

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN SISTEMA DE DIRECCIONALES PARA UN CASCO DE BICICLETA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Andrés Sebastián Araujo Arroba

andres.araujo01@epn.edu.ec

DIRECTOR: ING. LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.

leandro.pazmino@epn.edu.ec

CODIRECTOR: ING. MÓNICA VINUEZA, MSC.

monica.vinueza@epn.edu.ec

Quito, diciembre 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr Andrés Sebastián Araujo Arroba como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:



Leandro Pazmiño

DIRECTOR DEL PROYECTO

Mónica Vinuesa

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Andrés Sebastián Araujo Arroba con CI: 172728198-0 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



**Andrés Sebastián Araujo
Arroba**

DEDICATORIA

Dedicado a mi familia en general, por ser los mejores, y por ser de ejemplo para mí, me han ayudado en mis metas, y en cada caída, en cada victoria, han estado allí conmigo, por ello dedico esta tesis y mis logros a ellos. Gracias Totales.

AGRADECIMIENTO

A mis padres, Richard e Isabel, quienes han sido de gran guía y ayuda en todo el trayecto de mi vida, por apoyarme y darme siempre el empujón necesario para continuar con mis estudios. A mi hermano, Josué, porque él fue mi primer mejor amigo, y lo seguiré siendo toda mi vida.

También, me gustaría agradecer a los profesores de la ESFOT, quienes han logrado hacer de mí un profesional. A mi tutor el ingeniero Leandro Pazmiño quien me guió en la elaboración de esta tesis.

A mis amigos y compañeros de la universidad que me han ayudado a lo largo de la carrera.

Por último, pero no menos importante, agradecer a Jehová Dios, quien es el único que provee las fuerzas necesarias para sobrellevar cualquier dificultad, y quien da la salida a los problemas.

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN.....	12
1.1	Objetivo general	13
1.2	Objetivos específicos.....	13
1.3	Fundamentos.....	13
	Arduino Nano.....	13
	Sensor acelerómetro giroscopio	14
2	METODOLOGÍA.....	16
2.1	Descripción de la metodología usada	16
	Objetivo 1: Identificar los requerimientos necesarios para la implementación del prototipo	16
	Objetivo 2: Seleccionar el <i>hardware</i> adecuado que cumpla con los requerimientos necesarios para la implementación del prototipo	16
	Objetivo 3: Diseñar el sistema	16
	Objetivo 4: Implementar los dispositivos	16
3	RESULTADOS Y DISCUSIÓN	18
3.1	Requerimientos para la implementación del prototipo.....	18
	Microcontrolador	18
	Dispositivos de red inalámbrica	18
	Diodos LED	19
	Sensor capaz de detectar velocidades angulares.....	19
3.2	Selección del <i>Hardware</i>	19
	Microcontrolador	19
	Dispositivo de red inalámbrica	22
	Diodos LED	24
	Sensor capaz de detectar velocidades angulares.....	26
3.3	Diseño del sistema	27

Diseño de placas PCB	28
Dimensionamiento de las fuentes de alimentación	32
Diseño del sistema de direccionales	34
3.4 Configurar los dispositivos	39
3.5 Pruebas y análisis de resultados	52
3.6 Manual de Uso y Mantenimiento.....	56
Presupuesto del proyecto	57
4 CONCLUSIONES Y RECOMENDACIONES	58
4.1 Conclusiones	58
4.2 Recomendaciones	59
5 REFERENCIAS BIBLIOGRÁFICAS.....	60
ANEXOS.....	65
Código de programación para la configuración de los módulos <i>Bluetooth</i>	65
Código de programación para el sensor MPU6050 y el <i>Bluetooth</i> maestro.....	65
Código de programación para los LEDs matriciales 8x8 y el <i>Bluetooth</i> esclavo...	72
Anexo 1: Certificado de Funcionamiento.....	92
Anexo 2: Planos y Esquemas	¡Error! Marcador no definido.

ÍNDICE DE FIGURAS

Figura 1.1 Arduino Nano y sus pines	13
Figura 1.2 Representación gráfica del giroscopio.....	15
Figura 3.1 Conexión del SPI	21
Figura 3.2 Arduino Nano	22
Figura 3.3 Módulo Bluetooth HC-05.....	24
Figura 3.4 LED matricial 8x8 con MAX7219.....	26
Figura 3.5 Módulo sensor acelerómetro giroscopio MPU6050	27
Figura 3.6 Diagrama de funcionamiento del sistema de direccionales	28
Figura 3.7 Diagrama esquemático del módulo maestro	30
Figura 3.8 Diagrama esquemático del módulo esclavo	30
Figura 3.9 Diseño de la placa PCB del módulo maestro	31
Figura 3.10 Diseño de la placa PCB del módulo esclavo	31
Figura 3.11 Placa del módulo maestro.....	32
Figura 3.12 Placa del módulo esclavo.....	32
Figura 3.13 Diseño de la caja del módulo maestro.....	34
Figura 3.14 Diseño de la caja del módulo esclavo	34
Figura 3.15 Caja del módulo maestro	35
Figura 3.16 Caja del módulo esclavo	36
Figura 3.17 Caja del módulo maestro en el manubrio	37
Figura 3.18 Caja del módulo maestro en el manubrio	37
Figura 3.19 Caja del módulo esclavo en el casco	38
Figura 3.20 Diagrama de flujo del funcionamiento del sistema de direccionales	38
Figura 3.21 Diagrama de flujo de la programación del módulo Bluetooth.....	40
Figura 3.22 Diagrama de flujo de la programación del módulo maestro	49
Figura 3.23 Diagrama de flujo de la programación del módulo esclavo.....	52
Figura 3.24 Monitor serial de la prueba del HC-05	53
Figura 3.25 Monitor serial de los datos obtenidos del MPU6050.....	53
Figura 3.26 Monitor serial de la prueba de giro hacia la derecha	54
Figura 3.27 Monitor serial de la prueba de giro hacia la izquierda.....	54
Figura 3.28 Monitor serial del módulo maestro y esclavo al girar hacia la izquierda...	55
Figura 3.29 Monitor serial del módulo maestro y esclavo al girar a la derecha.....	55
Figura 3.30 Direccional hacia la derecha	56
Figura 3.31 Direccional hacia la izquierda.....	56
Figura 3.32 Código QR del video del manual de uso	57

Figura 3.33 Código QR del manual de mantenimiento	57
Figura 5.1 Peso de la caja maestro sin el circuito	90
Figura 5.2 Peso de la caja esclavo sin el circuito	90
Figura 5.3 Peso de la caja maestro con el circuito	91
Figura 5.4 Peso de la caja esclavo con el circuito	91

ÍNDICE DE TABLAS

Tabla 3.1	Comparación entre el Arduino Uno y el Arduino Nano	20
Tabla 3.2	Comparación entre los módulos HC-05 y HC-06	24
Tabla 3.3	Comparación entre el MPU6050 y el MPU9250.....	26
Tabla 3.4	Conexiones del módulo maestro.....	28
Tabla 3.5	Conexiones del módulo esclavo	29
Tabla 3.6	Consumo de corriente del circuito del módulo maestro.....	33
Tabla 3.7	Consumo de corriente del circuito del módulo esclavo	33
Tabla 3.8	Comparación del peso de la caja del módulo maestro	35
Tabla 3.9	Comparación del peso de la caja del módulo esclavo.....	36
Tabla 3.10	Comandos AT.....	40
Tabla 3.11	Presupuesto del proyecto	57

RESUMEN

El proyecto, sistema de direccionales para un casco de bicicleta, pretende apoyar a los ciclistas a que se mantengan seguros mientras conducen, de modo que el uso de direccionales en el casco del usuario ayudará a que dichos accidentes puedan disminuir. El prototipo usa un sensor acelerómetro giroscopio que detecta el sentido de giro del volante, y enciende las direccionales ubicadas en el casco del conductor mediante conexión inalámbrica *Bluetooth*.

El documento está dividido en 5 secciones, las cuales se detallan a continuación:

En la primera sección se encuentra la introducción, así como el objetivo general y los objetivos específicos, además de teoría relevante que permite comprender el funcionamiento de ciertos componentes que se usan en este proyecto.

En la segunda sección, se detalla la metodología usada para la implementación del prototipo, donde se explica de manera general el cumplimiento de cada uno de los objetivos planteados.

En la tercera sección se presentan los resultados sobre los requerimientos para la implementación del prototipo, la selección del *hardware*, el diseño del sistema de direccionales, la programación de los dispositivos, así como las pruebas realizadas para garantizar el correcto funcionamiento de todo el sistema y videos de un manual de uso y mantenimiento.

En la cuarta sección se encuentran las conclusiones obtenidas del análisis de resultados sobre la implementación del prototipo y también recomendaciones que permitirán en un futuro la mejora del prototipo.

Por último, en la quinta sección se hallan las referencias bibliográficas en las cuales se sustenta la realización de este trabajo de titulación.

PALABRAS CLAVE: giroscopio, sistema de direccionales, módulo *Bluetooth*, comandos AT.

ABSTRACT

The project, a system of directional signs for a bicycle helmet, aims to support cyclists to stay safe while riding, so that the use of directional signs on the user's helmet will help to reduce such accidents. The prototype uses a gyroscope accelerometer sensor that detects the direction of rotation of the steering wheel and turns on the turn signals located on the driver's helmet via Bluetooth wireless connection.

The document is divided into 5 sections, which are detailed below:

In the first section is the introduction, as well as the general objective and the specific objectives, also, relevant theory that allows to understand the operation of certain components that are used in this project.

In the second section, the methodology used for the implementation of the prototype is detailed, where the fulfillment of each of the proposed objectives is explained in a general way.

The third section presents the results on the requirements for the implementation of the prototype, the selection of the hardware, the design of the directional system, the programming of the devices, as well as the tests carried out to guarantee the correct operation of the entire system and videos of a use and maintenance manual.

The fourth section contains the conclusions obtained from the analysis of the results on the implementation of the prototype and recommendations that will allow the improvement of the prototype in the future.

Finally, the fifth section contains the bibliographic references on which the completion of this degree work is based.

KEY WORDS: *Gyroscope sensor, turn signal system, Bluetooth module, AT commands.*

1 INTRODUCCIÓN

En Ecuador, el uso de la bicicleta se ha vuelto una nueva alternativa para movilizarse, así lo indica el INEC (Instituto Nacional de Estadística y Censos) en una encuesta realizada en el año 2016, donde el 1.9% de las personas a nivel nacional usan la bicicleta como medio de transporte, además 608.614 personas usaron más la bicicleta con relación al 2015 [1] y su uso continúa en constante aumento.

Por otra parte, el creciente uso de la bicicleta ha conllevado a que existan varios accidentes de tránsito, por ejemplo, en California (Estados Unidos) se llevó a cabo estudios por parte de La Administración Nacional de Seguridad en el Tráfico de Carreteras (NHTSA, según sus siglas en inglés) la cual mostró los siguientes datos obtenidos en 2015: hubo 818 personas que murieron a causa de lesiones sufridas en accidentes de bicicletas y 45.000 personas resultaron heridas [2]. Así mismo, en el Ecuador se registraron 117 accidentes en los meses de enero a junio del 2020, de los cuales 13 fallecieron [3].

Las cifras anteriores muestran que no se está haciendo lo suficiente por quienes optan por el uso de la bicicleta, y es que los ciclistas son más propensos y vulnerables frente a los accidentes de tránsito. Los ciclistas pueden usar cascos y otros equipos de seguridad, pero esto no siempre los protegerá frente a lesiones o incluso la muerte [2].

Así pues, el manejar una bicicleta presenta riesgos, es por ello por lo que es conveniente realizar este proyecto piloto que propone un sistema mejorado que incluye luces direccionales en un casco de bicicleta, para que los ciclistas tengan oportunidad y facilidad de dar aviso cuando deseen girar hacia alguna dirección.

Lo que pretende este proyecto es apoyar a los ciclistas cuando estos se encuentren manejando sus bicicletas. El sistema de direccionales estará ubicado en el casco del ciclista, lo cual hará que sea más visible para los conductores del entorno, de modo que puedan observar el movimiento que realizará la bicicleta, de esta manera se pretende apoyar a reducir el número de accidentes de ciclistas.

El sistema de direccionales es controlado por dos microcontroladores Arduino, comunicados entre sí mediante conexión *Bluetooth*, configurados como maestro – esclavo. El módulo maestro es el encargado de enviar los datos obtenidos mediante el sensor acelerómetro del cambio de posición que se realiza cuando se gira el volante, una vez obtenidos estos datos, se envía hacia el módulo esclavo, el cual interpreta los datos y enciende el LED matricial correspondiente al giro.

1.1 Objetivo general

Implementar un sistema de direccionales para un casco de bicicleta.

1.2 Objetivos específicos

- Identificar los requerimientos necesarios para la implementación del prototipo.
- Seleccionar el *hardware* adecuado que cumpla con los requerimientos necesarios para la implementación del prototipo.
- Diseñar el sistema.
- Implementar los dispositivos.
- Realizar pruebas de funcionamiento del prototipo.

1.3 Fundamentos

Arduino Nano

Esta placa de Arduino es completa y compacta, el tamaño de la placa del circuito impreso (PCB por sus siglas en inglés *Printed Circuit Board*) es de 18 x 45 (mm) y su peso es de 7 (g) [4], haciéndola útil para proyectos donde se necesite ahorrar espacio y peso, pero sin perder la capacidad de una placa más grande, como la placa Arduino Uno. Utiliza un microcontrolador ATmega328P, cuenta con 22 pines de entrada y salida, 6 pines como PWM, y 8 pines para entradas analógicas. Su voltaje de operación es de 5 (V), y consume 19 (mA) de corriente [4].

En la **¡Error! No se encuentra el origen de la referencia.** se presenta la placa Arduino Nano con sus respectivos pines.

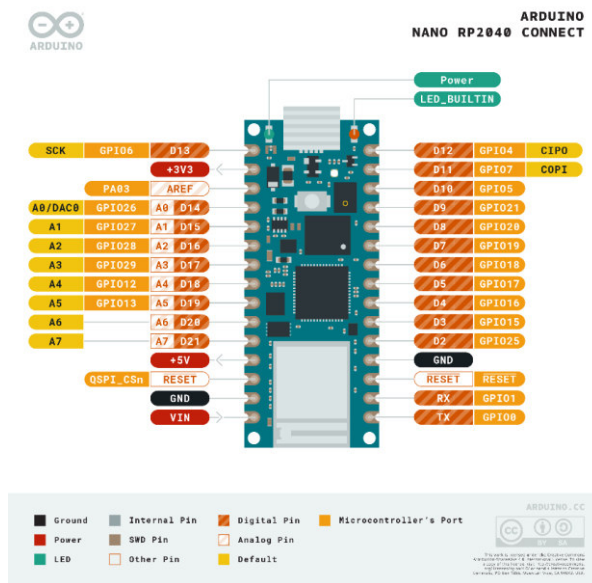


Figura 1.1 Arduino Nano y sus pines [5]

Sensor acelerómetro giroscopio

La aceleración es la variación de la velocidad por unidad de tiempo, es decir:

$$a = \frac{dV}{dt}$$

Ecuación 1.1 Ecuación de la aceleración [6]

Donde:

- a : (m/s²) aceleración
- dV : (m/s) diferencial de velocidad
- dt : (s) diferencial de tiempo

De la misma manera, la segunda Ley de Newton menciona que, en un cuerpo de masa constante, la aceleración de dicho cuerpo es proporcional a la fuerza aplicada sobre éste [6]. Así:

$$a = \frac{F}{m}$$

Ecuación 1.2 Ecuación de la aceleración según la segunda Ley de Newton [6]

Donde:

- a : (m/s²) aceleración
 F : (N) fuerza
 m : (Kg) masa

Este segundo concepto de la aceleración es utilizado en los sensores acelerómetros, los cuales contienen un *Micro Electro Mechanical System* (MEMS), que permite medir la aceleración en un sistema masa – resorte [6].

Por otra parte, el giroscopio mide velocidad angular, la cual es la variación del desplazamiento angular por unidad de tiempo, en otras palabras, qué tan rápido gira un objeto en su propio eje [6].

Es este concepto el que utiliza el giroscopio para medir la velocidad angular gracias al MEMS, usando el efecto Coriolis [6]. El efecto Coriolis indica que, cuando la velocidad angular es constante, existe una fuerza inercial llamada fuerza Coriolis, la cual actúa para desviar un cuerpo hacia los lados solo si dicho cuerpo se mueve en relación con la velocidad angular constante [7]. En la **¡Error! No se encuentra el origen de la referencia.** se representa gráficamente al giroscopio.

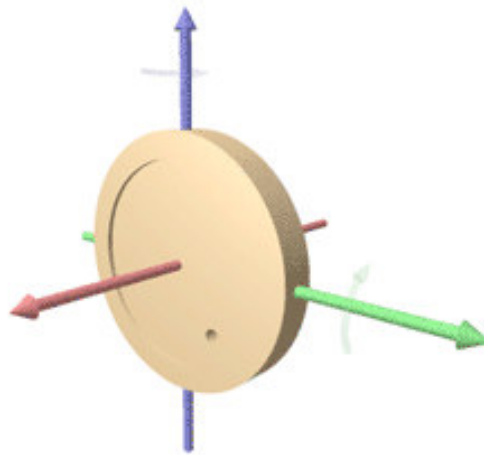


Figura 1.2 Representación gráfica del giroscopio [6]

2 METODOLOGÍA

2.1 Descripción de la metodología usada

Objetivo 1: Identificar los requerimientos necesarios para la implementación del prototipo

Para llevar a cabo este proyecto fue imprescindible definir las necesidades que se requirieron para el funcionamiento del prototipo, tales como: microcontroladores, dispositivos de red inalámbrica, diodos emisores de luz (LED) y un sensor capaz de detectar velocidades angulares. Además, el sistema de direccionales que está ubicado en la bicicleta fue sencillo y se adaptó a cualquier tipo de bicicleta.

Objetivo 2: Seleccionar el *hardware* adecuado que cumpla con los requerimientos necesarios para la implementación del prototipo

Una vez analizados los requerimientos, se seleccionó el *hardware* adecuado que cumpla con los requerimientos necesarios para la implementación del prototipo, por tal motivo, se tomó en cuenta las características técnicas, tamaño, calidad y precios disponibles en el mercado del país.

Se utilizaron los módulos Arduino para el control del sensor y de los diodos LED, además, para la conexión entre microcontroladores se utilizó la tecnología *Bluetooth* mediante módulos de Arduino y el sensor que se usó es un acelerómetro giroscopio.

Objetivo 3: Diseñar el sistema

El diseño que se realizó en la parte del casco debe acoplarse a la mayoría de los cascos para que este sistema pueda ser utilizado. Además, se procuró que el diseño no sea muy pesado y, por lo tanto, que no estorbe ni dificulte al ciclista cuando éste se encuentre manejando.

De igual manera, el diseño para contener la placa Arduino con el acelerómetro debió ser lo más sencillo posible, además, permitió que el sensor pueda tomar medidas sin ninguna interrupción ni tomó datos equivocados, de modo que el diseño y ubicación fueron precisos para su correcto uso.

Objetivo 4: Implementar los dispositivos

La programación de los módulos *Bluetooth* se realizaron en el entorno de desarrollo integrado (IDE) de Arduino, el módulo que está en el volante de la bicicleta será el maestro, este será el encargado de enviar los datos obtenidos del acelerómetro hacia

el módulo del casco, el cual estará constantemente recibiendo los datos del módulo maestro.

Para la programación del sensor acelerómetro, fue necesario utilizar la librería para la comunicación I2C, que es un protocolo de comunicación serial para transferir bits entre dos dispositivos. Este protocolo es más usado para comunicaciones con sensores digitales, ya que su arquitectura permite tener confirmación de datos suministrados por los sensores [8].

3 RESULTADOS Y DISCUSIÓN

El sistema de direccionales está controlado por dos microcontroladores Arduino, uno de ellos está colocado en el manubrio de la bicicleta junto con un sensor acelerómetro giroscopio y un módulo *Bluetooth* maestro; el sensor detecta cambios de velocidad angular cuando el conductor gira el manubrio, los datos son enviados por el módulo *Bluetooth* maestro hacia el otro módulo *Bluetooth* esclavo, el cual está ubicado en el casco del conductor, junto con el microcontrolador Arduino, el cual interpreta los datos enviados del giro del manubrio y activa la animación de los LED matriciales indicando el sentido de giro.

3.1 Requerimientos para la implementación del prototipo

Microcontrolador

Como se mencionó anteriormente, parte del sistema de direccionales estará ubicado en el casco, por lo que es necesario usar un microcontrolador de pequeñas dimensiones y que sea ligero. Además, debe admitir la interfaz periférica serial (SPI), la cual permite una conexión más rápida para la transmisión de datos [9], esta interfaz será útil para la conexión de los LEDs matriciales, y también debe contar con los pines suficientes para la conexión de todo el sistema que se colocará en el casco.

Por otra parte, el microcontrolador debe soportar la transmisión serial ya que el sensor acelerómetro giroscopio constantemente estará enviando datos y el microcontrolador debe procesar dichos datos con rapidez, sin que se llegue a saturar el búfer, por lo general, en microcontroladores Arduino basados en ATmega328 la capacidad del búfer es de hasta 64 bytes de forma predeterminada [10]. En pocas palabras, es necesario un microcontrolador pequeño, pero con gran capacidad que permita controlar todo el sistema de direccionales sin ningún problema.

Dispositivos de red inalámbrica

La conexión entre microcontroladores debe ser inalámbrica, porque el uso de cables para la conexión entre dichos microcontroladores hará que no sea cómodo para quien use el sistema de direccionales, ya que se necesitará varios cables desde el manubrio de la bicicleta hasta el casco del conductor. Para lo cual, el dispositivo de red inalámbrica debe permitir el envío y recepción de datos a corta distancia, además debe operar en la banda de frecuencias de 2.4 (GHz) y 5 (GHz), dado que este rango de frecuencias admite el libre uso del espectro radioeléctrico.

Diodos LED

El uso de diodos LED en este proyecto permitirá grandes ventajas con respecto a otros tipos de luminarias como pequeñas lámparas incandescentes. Algunas de sus ventajas son: gran duración ya que tienen un pequeño consumo de corriente, su tamaño es reducido y presentan una gran resistencia frente a impactos y vibraciones [11]. Estos dos últimos puntos serán de gran utilidad en el sistema de direccionales ya que el tamaño reducido representa también un bajo peso, ideal para la colocación de las direccionales en el casco, y la alta resistencia frente a impactos y vibraciones permite que los diodos LED funcionen correctamente pese al gran uso que se le pueda dar.

A la agrupación de diodos LED en matrices se lo conoce con el nombre de LED matriciales, estos LED son utilizados en algunos de los faros de carros, ya que permiten tener una mejor distribución de la luz [12], además de poder activar LED individuales permitiendo crear formas con ciertos LED encendidos. En el mercado existen varios tamaños de LED matriciales, para lo cual será necesario utilizar un tamaño adecuado que permita la colocación en el casco y que sobre todo permita la visualización del sentido de giro.

Sensor capaz de detectar velocidades angulares

Este sensor debe ir ubicado en el manubrio de la bicicleta por lo que debe detectar estos cambios en el giro del volante, ya sea izquierda o derecha, para que pueda activar la animación del LED matricial respectivo según el giro. Ahora bien, el sensor debe detectar el cambio en el eje X y poder convertir el ángulo de giro en grados para facilitar la calibración en el ángulo de giro adecuado. Por otro lado, este sensor debe ser compatible con el microcontrolador a utilizar ya que estará enviando datos constantemente.

Además, el sensor debe ser pequeño para que pueda obtener datos correctos al estar ubicados en el manubrio, por lo que si fuese un sensor muy grande no se podría colocar en dicho manubrio y por lo tanto tomar datos equivocados.

3.2 Selección del *Hardware*

Microcontrolador

Existen una gran cantidad de microcontroladores Arduino, como el Arduino Uno, el Arduino Mega o el Arduino Nano, sin embargo, cada uno de estos microcontroladores juegan un papel fundamental dentro de un proyecto en específico, pues poseen

características propias que los hacen mejores o peores en la implementación de un proyecto.

En la Tabla 3.1 Comparación entre el Arduino Uno y el Arduino Nano se presenta un cuadro comparativo con las principales características del Arduino Uno y el Arduino Nano.

Tabla 3.1 Comparación entre el Arduino Uno y el Arduino Nano [4] [13] [14]

Características	Uno	Nano
Microcontrolador	ATMega328P	ATMega328P
Voltaje de operación	5 (V)	5 (V)
Pines de E/S digitales	14	14
Pines de E/S analógicas	6	8
EEPROM	1 (KB)	1 (KB)
Consumo de corriente	20 (mA)	19 (mA)
Velocidad de reloj	16 (MHz)	16 (MHz)
Dimensiones del PCB	68,6 x 53,4 (mm)	18 x 45 (mm)
Peso	25 (g)	7 (g)

Mediante la Tabla 3.1 *Comparación entre el Arduino Uno y el Arduino Nano* se puede concluir que a pesar de que el Arduino Nano es más pequeño y menos pesado, posee las mismas características que el Arduino Uno, por lo que no pierde utilidad frente a una placa más grande.

En cuanto a las funcionalidades de los pines del Arduino, será necesario que admitan la comunicación SPI, pues su tipo de arquitectura maestro – esclavo permite iniciar la comunicación con varios dispositivos y recibir datos de estos [15].

El bus de comunicación SPI tiene dos líneas independientes, lo que permite que el maestro envíe datos al esclavo y el esclavo envíe datos al maestro simultáneamente; es decir, la comunicación es full dúplex [15]. También SPI posee un bus síncrono, lo que

significa que mantiene todos los dispositivos sincronizados gracias a una señal de reloj, por lo tanto, existe una reducción en la complejidad al enviar y recibir datos [15].

Para la conexión del SPI, es necesario un mínimo de 3 líneas de comunicación. La comunicación se presenta en la Figura 3.1 Conexión del SPI .



Figura 3.1 Conexión del SPI [15]

El bus *master-out, slave-in* (MOSI) sirve para la comunicación desde el maestro hacia el esclavo. El bus *master-in, slave-out* (MISO) sirve para la comunicación desde el esclavo hacia el maestro. Por último, el *serial clock* (SCK) es la señal del reloj proveniente del maestro para la sincronización de los dispositivos [15]. Estos buses son necesarios para la comunicación SPI. Sin embargo, existe un cuarto bus el *slave select* (SS) que se lo utiliza cuando existen varios dispositivos esclavos conectados y se necesite seleccionar uno en específico para realizar la comunicación [15].

Ambas placas Arduino cuentan con dos pines para transmisión y recepción serial, un pin de transmisión (TX) y otro pin para recepción (RX), además cuenta con los pines 10 (SS), 11 (MOSI), 12 (MISO) y 13 (SCK) [4] [14].

En síntesis, la placa ideal para poder desarrollar este proyecto es la del Arduino Nano, ya que debido a su reducido tamaño y peso es ideal al momento de ahorrar espacio y que su peso no sea un inconveniente para quienes usen el sistema de direccionales en el casco.

Ahora bien, será necesario utilizar dos placas Arduino, la primera llamada módulo maestro, que irá ubicada en el manubrio de la bicicleta y la otra placa Arduino, llamada módulo esclavo, irá ubicada en el casco del conductor. En la Figura 3.2 Arduino Nano se muestra al Arduino Nano.

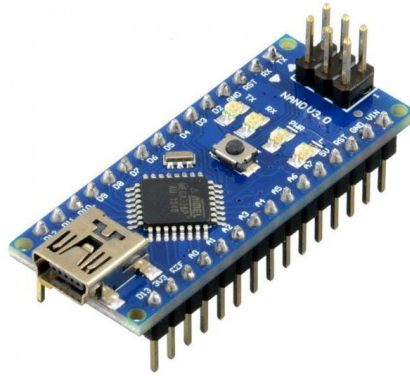


Figura 3.2 Arduino Nano [13]

Dispositivo de red inalámbrica

Las redes de área personal o *Wireless Personal Area Network* (WPAN) son redes inalámbricas de corto alcance que abarcan un área no mayor a una decena de metros, por lo que su alcance es ilimitado.

Existen algunos estándares dentro de las redes WPAN como ZigBee, *Bluetooth* o incluso dispositivos que utilicen la tecnología infrarroja, sin embargo, hay aspectos que se deben considerar como la cantidad de información que se envía, el consumo energético y la velocidad de transmisión de cada tecnología utilizada en las redes WPAN.

Al respecto, ZigBee es muy bueno para proyectos donde se necesite el ahorro de energía y la baja velocidad de transmisión, haciéndolo perfecto para proyectos domóticos, además se los puede utilizar en una topología de red tipo malla, pues permite la conexión entre múltiples dispositivos ZigBee [16]. Por otra parte, *Bluetooth* no posee ahorro de energía, sin embargo, el consumo energético no es tan alto en comparación con ZigBee, pues *Bluetooth* consume 40 (mA) y ZigBee 30 (mA) cuando está transmitiendo y en cuanto a la velocidad de transmisión, *Bluetooth* puede alcanzar el 1 (Mbps) frente a los 250 (kbps) de ZigBee [17], siendo *Bluetooth* 4 veces más rápido que ZigBee.

Lo que se pretende al escoger un dispositivo de red inalámbrica es eliminar las conexiones mediante cables y de esta manera hacer que el sistema de direccionales sea más cómodo en su uso.

La comunicación entre microcontroladores Arduino se realizará por *Bluetooth*, que es una especificación industrial estandarizada en la norma IEEE 802.15.1. El estándar *Bluetooth* utiliza la radiofrecuencia de 2,4 (GHz) en la banda no licenciada *Industrial*,

Scientific and Medical (ISM) esto indica que se puede utilizar abiertamente sin necesidad de pagar por el uso del espectro radioeléctrico [18].

La ventaja que brinda *Bluetooth* frente a otras conexiones de red inalámbricas es que permite establecer comunicación con dispositivos cercanos que se encuentren a pocos metros de distancia, sin que exista la necesidad de tener un control explícito de parte del usuario, direccionamientos de red y permisos, lo que son típicos de otras redes de conexión inalámbrica [18]. Además, su alta velocidad de transmisión permite tiempos de respuesta más rápidos.

Existen estándares de *Bluetooth* llamados “perfiles de dispositivos”, estos “perfiles” determinan la manera de comunicarse entre dispositivos *Bluetooth*, por lo que les permite actuar de una manera u otra según el perfil que se les dé [18].

Dentro de Arduino se utilizan dos perfiles: el primero es el *Serial Port Profile* (SPP) que sirve para establecer una comunicación serial con otros dispositivos, tal como una comunicación transmisión – recepción asíncrona universal (UART). El segundo perfil es el *Human Interface Device Profile* (HID) el cual permite que un dispositivo pueda comportarse como un dispositivo de entrada, tal como un teclado, un ratón o un *joystick* [18] [19].

El uso de módulos *Bluetooth* para establecer la comunicación entre microcontroladores Arduino permite poderlos configurar como maestro y esclavo mediante comandos AT, de esta manera se puede establecer quien envía y recibe datos.

En el mercado existen dos módulos *Bluetooth* para placas Arduino, el HC – 05 y el HC – 06, en la Tabla 3.2 Comparación entre los módulos HC-05 y HC-06 se muestra una comparación entre ambos módulos.

Tabla 3.2 Comparación entre los módulos HC-05 y HC-06 [20] [21]

Características	HC - 05	HC - 06
Voltaje de operación	3.6 (V) a 6 (V)	3.6 (V) a 6 (V)
Consumo de corriente	30 (mA) a 50 (mA)	30 (mA) a 50 (mA)
Frecuencia de operación	2.4 (GHz)	2.4 (GHz)
Modulación	GFSK	GFSK
Potencia de transmisión	4 (dBm)	4 (dBm)
Distancia	Hasta 10 (m)	Hasta 10 (m)
Dimensiones	4.4 x 1.6 x 0.7 (cm)	4.4 x 1.6 x 0.7 (cm)
Configuración	Maestro / esclavo	Esclavo

Como se pudo observar en la Tabla 3.2 Comparación entre los módulos HC-05 y HC-06 existe solo una diferencia entre el módulo HC – 05 y el módulo HC – 06, donde el módulo HC – 05 sí puede ser configurado como maestro y esclavo. Por esta razón, lo vuelve el más apto para el desarrollo de este proyecto al momento de configurar el módulo que irá en el manubrio de la bicicleta como maestro y al módulo que va en el casco como esclavo, de esta manera el módulo *Bluetooth* maestro es quien podrá enviar datos hacia el módulo *Bluetooth* esclavo. En la Figura 3.3 *Módulo Bluetooth HC-05* se muestra una representación del módulo *Bluetooth* HC – 05.

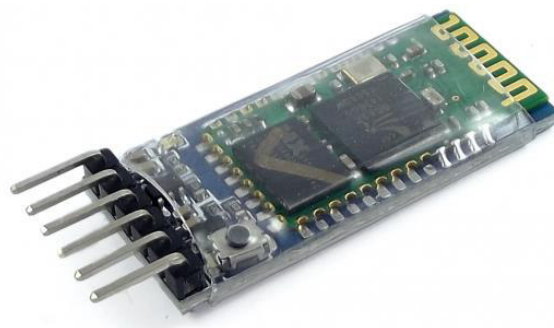


Figura 3.3 Módulo *Bluetooth* HC-05 [22]

Diodos LED

En el análisis sobre los diodos LED, se mencionó el uso de los LED matriciales y las ventajas que presentan estos en su uso. En este caso, los LED matriciales que existen en el mercado solo se diferencian por los números de LED presentes dentro de una matriz, por ejemplo, el de 64 LED es de 8 x 8, el de 256 LED es de 8 x 32. En vista de que, entre más números de LED en la matriz, esta será de mayor dimensión, por lo tanto, mayor peso, lo que se traduce en menos comodidad al usar el sistema de direccionales en el casco.

Por consiguiente, los LED matriciales que se usarán para este proyecto son de tamaño 8 x 8, es decir, 8 filas por 8 columnas ya que al no ser ni tan grandes ni tan pequeños, son ideales para la ubicación en el casco del conductor y visibles para las indicaciones de las direccionales.

Hay que tomar en cuenta que la utilización de LED matriciales implica el uso de varios pines del microcontrolador Arduino, por ejemplo, para los de 8 x 8 es necesario el uso de 16 pines para el control individual de cada diodo LED de la matriz, por lo que es necesaria la utilización de un dispositivo de control.

Este dispositivo de control es el circuito integrado MAX7219, el cual es capaz de controlar 64 diodos LED independientes dentro de una matriz. El circuito integrado tiene un decodificador BCD code-B, realiza multiplexado y posee una memoria RAM estática de 8 x 8 para almacenamiento de cada dígito [23] [24]. La memoria RAM estática presenta grandes ventajas pues utiliza menos energía y puede realizar cálculos con más rapidez que la memoria RAM dinámica [25].

Otro rasgo importante del MAX7219 es que, con solo 3 pines mediante la interfaz SPI se puede controlar toda una matriz de LED, lo que permite la conexión en cascada por lo que se pueden conectar varios MAX7219, por lo tanto, se puede controlar varias matrices de LED con los mismos 3 pines conectados al microcontrolador Arduino. En cuanto a sus características, posee un modo bajo consumo con 120 (uA) [26], el voltaje de operación es desde los 4.7 (V) hasta los 5.3 (V) y las dimensiones de la PCB es de 32 x 50 (mm) [27].

En el mercado del país, existen LED matriciales que ya incorporan el controlador MAX7219 en una PCB, tal como se observa en la Figura 3.4 LED matricial 8x8 con MAX7219 .

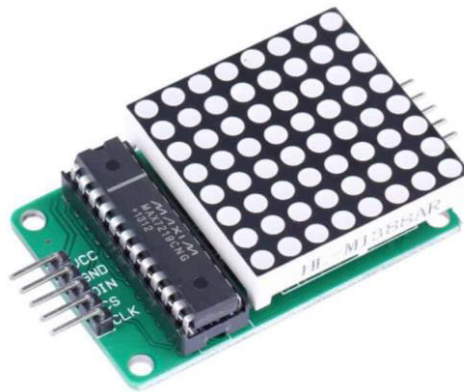


Figura 3.4 LED matricial 8x8 con MAX7219 [28]

Sensor capaz de detectar velocidades angulares

En cuanto al sensor capaz de detectar velocidades angulares, se puede establecer la comparación entre dos sensores que más se utilizan en el mercado del país. Estos son el MPU6050 y el MPU9250. En la Tabla 3.3 Comparación entre el MPU6050 y el MPU9250 se muestra una comparación entre sus principales características.

Tabla 3.3 Comparación entre el MPU6050 y el MPU9250 [29] [30]

Características	MPU6050	MPU9250
DOF	6	9
Comunicación	SPI e I2C	SPI e I2C
Voltaje de operación	2.4 (V) - 3.6 (V)	2.4 (V) - 3.6 (V)
Consumo de corriente	3.5 (mA)	3.5 (mA)
Procesador interno	DMP	DMP

Como se observa en la Tabla 3.3 la diferencia entre el MPU6050 y el MPU9250 radica en que este último posee 3 grados de libertad (DOF) más que el MPU6050 debido a que posee un magnetómetro además del acelerómetro y giroscopio [30], por lo que hace que su precio sea mucho más elevado que el MPU6050. Sin embargo, ya que en este proyecto será necesario únicamente el acelerómetro y el giroscopio, no se utilizará el MPU9250.

Por consiguiente, se usará el sensor MPU6050 el cual posee un sensor acelerómetro giroscopio. El sensor es capaz de medir la velocidad angular y la aceleración en los 3 ejes: X, Y y Z [6]. La comunicación entre el microcontrolador Arduino y el sensor es mediante I2C, los pines SCL y SDA del sensor MPU6050 vienen con resistencias *pull-*

up lo que le permite la conexión directa de los pines al microcontrolador [6]. El pin SDA sirve para la transmisión de datos y el pin SCL para la señal de reloj. El rango del acelerómetro puede ser ajustado a $\pm 2g$, $\pm 4g$, $\pm 8g$, y $\pm 16g$, el del giroscopio a ± 250 , ± 500 , ± 1000 , and ± 2000 ($^{\circ}/s$). También posee un procesador interno (DMP) que puede ejecutar complejos algoritmos para no usar filtros exteriores [29].

Para el uso y control del sensor MPU6050 será necesario el uso de una librería adicional para la comunicación I2C, lo que permitirá el mejor envío de datos del sensor y que el búfer del microcontrolador no se sature. Y otra librería adicional que permitirá una mejor calibración del sensor. En la Figura 3.5 *Módulo sensor acelerómetro giroscopio MPU6050* se muestra al sensor acelerómetro giroscopio MPU6050.

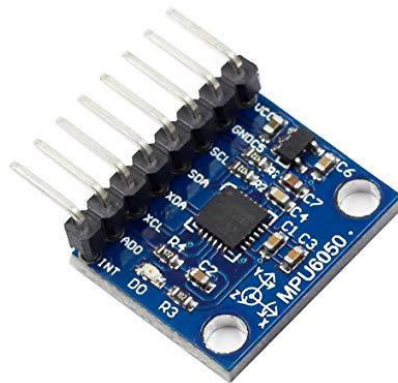


Figura 3.5 Módulo sensor acelerómetro giroscopio MPU6050 [31]

3.3 Diseño del sistema

Para realizar el diseño del sistema, se divide en dos secciones, el primero que es el sistema del manubrio y el segundo es el sistema del casco. En cuanto al sistema del manubrio, se encuentra un Arduino Nano maestro, un sensor MPU6050 y un módulo *Bluetooth* HC – 05 maestro. El sensor está conectado al Arduino Nano mediante la comunicación I2C, mientras que el módulo *Bluetooth* está conectado mediante comunicación serial.

El sistema del casco está conformado por un Arduino Nano esclavo, un módulo *Bluetooth* HC – 05 esclavo y dos LED matriciales 8x8 con MAX7219. El módulo *Bluetooth* está conectado al Arduino Nano por comunicación serial, mientras que el LED matricial izquierdo está conectado directamente por la interfaz SPI y el LED matricial derecho está conectado en cascada con el LED matricial izquierdo, también por SPI.

En el sistema del manubrio, el sensor detectará cambios en la aceleración cuando el conductor gire el manubrio, ya sea derecha o izquierda, envía la información al

microcontrolador, quien, a su vez, envía dicha información del giro por el módulo *Bluetooth* maestro. Por otro lado, en el sistema del casco, el módulo *Bluetooth* esclavo recibe la información enviada y la pasa al microcontrolador, el cual activará la animación del LED matricial correspondiente al giro realizado.

En la Figura 3.6 se muestra un diagrama general del funcionamiento del sistema de direccionales.

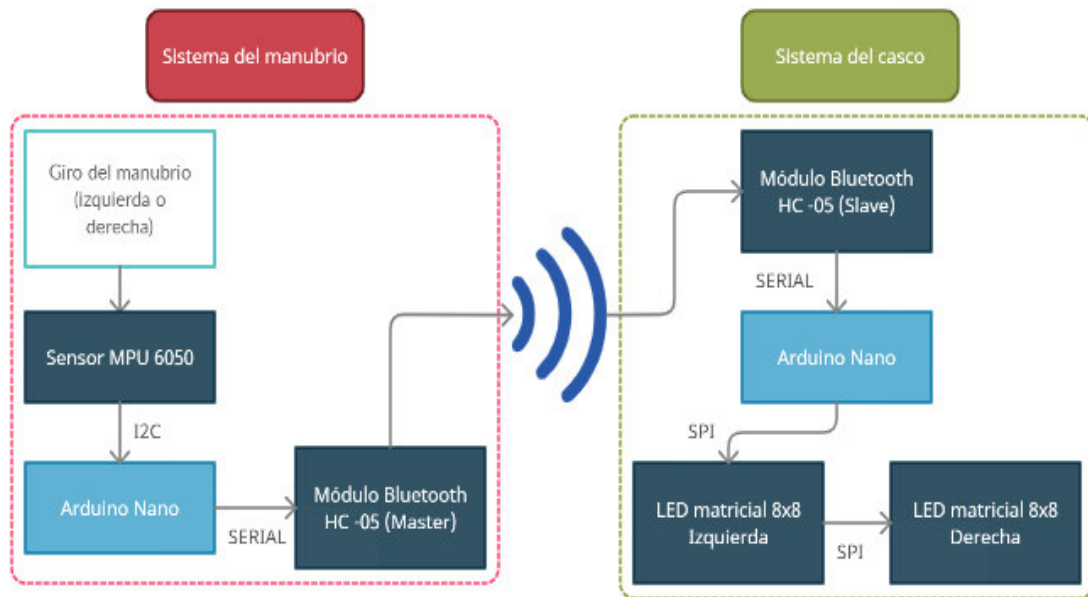


Figura 3.6 Diagrama de funcionamiento del sistema de direccionales

Diseño de placas PCB

Con respecto al diseño de las placas PCB en la que se implementarán los dispositivos, se realizó en el programa Proteus ya que presenta una interfaz amigable con quien quiera crear un circuito electrónico y una placa PCB, esto debido a que cuenta con las librerías de los componentes utilizados en este proyecto.

En la Tabla 3.4 y en la Tabla 3.5 se muestran las conexiones del módulo maestro y del módulo esclavo respectivamente.

Tabla 3.4 Conexiones del módulo maestro

Módulo	Pines del módulo	Pines del Arduino	Pines de la batería LiPo
HC - 05	RX	D10	-
	TX	D11	-
	VCC	5V	-
	GND	GND	-

Módulo	Pines del módulo	Pines del Arduino	Pines de la batería LiPo
MPU6050	SCL	A5	-
	SDA	A4	-
	VCC	5V	-
	GND	GND	-
TP - 4050	OUT+	5V	-
	OUT-	GND	-
	B+	-	Positivo
	B-	-	Negativo
	J1	-	-
	J2	-	-

Tabla 3.5 Conexiones del módulo esclavo

Módulo	Pines del módulo	Pines del Arduino	Pines del MAX7219 (Matriz1)
HC - 05	RX	D9	-
	TX	D8	-
	VCC	5V	-
	GND	GND	-
MAX7219 (Matriz)	DIN	D11	-
	CS	D10	-
	CLK	D13	-
	VCC	5V	-
	GND	GND	-
MAX7219 (Matriz2)	DIN	-	DIN
	CS	-	CS
	CLK	-	CLK
	VCC	-	VCC
	GND	-	GND

En la Figura 3.7 y Figura 3.8 se muestra el diagrama esquemático de conexión para cada placa Arduino.

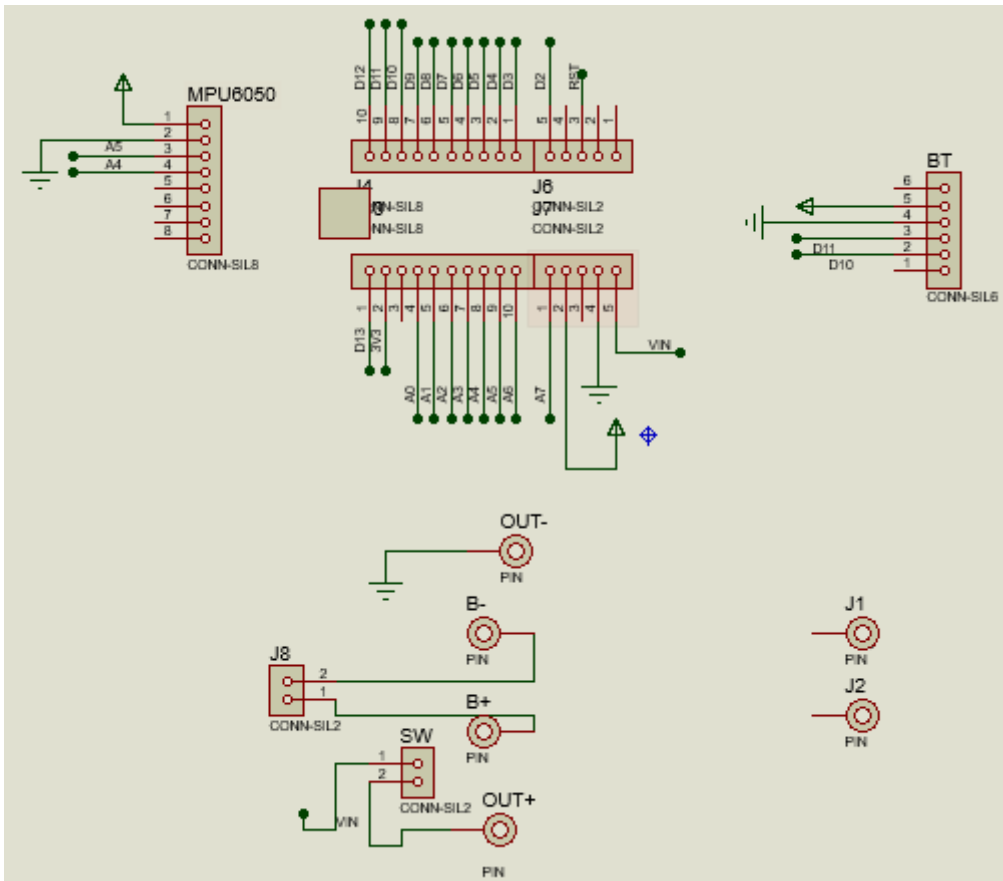


Figura 3.7 Diagrama esquemático del módulo maestro

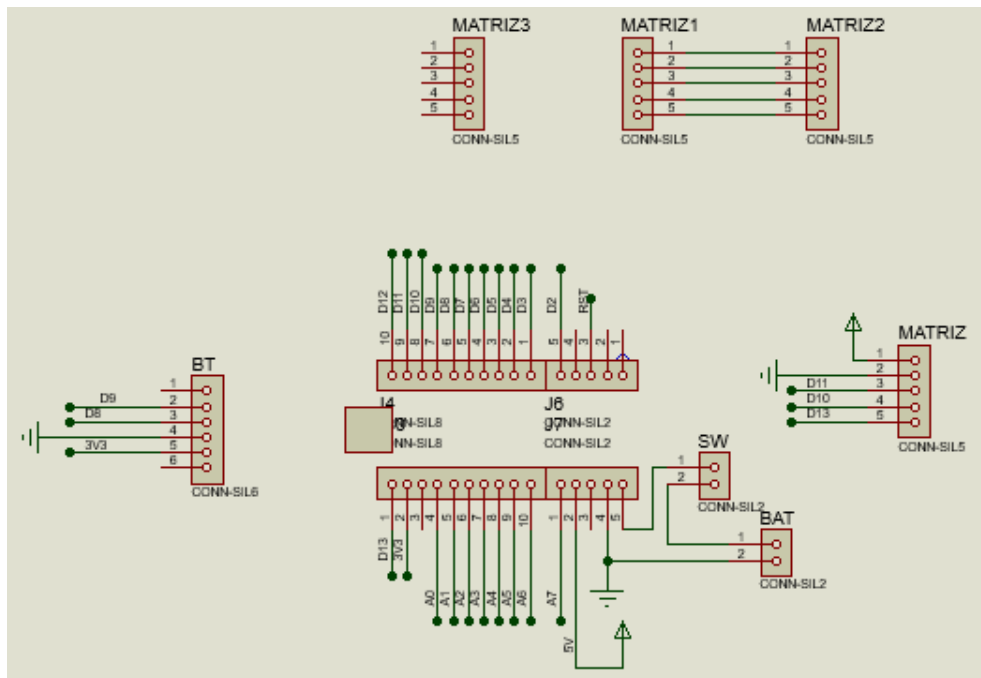


Figura 3.8 Diagrama esquemático del módulo esclavo

Para el diseño de las placas PCB, se tomaron en cuenta ciertas consideraciones, tales como: seleccionar el tamaño adecuado de las pistas, ya que el ancho debe ser suficiente para permitir el paso de la corriente máxima que circule por la pista. En el caso de ambas placas, tanto para el maestro, como para el esclavo, el ancho de la pista es de 0.2 (mm) [32]. Por otra parte, para voltajes de entre 5 (V) y 10 (V) la separación entre pistas es de 0.3 (mm) [32]. Además, cuando existan curvas de las pistas, éstas no deben ser de ángulos de 90° [32].

La Figura 3.9 y Figura 3.10 muestran el diseño de la placa PCB en Proteus.

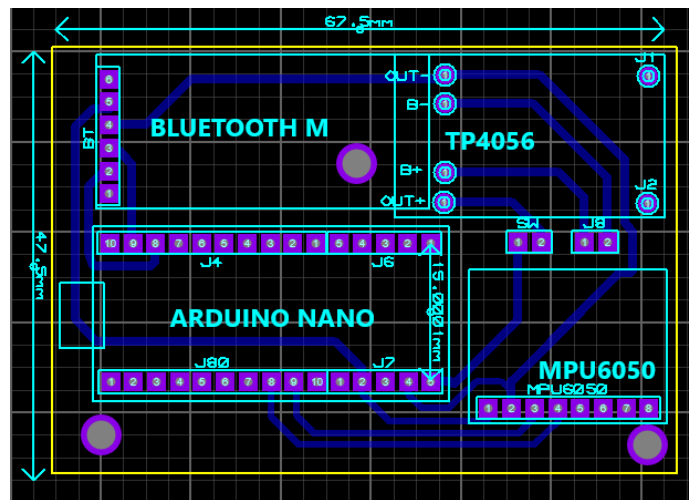


Figura 3.9 Diseño de la placa PCB del módulo maestro

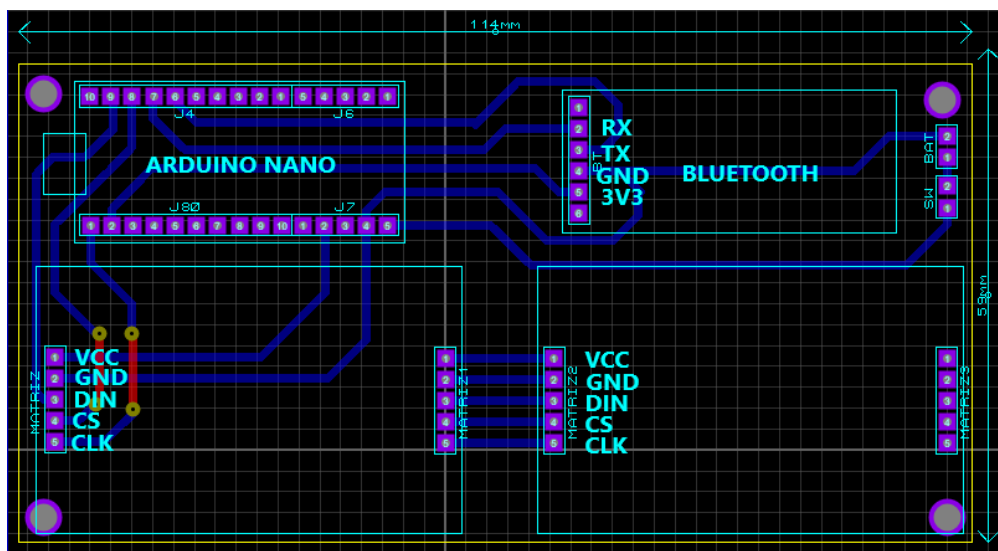


Figura 3.10 Diseño de la placa PCB del módulo esclavo

Una vez realizadas las placas, se procede a soldar los componentes respectivos en cada placa PCB. En la Figura 3.11 se presenta la placa PCB soldada con los componentes del módulo maestro.

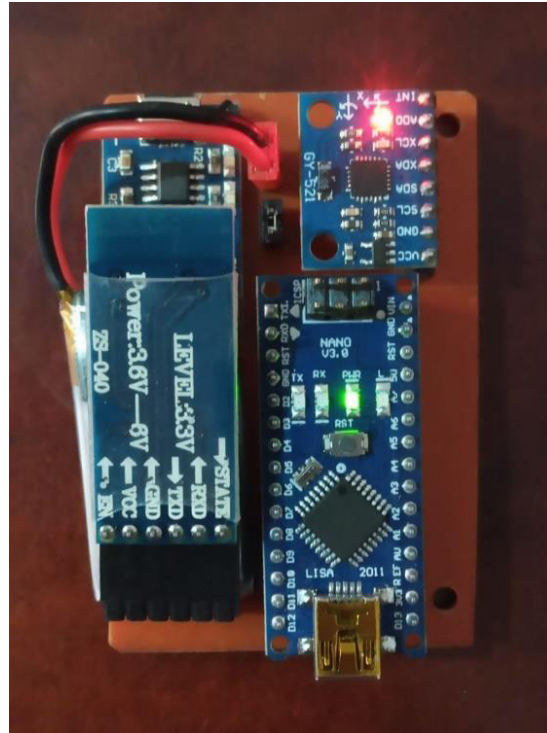


Figura 3.11 Placa del módulo maestro

En la Figura 3.12 se muestra la placa PCB del módulo esclavo con sus componentes soldados.

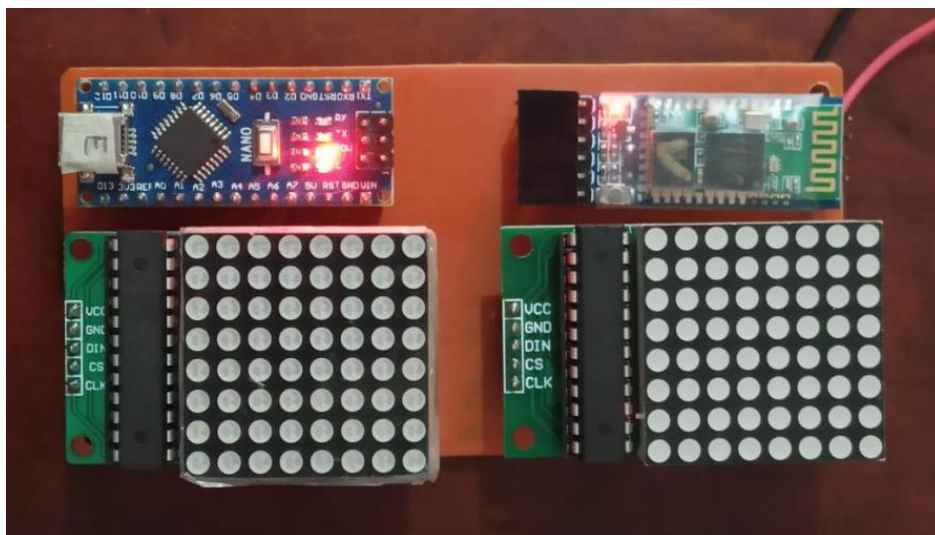


Figura 3.12 Placa del módulo esclavo

Dimensionamiento de las fuentes de alimentación

En la Tabla 3.6 y en la Tabla 3.7 se detallan los valores de corriente del circuito del módulo maestro y del circuito del módulo esclavo respectivamente.

Tabla 3.6 Consumo de corriente del circuito del módulo maestro

Elemento	Corriente (mA)
Arduino Nano	15
MPU6050	3.8
HC-05	50
Total	68.8

Tabla 3.7 Consumo de corriente del circuito del módulo esclavo

Elemento	Corriente (mA)
Arduino Nano	15
LED matricial 1	330
LED matricial 2	330
HC-05	50
Total	725

Como se pudo observar, el consumo de corriente es distinto para cada uno, por esta razón es necesario el uso de diferentes baterías para cada módulo, de esta manera, las baterías deben satisfacer la necesidad de voltaje y corriente sin que se sobre dimensionen pues esto implica baterías más grandes y por ende más peso, haciendo que no se cumpla con el requisito de ser pequeño y ligero para el uso del sistema de direccionales.

Así pues, ambos módulos cuentan con baterías recargables, en el caso del módulo maestro, cuenta con una batería tipo LiPo de 3.7 (V) y con 350 (mAh) de corriente, suficiente para alimentar todo el circuito del módulo maestro, cuenta con un módulo TP4050 que permite cargar a la batería LiPo. La batería tiene una duración de aproximadamente 5 horas sin interrupción. Esta batería al ser pequeña también cumple con el objetivo de hacerle compacto al diseño del módulo que se coloca en el manubrio de la bicicleta.

En cuanto a la batería del módulo esclavo, se usa una batería tipo *Power Bank* de 5 (V) y con 3000 (mAh), en este caso, se utiliza una batería de más capacidad para poder encender los LED matriciales. La batería tiene una duración de aproximadamente 4 horas de uso sin interrupción. De igual manera, esta batería es ligera y compacta, que se ajusta al tamaño de la placa.

Diseño del sistema de direccionales

En cuanto al diseño del sistema de direccionales, éste debe ser lo más simple posible y de un material resistente pero que su peso no sea tan elevado, de esta manera se garantiza que el sistema de direccionales sea cómodo para el usuario.

El diseño se lo realizó en el programa *SolidWorks* en el cual se pueden crear diseños en 3D, este programa presenta grandes ventajas con respecto a otros programas para diseño en 3D pues tiene una interfaz altamente intuitiva, por lo que el usuario puede crear, editar, fabricar y simular su diseño en 3D [33]. Por otra parte, el programa incorpora tutoriales y reajustes para que el diseño sea rápido y sin complicaciones [33]. El programa permite que los diseños puedan ser impresos en una impresora 3D y facilitar la obtención del diseño.

En la Figura 3.13 y Figura 3.14 se presentan los diseños realizados en el programa *SolidWorks*.

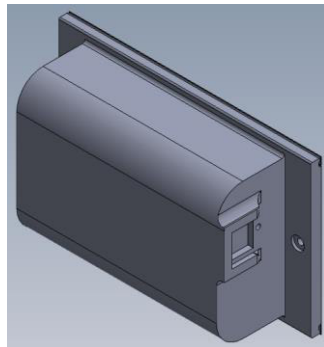


Figura 3.13 Diseño de la caja del módulo maestro

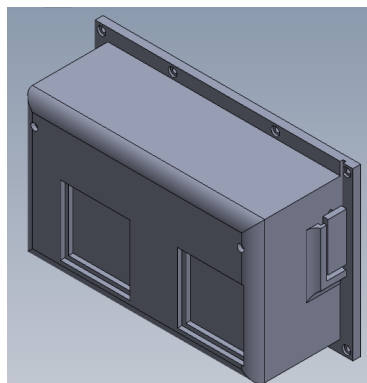


Figura 3.14 Diseño de la caja del módulo esclavo

El tamaño de la caja del módulo maestro que se coloca en el manubrio de la bicicleta es de 9.6 (cm) de largo por 6.6 (cm) de ancho por 3.8 (cm) de alto, haciéndola muy compacta para ubicarla en cualquier manubrio de bicicleta, además, tiene dos ranuras

para colocar correas o amarras plásticas para su sujeción en el manubrio. En la Figura 3.15 se muestra la caja del módulo maestro.



Figura 3.15 Caja del módulo maestro

En la Tabla 3.8 Comparación del peso de la caja del módulo maestro se compara el peso de la caja del módulo maestro sin el circuito y con el circuito.

Tabla 3.8 Comparación del peso de la caja del módulo maestro

	Caja sin circuito	Caja con circuito
Peso (kg)	0.070	0.100

Como se puede observar en la Tabla 3.8 la caja es liviana, cumpliendo así el requerimiento de hacer que el módulo maestro no sea pesado y que no estorbe al ciclista cuando éste esté manejando.

Por otra parte, el tamaño de la caja del módulo esclavo que se coloca en el casco es de 13.3 (cm) de largo por 8.6 (cm) de ancho por 4.7 (cm) de alto, el diseño de esta caja se procuró que sea lo más pequeña posible y que no sea muy pesada, pues si así lo fuese incomodaría al usuario. En la Figura 3.16 se observa la caja del módulo esclavo.



Figura 3.16 Caja del módulo esclavo

En la Tabla 3.9 Comparación del peso de la caja del módulo esclavo se compara el peso de la caja del módulo esclavo, sin y con el circuito.

Tabla 3.9 Comparación del peso de la caja del módulo esclavo

	Caja sin circuito	Caja con circuito
Peso (kg)	0.070	0.220

De acuerdo con la Tabla 3.9 se puede apreciar que el módulo esclavo es ligero para ubicarlo en el casco del usuario, por lo tanto, el diseño de la caja cumple con los requerimientos planteados para ser utilizado en el casco, ya que es compacto y ligero, permitiendo así que el usuario use cómodamente el sistema de direccionales en su casco.

En la Figura 3.17 y en la Figura 3.18 se presenta la colocación de la caja del módulo maestro en el manubrio de la bicicleta. Contiene un soporte para poderla ubicar en cualquier parte del manubrio; sin embargo, la colocación para este proyecto es de manera vertical.



Figura 3.17 Caja del módulo maestro en el manubrio



Figura 3.18 Caja del módulo maestro en el manubrio

En la Figura 3.19 se muestra la caja del módulo esclavo colocada en el casco. Para la sujeción al casco se utilizan tornillos, de esta manera se asegura que las direccionales estén sujetas al casco.



Figura 3.19 Caja del módulo esclavo en el casco

En la Figura 3.20 se muestra un diagrama general del funcionamiento del sistema de direccionales.

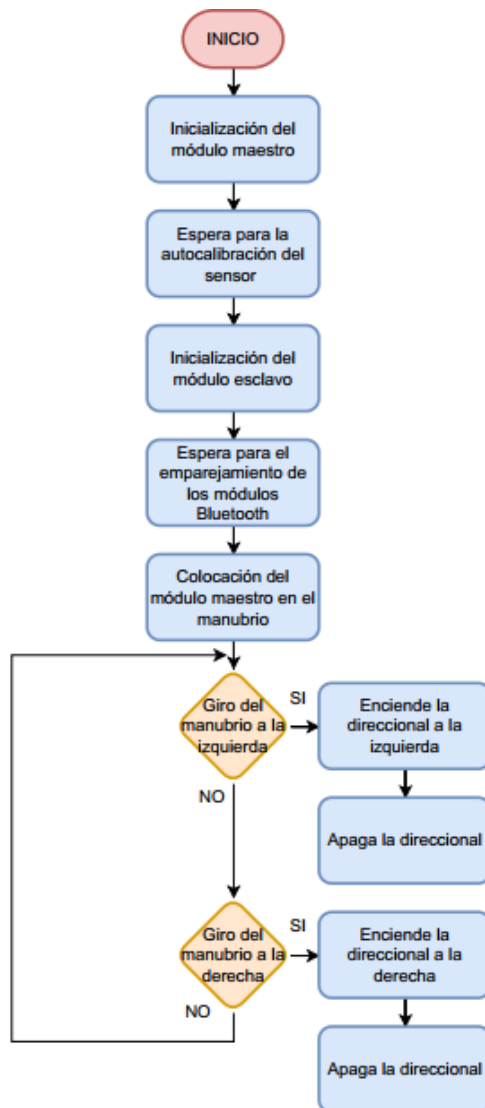


Figura 3.20 Diagrama de flujo del funcionamiento del sistema de direccionales

3.4 Configurar los dispositivos

La programación de los dispositivos, se lo realizó en el IDE de Arduino, un IDE es un entorno de programación que consiste en un editor de código, compilador, depurador y constructor de interfaz de usuario gráfica (GUI) [34]. La programación en el IDE de Arduino está basada en un lenguaje propio llamado *Wiring*, que está escrito en los lenguajes de programación C y C++, por esta razón es que la sintaxis es parecida a estos lenguajes [35]. Este programa se usa para cargar la programación en los microcontroladores Arduino mediante un *bootloader* que permite subir dicha programación al microcontrolador sin usar *hardware* adicional [34]. Además, el IDE de Arduino contiene librerías incluidas dentro del programa para el mejor control del microcontrolador, así como librerías que se pueden obtener de páginas de descarga como GitHub, las cuales pueden ser de mucha utilidad al momento de controlar algún módulo de Arduino.

Para los módulos *Bluetooth*, se empleó un programa básico para la transmisión y recepción de datos con la que se puede establecer la comunicación entre módulos con comandos AT en el monitor serial. Los comandos AT o comandos AT de Hayes sirven para establecer una comunicación de envío y recepción de datos para los comandos de control. El nombre AT hace referencia a la palabra atención, además de que cada comando debe empezar con AT [36]. En la Figura 3.21 se presenta el diagrama de flujo de la programación de los módulos *Bluetooth*.

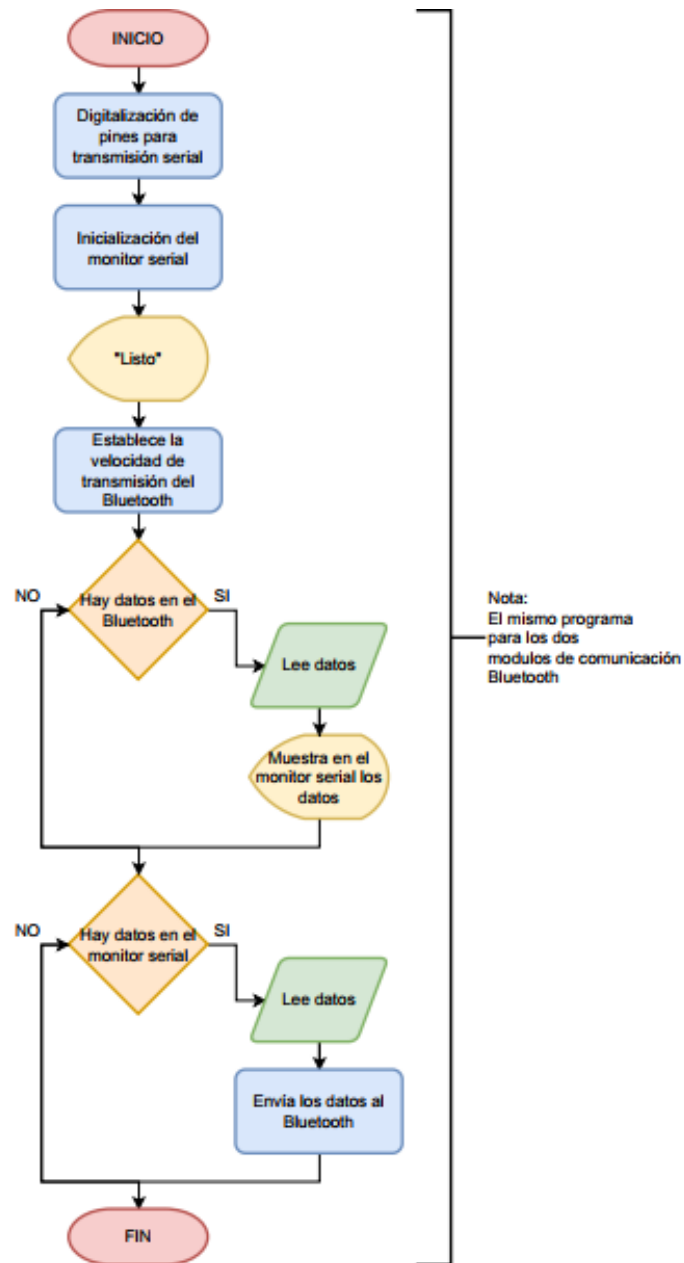


Figura 3.21 Diagrama de flujo de la programación del módulo *Bluetooth*

En la Tabla 3.10 se presentan todos los comandos utilizados en este proyecto para el emparejamiento de los módulos *Bluetooth* HC - 05.

Tabla 3.10 Comandos AT [37]

Comando	Respuesta	Parámetro	Función
AT	OK	Ninguno	Test para saber si el módulo <i>Bluetooth</i> recibe los comandos

Comando	Respuesta	Parámetro	Función
AT+ORGL	OK	Ninguno	Restaura el módulo <i>Bluetooth</i> al estado predeterminado
AT+ADDR?	+ADDR:<Param> OK	Dirección del módulo <i>Bluetooth</i>	Muestra la dirección del módulo <i>Bluetooth</i>
AT+NAME?	+NAME:<Param> OK	Nombre del módulo <i>Bluetooth</i> Por defecto: HC-05	Muestra el nombre del módulo <i>Bluetooth</i>
AT+NAME=<Param>	OK	Nombre del módulo <i>Bluetooth</i>	Cambia el nombre del módulo <i>Bluetooth</i>
AT+ROLE?	+ROLE:<Param> OK	0 → Rol esclavo 1 → Rol maestro 2 → Rol de esclavo enlazado Por defecto: 0	Muestra el rol en el que se encuentra el módulo <i>Bluetooth</i>
AT+ROLE=<Param>	OK	0 → Rol esclavo 1 → Rol maestro 2 → Rol de esclavo enlazado	Cambia el rol del módulo <i>Bluetooth</i>
AT+UART?	+UART=<Param1>,<Param2>,<Param3> OK	Parámetro1: <i>baud rate</i> (bits/s). El valor puede ser uno de los siguientes: 4800 9600 19200 38400 57600 115200	Muestra la velocidad del puerto serial en baudios, el bit de parada y el bit de paridad del módulo <i>Bluetooth</i>

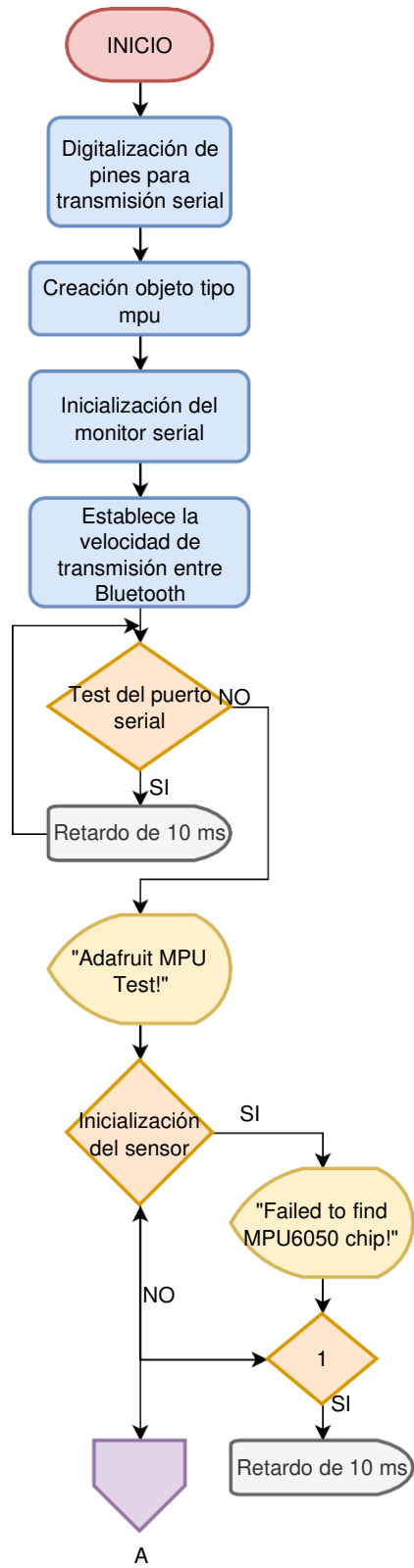
Comando	Respuesta	Parámetro	Función
		230400 460800 921600 1382400 Parámetro2: bit de parada: 0 → 1 bit 1 → 2 bits Parámetro3: bit de paridad: 0 → Ninguno 1 → Paridad impar 2 → Paridad par Por defecto: 9600, 0, 0	
AT+UART=<Param1>,<Param2>,<Param3>	OK	Parámetro1: <i>baud rate</i> (bits/s). El valor puede ser uno de los siguientes: 4800 9600 19200 38400 57600 115200 230400 460800 921600 1382400 Parámetro2: bit de parada: 0 → 1 bit 1 → 2 bits	Cambia la velocidad del puerto serial en baudios, el bit de parada y el bit de paridad del módulo <i>Bluetooth</i>

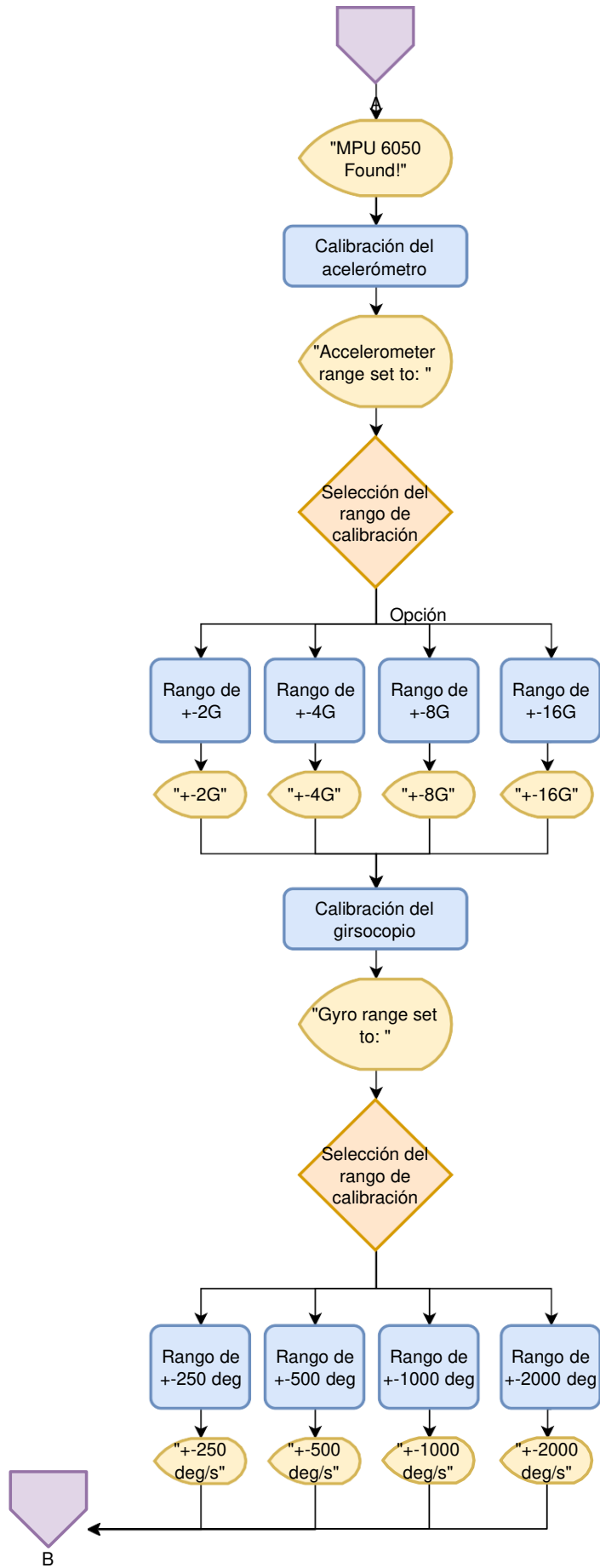
Comando	Respuesta	Parámetro	Función
		Parámetro3: bit de paridad: 0 → Ninguno 1 → Paridad impar 2 → Paridad par	
AT+CMODE?	+CMODE:<Param> OK	0 → Conecta el módulo <i>Bluetooth</i> a una dirección específica de <i>Bluetooth</i> (La dirección <i>Bluetooth</i> tiene que ser especificada en el comando BIND) 1 → Conecta el módulo <i>Bluetooth</i> a cualquier dirección 2 → Conecta esclavos en cadena Modo de conexión por defecto: 0	Muestra el modo de conexión del módulo <i>Bluetooth</i>
AT+CMODE:<Param>	OK	0 → Conecta el módulo <i>Bluetooth</i> a una dirección específica de <i>Bluetooth</i> (La dirección <i>Bluetooth</i> tiene que ser	Cambia el modo de conexión del módulo <i>Bluetooth</i>

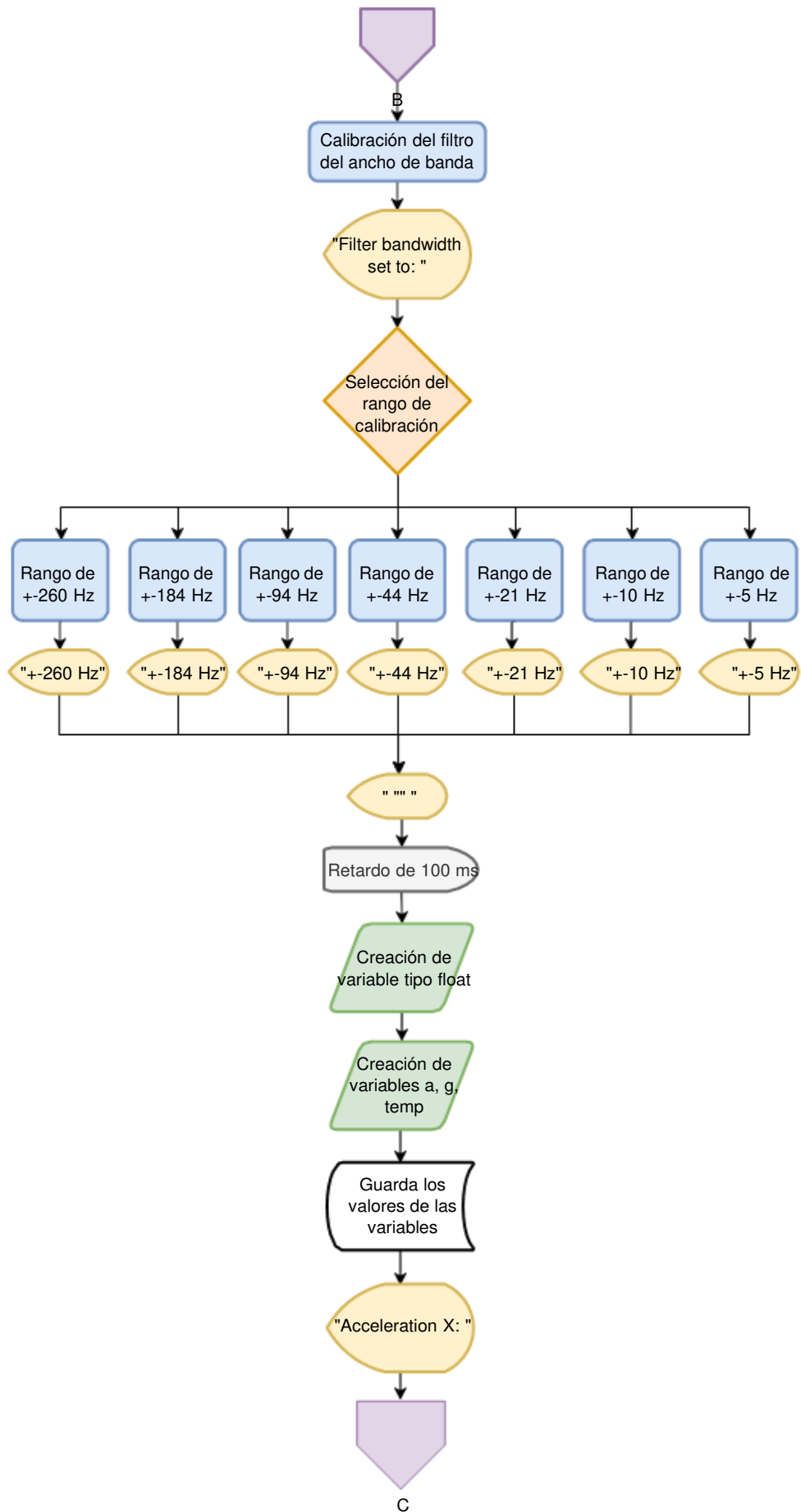
Comando	Respuesta	Parámetro	Función
		especificada en el comando BIND) 1 → Conecta el módulo <i>Bluetooth</i> a cualquier dirección 2 → Conecta esclavos en cadena	
AT+BIND?	+BIND:<Param> OK	Dirección del <i>Bluetooth</i> emparejada Dirección por defecto: 00:00:00:00:00:00	Muestra la dirección de <i>Bluetooth</i> emparejada
AT+BIND:<Param>	OK	Dirección de <i>Bluetooth</i> para emparejar	Ingresa la dirección de <i>Bluetooth</i> para emparejar

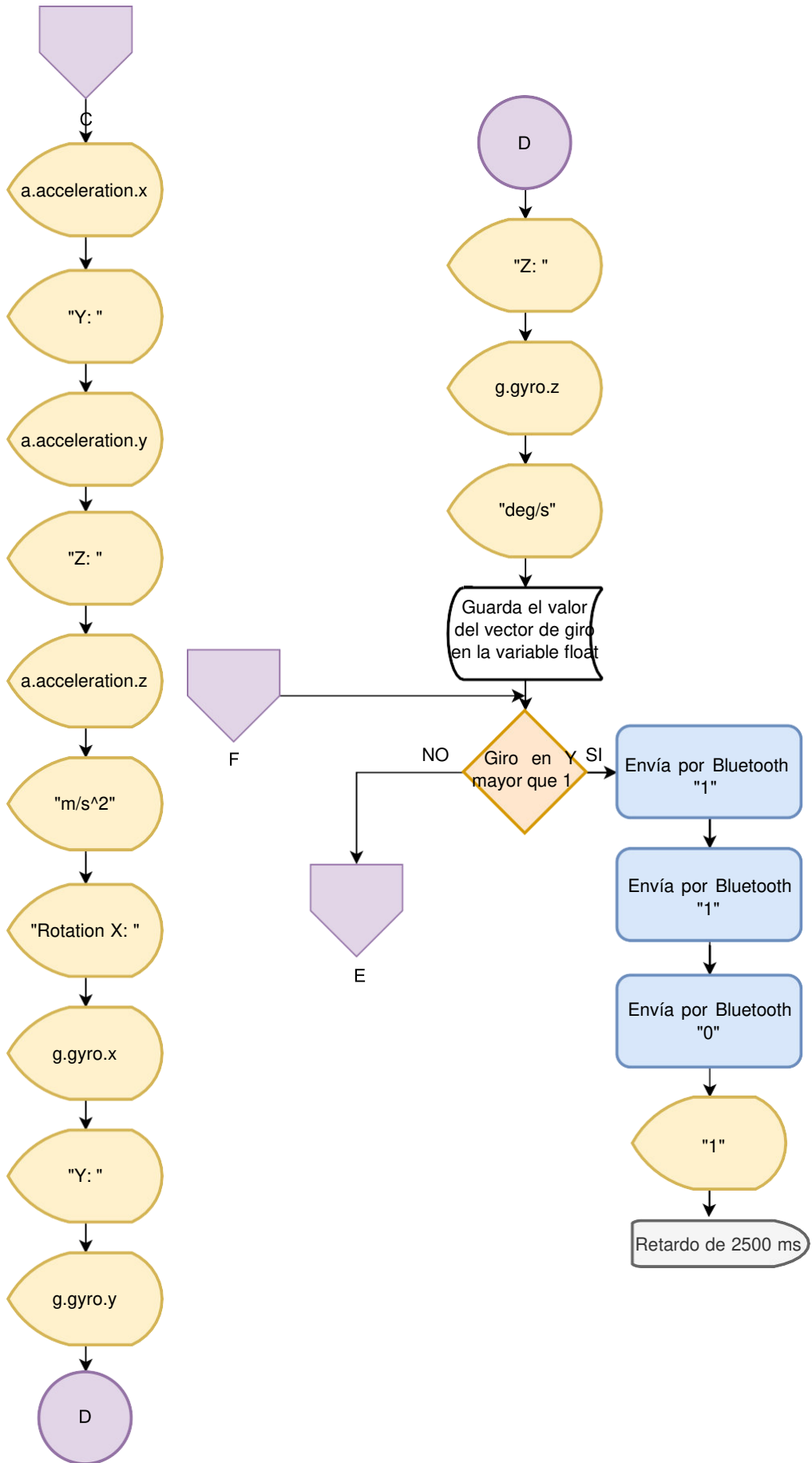
Una vez vinculados los dispositivos *Bluetooth* maestro y esclavo, dentro de la programación para el microcontrolador maestro y esclavo es necesario utilizar la librería *SoftwareSerial.h*, la cual hace que los pines digitales del microcontrolador permitan la comunicación serial, y de esta manera facilita el uso de otros pines.

Para la programación del módulo maestro con el sensor MPU6050 es necesario utilizar la librería *wire.h* para la comunicación serial I2C entre el sensor y el microcontrolador Arduino. Adicional a esta librería es necesario el uso de la librería *adafruit_MPU6050* para el control del sensor MPU6050 [38]. En la Figura 3.22 se muestra el diagrama de flujo de la programación del módulo maestro.









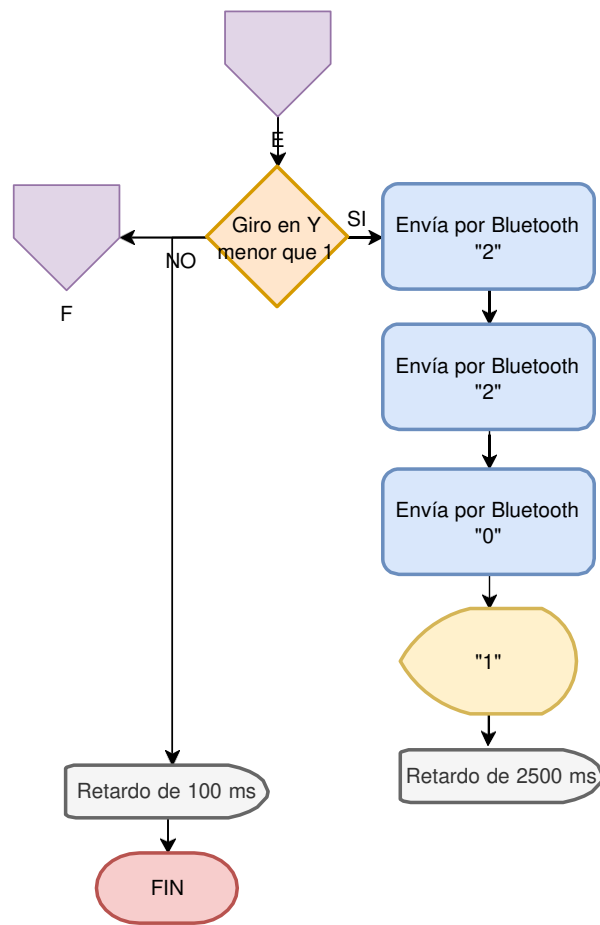
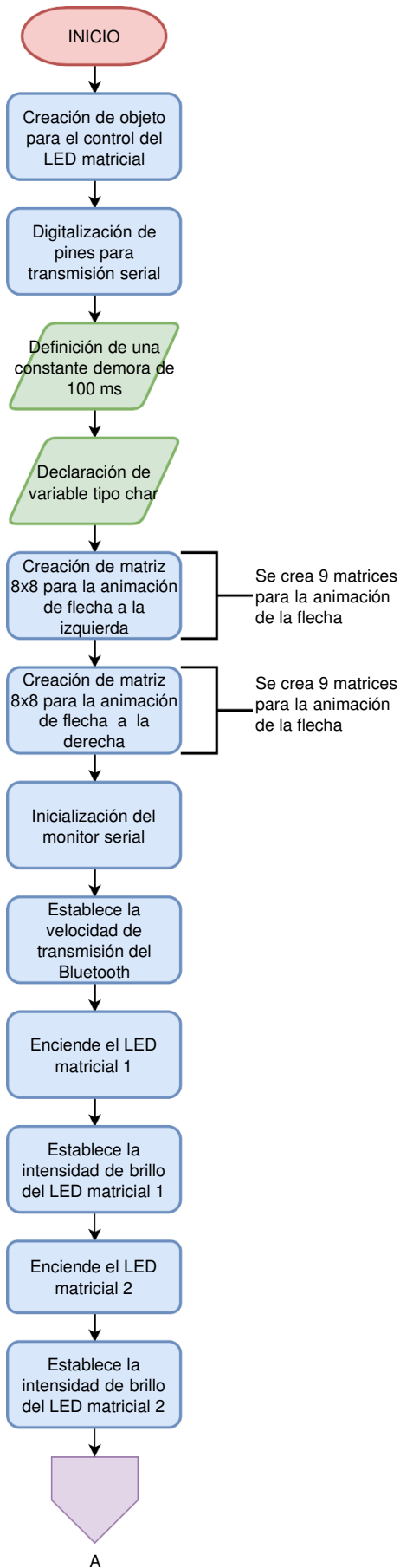
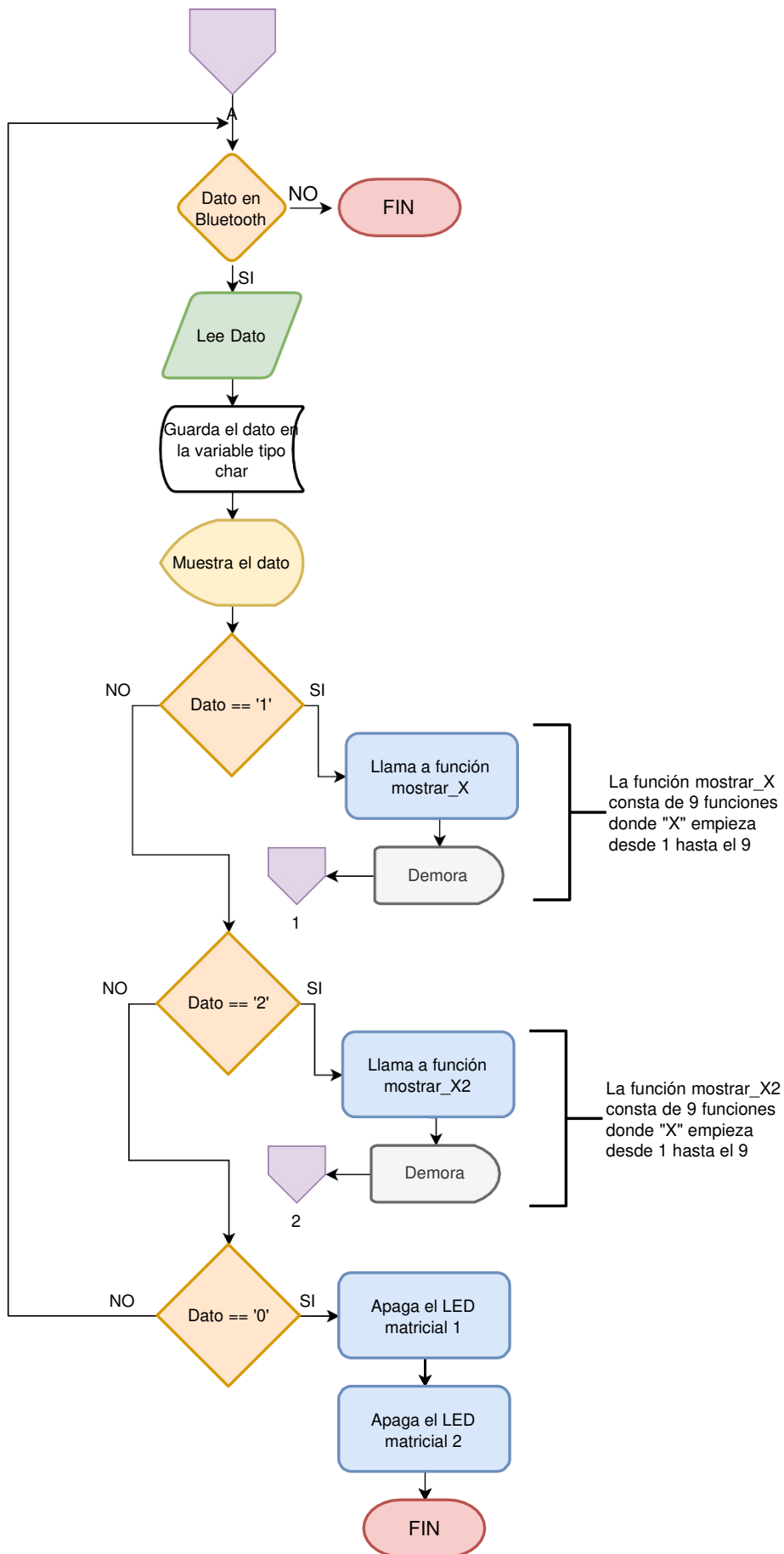


Figura 3.22 Diagrama de flujo de la programación del módulo maestro

Ahora bien, en el módulo esclavo se realizó la programación de los LED matriciales, se utilizó la librería *LedControl.h* que permite el control del circuito integrado MAX7219 y con éste el control de cada LED matricial. La librería permite apagar y prender todos los LED de la matriz, controlar todos los dispositivos conectados en cascada, también permite el control de la intensidad luminosa de los LED y permite el control individual de cada LED mediante la creación de matrices, de este modo se pueden crear imágenes en el LED matricial [39]. En la Figura 3.23 se presenta el diagrama de flujo de la programación del módulo esclavo.





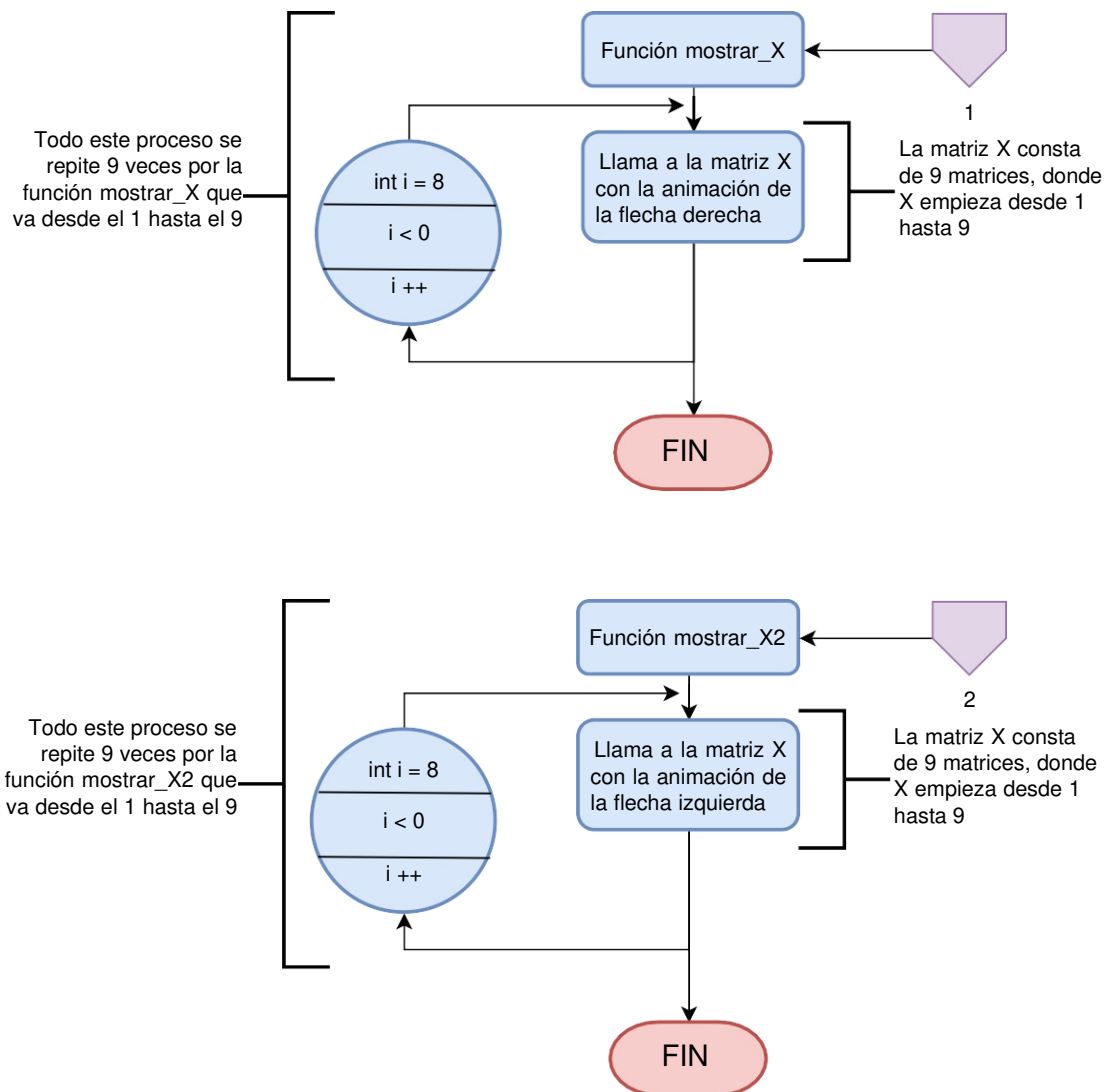


Figura 3.23 Diagrama de flujo de la programación del módulo esclavo

3.5 Pruebas y análisis de resultados

La primera prueba se realizó a los módulos *Bluetooth* HC - 05 para comprobar los estados de conexión y envío correcto de datos, a lo cual se realizó un programa base para el envío y recepción de datos que permite visualizar dichos datos de envío en el monitor serial del IDE de Arduino. El envío de datos se lo hizo desde un dispositivo celular Android mediante una aplicación para conexión de dispositivos *Bluetooth* con Arduino.

En la Figura 3.24 se muestra el monitor serial del IDE de Arduino con los caracteres enviados desde el celular hasta el módulo *Bluetooth* HC-05, demostrando que el módulo recepta con normalidad los datos.

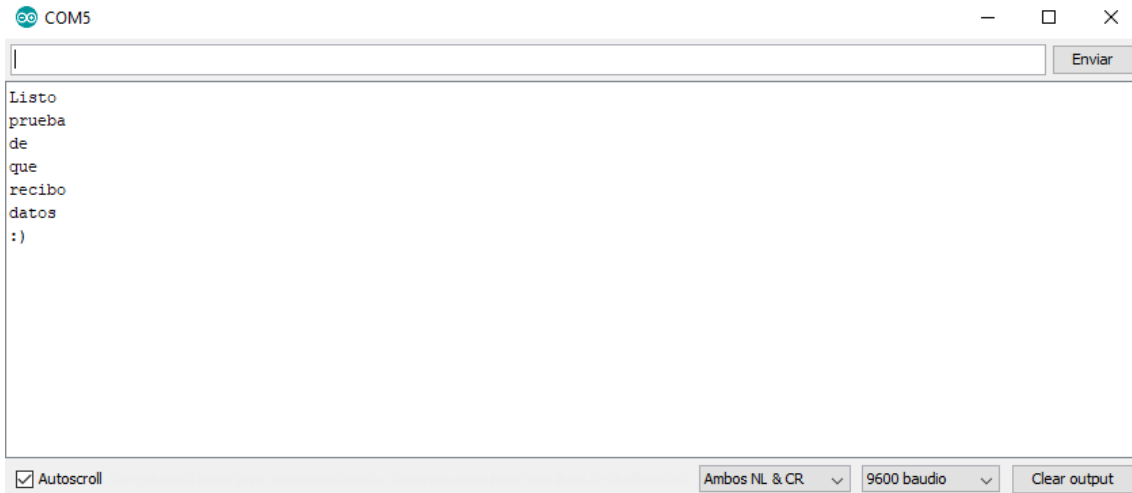


Figura 3.24 Monitor serial de la prueba del HC-05

En segundo lugar, se realizó la prueba de la calibración del sensor MPU6050, de modo que se puede adaptar a cualquier sentido de giro dependiendo del lugar que se coloque el módulo en el manubrio. En la Figura 3.25 se muestra el monitor serial con los datos obtenidos de la medición del MPU6050.

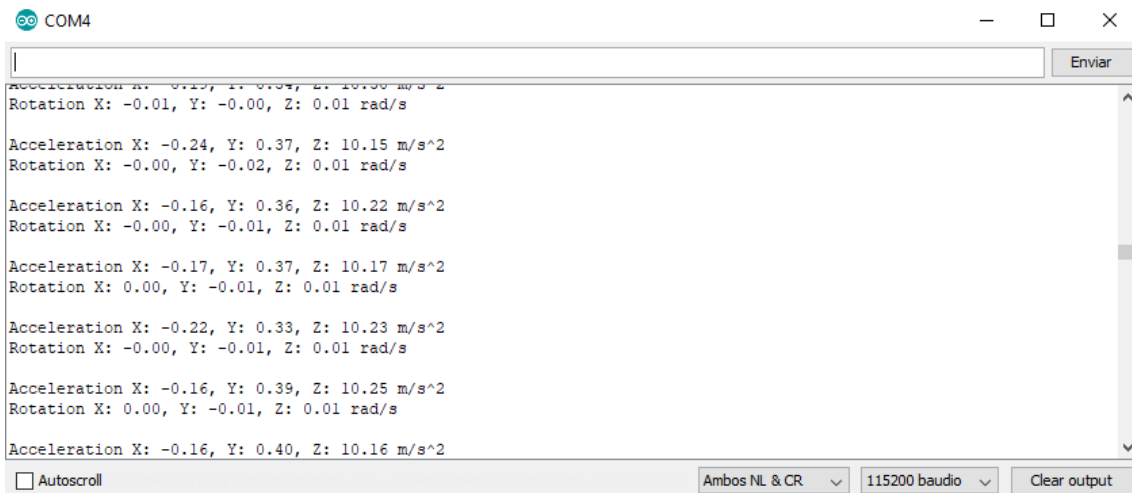


Figura 3.25 Monitor serial de los datos obtenidos del MPU6050

Como se puede observar, el sensor mide correctamente la aceleración y el giro en los 3 distintos ejes X, Y y Z. En este proyecto se tomará en cuenta la aceleración en el eje X cuando el manubrio gire hacia la derecha o a la izquierda.

Se realizó una modificación al programa principal del módulo maestro para que, cuando el sensor detecte una aceleración a partir de 1 (m/s^2) muestre el número <2>, indicando que se realizó el sentido de giro hacia la derecha. Por otro lado, cuando el sensor detecte una aceleración a partir de -1 (m/s^2) muestre el número <1>, indicando que se realizó el

sentido de giro hacia la izquierda. Además, se empleó un retardo de 100 (ms) para cuando el manubrio regresa a su posición inicial. En la Figura 3.26 se muestra el monitor serial con los datos obtenidos del giro del MPU6050.

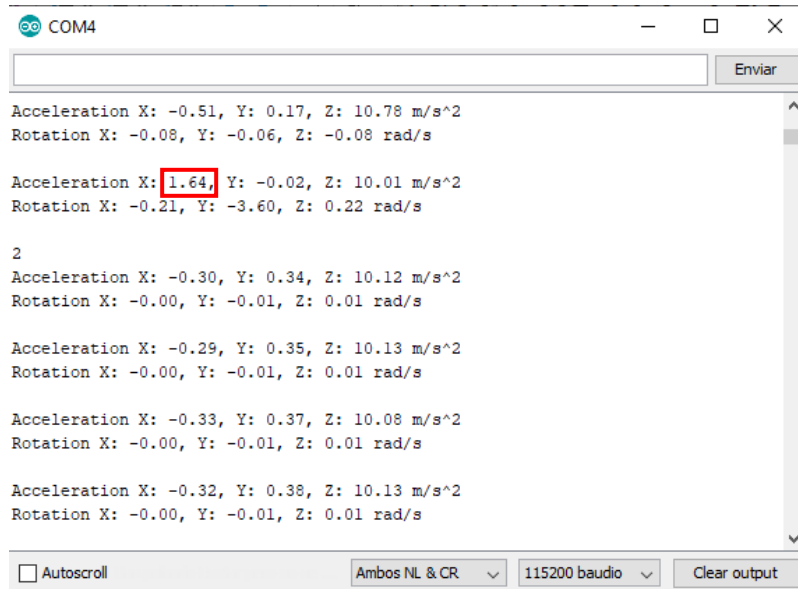


Figura 3.26 Monitor serial de la prueba de giro hacia la derecha

Tal como se observa en la Figura 3.26, cuando se gira el sensor hacia la derecha y mide una aceleración en el eje X mayor a 1 (m/s^2), en este caso 1,64 (m/s^2) muestra en el monitor serial el número 2, indicando que el sensor se movió hacia la derecha. Por otra parte, en la Figura 3.27 se observan los datos obtenidos al girar el sensor hacia la izquierda.

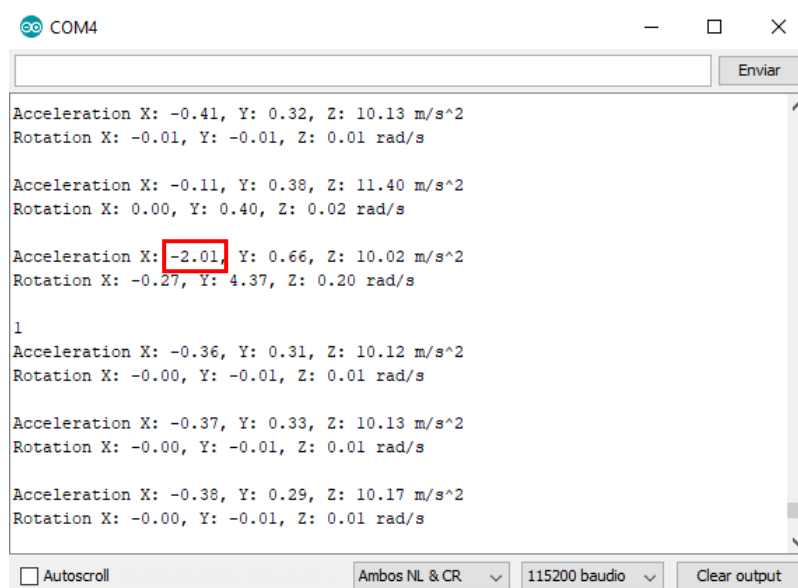


Figura 3.27 Monitor serial de la prueba de giro hacia la izquierda

En la Figura 3.27 se observa cuando el sensor es girado hacia la izquierda y detecta una aceleración de $-1 \text{ (m/s}^2\text{)}$, en este caso de $-2,01 \text{ (m/s}^2\text{)}$, mostrando en el monitor serial el número 1, de modo que la prueba de funcionamiento en la detección de la aceleración en los distintos sentidos de giro es satisfactoria.

Finalmente, en la Figura 3.28 y en la Figura 3.29 se observa que, con los datos obtenidos del sensor MPU6050, los números 1 y 2 son enviados por el módulo *Bluetooth* según su sentido de giro, de tal manera que, cuando se gire hacia la izquierda, se envíe el número 1 y se encienda la direccional izquierda y, cuando se gire hacia la derecha, se envíe el número 2 y se encienda la direccional hacia la derecha. Además, en la Figura 3.30 se muestran la animación de la direccional cuando el manubrio se gira hacia la derecha, y en la Figura 3.31 se muestra la animación de la direccional cuando el manubrio se gira hacia la izquierda.

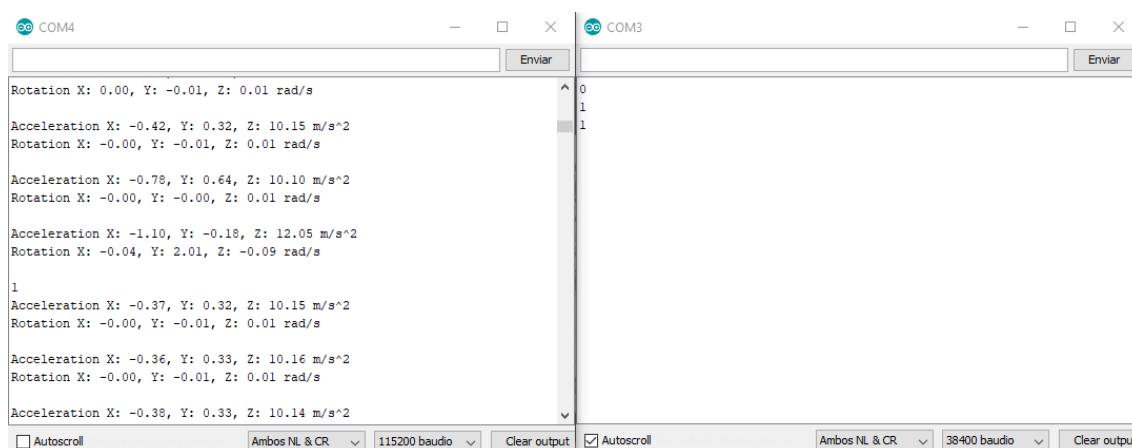


Figura 3.28 Monitor serial del módulo maestro y esclavo al girar hacia la izquierda

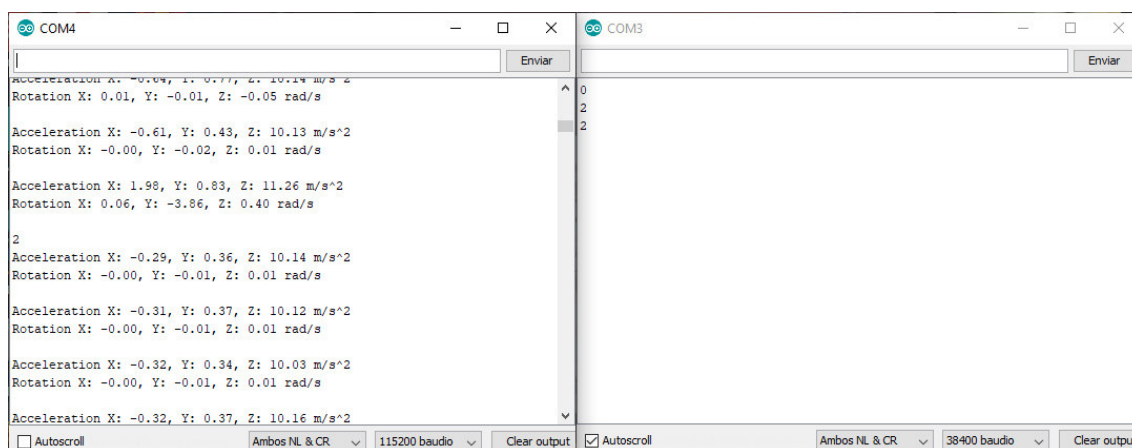


Figura 3.29 Monitor serial del módulo maestro y esclavo al girar a la derecha



Figura 3.30 Direccional hacia la derecha



Figura 3.31 Direccional hacia la izquierda

3.6 Manual de Uso y Mantenimiento

Escanee el código QR de la Figura 3.32 para ver el video del manual de uso del sistema de direccionales para un casco de bicicleta.



Figura 3.32 Código QR del video del manual de uso

Escanee el código QR de la Figura 3.33 para ver el vídeo del manual de mantenimiento del sistema de direccionales para un casco de bicicleta.



Figura 3.33 Código QR del manual de mantenimiento

Presupuesto del proyecto

En la Tabla 3.11 se detalla el presupuesto del proyecto.

Tabla 3.11 Presupuesto del proyecto

Cantidad	Materiales	Costo por hora	Horas empleadas	Costo total
2	Arduino Nano	-	-	\$ 18,00
2	HC - 05	-	-	\$ 20,00
1	MPU 6050	-	-	\$ 6,50
1	Power Bank	-	-	\$ 5,50
1	Batería LiPo	-	-	\$ 7,00
1	TP4050	-	-	\$ 1,50
2	Matrix MAX7219	-	-	\$ 12,50

Cantidad	Materiales	Costo por hora	Horas empleadas	Costo total
1	Soporte para celular	-	-	\$ 12,00
1	Casco de bicicleta	-	-	\$ 25,00
2	Baquelitas	-	-	\$ 2,50
1	Cloruro férrico	-	-	\$ 2,50
1	Espadines	-	-	\$ 0,50
1	Estaño	-	-	\$ 3,00
15	Tornillos	-	-	\$ 0,25
10	Tuercas	-	-	\$ 0,50
4	Impresión en 3D	\$ 3,00	12	\$ 36,00
2	Fabricación de PCB	\$ 8,00	-	\$ 8,00
2	Diseño de PCB	\$ 5,00	5	\$ 25,00
3	Programación	\$ 10,00	20	\$ 200,00
4	Diseño en 3D	\$ 15,00	25	\$ 375,00
			Total	\$ 761,25

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Mediante este proyecto se puede concluir que el sistema de direccionales en el casco de bicicleta es muy útil para dar aviso del sentido de giro, pues lo que pretende es disminuir los accidentes ya que con las luces preventivas se anuncia de la presencia del ciclista. Además, la ausencia de cableado en el sistema de direccionales hace que el usuario se sienta cómodo con su uso.
- Mediante la identificación de los requerimientos necesarios y al realizar una comparación entre los distintos dispositivos a utilizarse, se pudieron escoger los elementos más adecuados para la realización de este proyecto, ya que cumplen con los requerimientos planteados con el fin de que el sistema de direccionales funcione correctamente.
- Las placas Arduino presentan una ventaja en cuanto a su programación, pues su IDE incorpora una gran variedad de librerías para sus distintos modelos de placas, además, los desarrolladores de los módulos desarrollan programaciones básicas para poder controlar dichos módulos, facilitando al programador la creación de nuevos programas.
- En cuanto al diseño del sistema de direccionales, la realización de las placas PCB cumplieron un papel fundamental al ahorrar espacio ya que en este proyecto se necesitaba limitar las dimensiones. Por otro lado, el diseño de las

cajas en 3D tanto como del manubrio como del casco, cumplieron los requisitos necesarios de ser ligeros y acoplarse a la necesidad del usuario.

- La configuración de los dispositivos mediante la programación de estos, facilitó la modificación y creación de diferentes programas, de modo que se pudieron utilizar sus características en específico como el acelerómetro en el sensor y así usar solo los datos de la aceleración para detectar el giro del manubrio. Por otra parte, los comandos AT ayudaron a la vinculación entre los módulos *Bluetooth* y de esta manera configurarlos como maestro – esclavo.
- Al realizar las pruebas de funcionamiento del sistema de direccionales por etapas, se obtiene un mayor control de cada sistema dentro de este proyecto, de esta manera se pueden detectar algunos fallos en específico dentro del sistema de direccionales, así pues, se pudo comprobar el correcto trabajo de los módulos *Bluetooth* HC-05 al enviar y recibir datos, así como del sensor acelerómetro giroscopio MPU6050 al realizar los giros y no tomar datos erróneos.

4.2 Recomendaciones

- Se puede usar el sensor acelerómetro del celular, en lugar de todo el módulo que se coloca en el manubrio. Desarrollando una aplicación telefónica que pueda tomar los datos del sensor y enviarlos por *Bluetooth* al módulo del casco y de este modo se ahorraría tanto en costo como en tiempo de empleo del proyecto.
- Se recomienda el uso de las librerías de los desarrolladores para que el sensor acelerómetro giroscopio funcione de la mejor manera, sin embargo, se debe evitar el uso de los retardos en la programación como el comando *delay()* ya que genera un retraso en la recepción de los datos, más bien, se deben usar interrupciones, de este modo el programa funcionaría de mejor manera.
- Se podría mejorar el diseño de la caja del módulo esclavo que se coloca en el casco a uno más vistoso.
- Se pueden colocar interruptores de encendido y apagado para facilitar el control de los módulos maestro y esclavo, de modo que se ahorraría el uso de la batería.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] INEC, «A Pedalear [Infografía],» 2017. [En línea]. Available: <https://www.ecuadorencifras.gob.ec/documentos/web-inec/Infografias-INEC/2017/170417.Bicicleta.pdf>. [Último acceso: 27 Junio 2021].
- [2] Oficina Legal de Fernando D. Vargas, «Nuevas estadísticas acerca de los accidentes de bicicleta fatales: las noticias no son buenas,» Blog, s/f. [En línea]. Available: <https://www.oficinalegaldeaccidentes.com/nuevas-estadisticas-acerca-de-los-accidentes-de-bicicleta-fatales-las-noticias-no-son-buenas/>. [Último acceso: 15 Enero 2021].
- [3] Agencia Nacional de Tránsito, *Reportes nacionales junio 2020*, Quito, 2020.
- [4] Arduino, «Arduino Nano,» Arduino Store, s/f. [En línea]. Available: <https://store.arduino.cc/usa/arduino-nano>. [Último acceso: 4 Julio 2021].
- [5] Seralo y L. B, «Raspberry PI Pico vs Arduino Nano vs BBC MicroBits vs Seeeduno Xiao,» SocialCompare, 3 Junio 2021. [En línea]. Available: <https://socialcompare.com/es/comparison/raspberry-pi-pico-vs-arduino-nano-vs-bbc-microbits>. [Último acceso: 4 Julio 2021].
- [6] Naylamp Mechatronics SAC, «TUTORIAL MPU6050, ACELERÓMETRO Y GIROSCOPIO,» Naylamp Mechatronics SAC, s/f. [En línea]. Available: https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html. [Último acceso: 17 Julio 2021].
- [7] D. C. Giancoli, «Efecto Coriolis,» de *FÍSICA. Principios con aplicaciones. Volumen 1*, Naucalpan de Juárez, Edo. de México, Pearson Educación de México, S.A. de C.V, 2006, pp. A-18.
- [8] Administrador, «I2C – Puerto, Introducción, trama y protocolo,» HeTPro, 28 Octubre 2017. [En línea]. Available: <https://hetpro-store.com/TUTORIALES/i2c/>. [Último acceso: 15 Enero 2021].
- [9] A. Pini, «Por qué y cómo usar la interfaz periférica serial para simplificar las conexiones entre distintos dispositivos,» Digi-Key Electronics, 14 Febrero 2019. [En línea]. Available: <https://www.digikey.com/es/articles/why-how-to-use-serial->

peripheral-interface-simplify-connections-between-multiple-devices. [Último acceso: 23 Agosto 2021].

- [10] InternetOfHomeThings, «Expanding Arduino Serial Port Buffer Size,» Internet of Home Things, 22 Septiembre 2015. [En línea]. Available: <https://internetofhomethings.com/homethings/?p=927>. [Último acceso: 3 Febrero 2022].
- [11] P. Alcalde San Miguel, «Diodos luminiscentes (LED),» de *Electrónica*, Madrid, Ediciones Paraninfo, SA, 2014, p. 108.
- [12] M. Martínez González, «Luz de carretera,» de *MF0624_1 - Técnicas básicas de electricidad de vehículos*, España, Elearning S.L., 2015, p. 180.
- [13] N. Goilav y G. Loi, «Arduino Nano,» de *Arduino. Aprender a desarrollar para crear objetos inteligentes*, Barcelona, Ediciones ENI, 2016, p. 227.
- [14] Proyecto Arduino, «Arduino UNO R3,» Proyecto Arduino, s/f. [En línea]. Available: <https://proyectoarduino.com/arduino-uno-r3/>. [Último acceso: 23 11 2021].
- [15] L. Llamas, «EL BUS SPI EN ARDUINO,» Luis Llamas, 16 Mayo 2016. [En línea]. Available: <https://www.luisllamas.es/arduino-spi/>. [Último acceso: 23 Noviembre 2021].
- [16] Ashwak, «Zigbee v/s Bluetooth,» ElectronicsHub, 23 Abril 2021. [En línea]. Available: <https://www.electronicshub.org/zigbee-vs-bluetooth/>. [Último acceso: 2 Febrero 2022].
- [17] Domodesk SL, «A FONDO: ZIGBEE,» Domodesk, s/f. [En línea]. Available: [https://www.domodesk.com/216-a-fondo-zigbee.html#:~:text=¿Qué%20características%20tiene%20la%20tecnología,y%200915%20MHz%20\(EEUU\)..](https://www.domodesk.com/216-a-fondo-zigbee.html#:~:text=¿Qué%20características%20tiene%20la%20tecnología,y%200915%20MHz%20(EEUU)..) [Último acceso: 3 Febrero 2022].
- [18] Ó. Torrente Artero, «¿Qué es Bluetooth?,» de *ARDUINO Curso práctico de formación*, México D.F., Alfaomega Grupo Editor, S.A. de C.V., 2013, p. 545.
- [19] Sony, «Perfiles Bluetooth con tecnología inalámbrica y descripciones,» Sony, 14 Abril 2021. [En línea]. Available: <https://www.sony->

latin.com/es/electronics/support/articles/00007501. [Último acceso: 23 Noviembre 2021].

- [20] Geek Factory, «HC-05 Módulo Bluetooth maestro esclavo,» Geek Factory, s/f. [En línea]. Available: <https://www.geekfactory.mx/tienda/modulos/radiofrecuencia/hc-05-modulo-bluetooth-maestro-esclavo/>. [Último acceso: 23 Noviembre 2021].
- [21] Geek Factory, «HC-06 Módulo Bluetooth esclavo,» Geek Factory, s/f. [En línea]. Available: <https://www.geekfactory.mx/tienda/modulos/radiofrecuencia/hc-06-modulo-bluetooth-esclavo/>. [Último acceso: 23 Noviembre 2021].
- [22] Naylamp Mechatronics SAC, «MÓDULO BLUETOOTH HC05,» Naylamp Mechatronics, s/f. [En línea]. Available: <https://naylampmechatronics.com/inalambrico/43-modulo-bluetooth-hc05.html>. [Último acceso: 25 Julio 2021].
- [23] Bricogeek, «MAX7219 - Controlador de LED 8 dígitos,» Bricogeek, s/f. [En línea]. Available: <https://tienda.bricogeek.com/componentes/168-max7219-controlador-de-led-8-digitos.html>. [Último acceso: 25 Julio 2021].
- [24] Maxim Integrated Products, Inc., «MAX7219/MAX7221 Serially Interfaced, 8-Digit LED Display Drivers,» 2021. [En línea]. Available: <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>. [Último acceso: 3 Febrero 2022].
- [25] Seabrookewindows, «Las ventajas y desventajas de la RAM estática,» seabrookewindows.com, 5 Marzo 2021. [En línea]. Available: <https://www.seabrookewindows.com/3xWz24jWd/>. [Último acceso: 4 Febrero 2022].
- [26] J. Guerra Carmenate, «Matriz LED con Arduino MAX7219 MAX7221,» PROGRAMAR FÁCIL, s/f. [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/matriz-led-arduino-max7219/>. [Último acceso: 25 Julio 2021].

- [27] AV Electronics, «Módulo Matriz LED 8x8,» AV Electronics, s/f. [En línea]. Available: <https://avelectronics.cc/producto/modulo-matriz-led/>. [Último acceso: 23 Noviembre 2021].
- [28] PCBoard.ca, «MAX7219 8x8 Dot Matrix Display With Red LED,» PCBoard.ca, s/f. [En línea]. Available: <https://www.pcboard.ca/max7219-display-driver-kit>. [Último acceso: 27 Julio 2021].
- [29] L. Llamas, «DETERMINAR LA ORIENTACIÓN CON ARDUINO Y EL IMU MPU-6050,» Luis Llamas, 22 Septiembre 2016. [En línea]. Available: <https://www.luisllamas.es/arduino-orientacion-imu-mpu-6050/>. [Último acceso: 26 Noviembre 2021].
- [30] L. Llamas, «USAR ARDUINO CON LOS IMU DE 9DOF MPU-9150 Y MPU-9250,» Luis Llamas, 26 Septiembre 2016. [En línea]. Available: <https://www.luisllamas.es/usar-arduino-con-los-imu-de-9dof-mpu-9150-y-mpu-9250/>. [Último acceso: 26 Noviembre 2021].
- [31] Ryspee, «Ryspee,» Ryspee, s/f. [En línea]. Available: <https://ryspee.com/tienda/acelerometro-y-giroscopio-mpu6050/>. [Último acceso: 27 Julio 2021].
- [32] M. Bellido, «Normas Básicas y Recomendaciones en el Diseño de,» Octubre 2015. [En línea]. Available: [file:///C:/Users/JOSU/Downloads/Tema5-NormasBasicasyRecomendacionesenDisenoPCB%20\(2\).pdf](file:///C:/Users/JOSU/Downloads/Tema5-NormasBasicasyRecomendacionesenDisenoPCB%20(2).pdf). [Último acceso: 15 Septiembre 2021].
- [33] Talentum, «¿Por qué elegir SOLIDWORKS?,» Talentum Digital 2021, 20 Enero 2020. [En línea]. Available: <https://talentumdigital.cl/2020/01/20/por-que-elegir-solidworks/>. [Último acceso: 15 Septiembre 2021].
- [34] Aprendiendo Arduino, «IDE Arduino,» WordPress, 11 Diciembre 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>. [Último acceso: 30 Agosto 2021].
- [35] R. Concepción, «¿Qué lenguaje de programación usa Arduino?,» rjconcepcion, 16 Julio 2020. [En línea]. Available: <https://www.rjconcepcion.com/podcast/que-lenguaje-de-programacion-usa-arduino/>. [Último acceso: 5 Febrero 2022].

- [36] L. Bies, «Hayes modem AT command set,» Lammert Bies, s/f. [En línea]. Available: <https://www.lammertbies.nl/comm/info/hayes-at-commands>. [Último acceso: 31 Agosto 2021].
- [37] s/a, «HC-03/05 Embedded Bluetooth Serial Communication Module,» Abril 2011. [En línea]. Available: <https://content.instructables.com/ORIG/FKY/Z0UT/HX7OYY7I/FKYZ0UTHX7OYY7I.pdf>. [Último acceso: 31 Agosto 2021].
- [38] B. Siepert, «MPU6050 6-DoF Accelerometer and Gyro,» A Minority and Woman-owned Business Enterprise (M/WBE), 6 Noviembre 2019. [En línea]. Available: <https://learn.adafruit.com/mpu6050-6-dof-accelerometer-and-gyro/arduino>. [Último acceso: 29 Agosto 2021].
- [39] jlquijado, «LA LIBRERÍA LedControl,» El desván de Jose - Recursos para programadores, 7 Abril 2016. [En línea]. Available: <https://eldesvandejose.com/2016/04/07/la-libreria-ledcontrol/>. [Último acceso: 31 Agosto 2021].