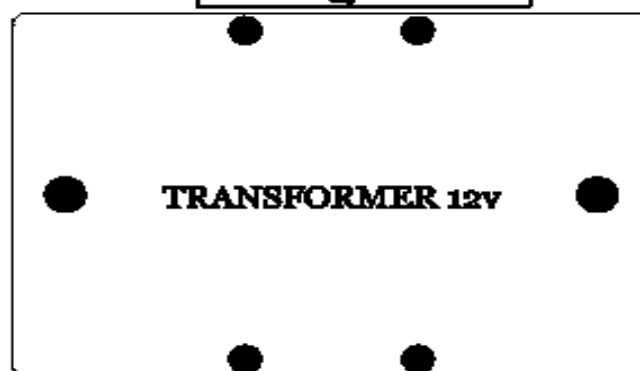
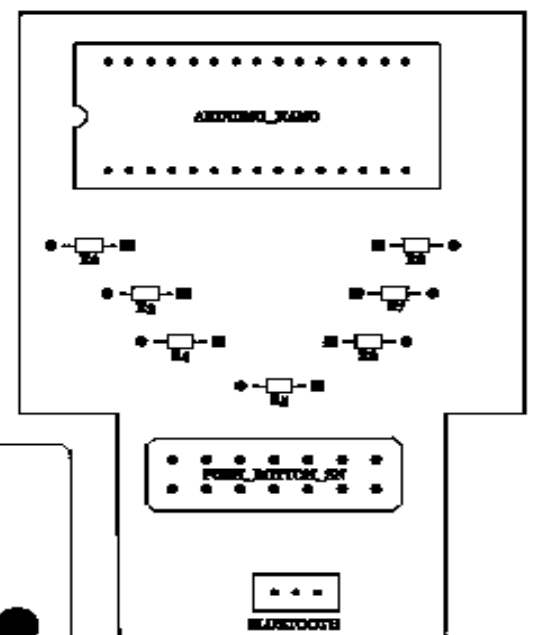
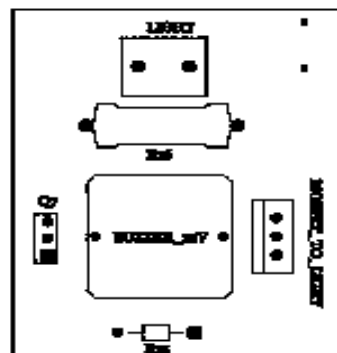
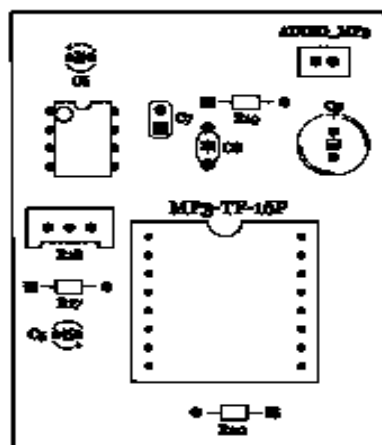
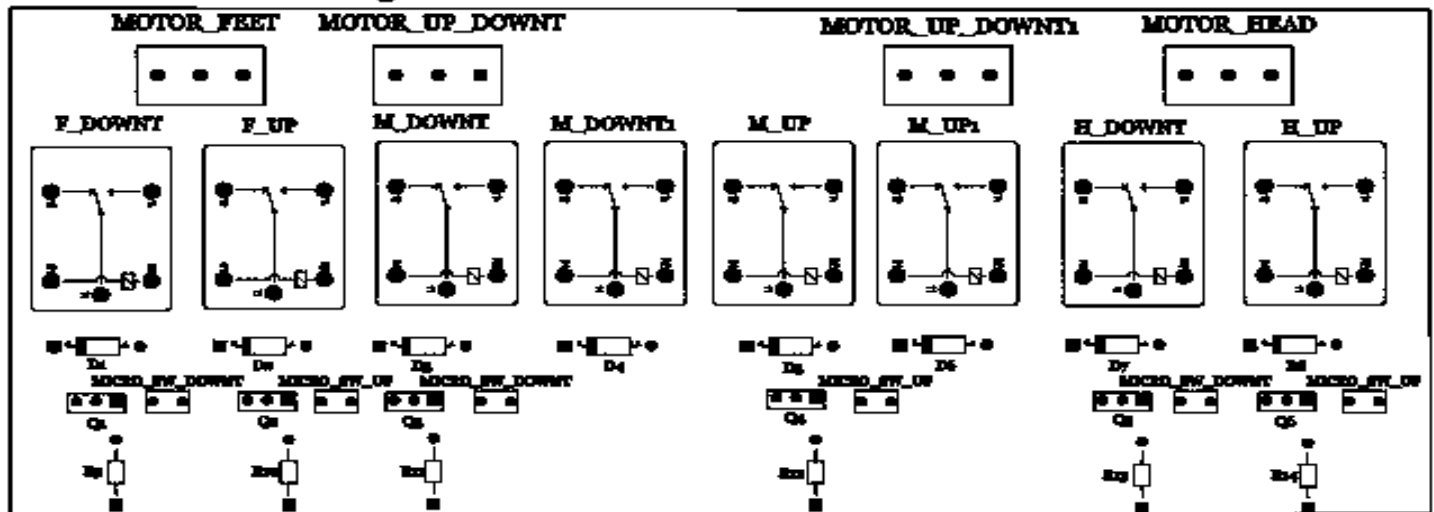
```

ANEXO B: PCB del proyecto (Cooper Botton).



ANEXO C: PCB del proyecto (Cooper Top).

EQUIPO MEDICO DE MORELIA



EQUIPO MEDICO DE MORELIA



ANEXO D: Diagrama electrónico.

