ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA AUTOMOTRIZ

"CONSTRUCCIÓN DE UN DATALOGGER PARA RECOLECCIÓN DE DATOS DE CONSUMO ENERGÉTICO DE UN VEHÍCULO ELÉCTRICO"

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO AUTOMOTRIZ

AUTORES: ALEXIS JAVIER GONZÁLEZ ROSILLO

WILSON OSWALDO ORTIZ ESPINOZA

DIRECTOR: ING. LUIS FERNANDO BUENAÑO MOYANO

Riobamba – Ecuador

2024

Nosotros, Alexis Javier González Rosillo y Wilson Oswaldo Ortiz Espinoza, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 17 de enero del 2024

Alexis Javier González Rosillo

C. I: 110535195-9

Wilson Oswaldo Ortiz Espinoza

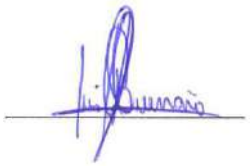
C. I: 050323555-8

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE MECÁNICA

CARRERA INGENIERÍA AUTOMOTRIZ

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, **"CONSTRUCCIÓN DE UN DATALOGGER PARA RECOLECCIÓN DE DATOS DE CONSUMO ENERGÉTICO DE UN VEHÍCULO ELÉCTRICO"**, realizado por los señores: **ALEXIS JAVIER GONZÁLEZ ROSILLO Y WILSON OSWALDO ORTIZ ESPINOZA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	**FECHA**
Ing. Juan Carlos Rocha Hoyos **PRESIDENTE DEL TRIBUNAL**		2024-01-17
Ing. Luis Fernando Buenaño Moyano **DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2024-01-17
Ing. Fabián Celso Gunsha Maji **ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**		2024-01-17

DEDICATORIA

Este trabajo está dedicado a mi familia en especial a mis padres, Lucia Rosillo y Desiderio González.

Alexis

A Dios por las bendiciones que recibo de él cada día, ya que él es la fuerza que me inspira día a día para seguir superándome como verdadero ser humano, a mis padres Oswaldo Ortiz y Anita Espinoza, por haber brindado todo su apoyo durante este periodo y a todas las personas que han estado pendientes de mi en todos los momentos de la vida universitaria.

Wilson

AGRADECIMIENTO

Agradezco a mis padres, Lucia Rosillo y Desiderio González, por ser mis apoyos principales en el transcurso de mis estudios, al Ing. Fabian Gunsha por el asesoramiento brindado durante todo el transcurso del proyecto, aquellos docentes que me han formado académicamente para lograr mis objetivos durante mis estudios en educación general básica, bachillerato y universidad, a mis abuelas Amelia Labanda y Eloisa Rosillo, mi tío Sixto González y hermana Yazmín González por el apoyo para seguir adelante en la carrera y finalmente a mis parientes y amigos que me han animado en aquellos momentos que lo necesitaba.

Alexis

Agradezco a Dios por haberme dado la vida, a mis padres Oswaldo Ortiz y Anita Espinoza, por siempre contar con su cariño, afecto y apoyo incondicional durante el lapso de los estudios universitarios, a todos los educadores de la prestigiosa Escuela Superior Politécnica de Chimborazo “ESPOCH” de manera especial al ingeniero Luis Buenaño e ingeniero Fabian Gunsha Mgtr, por su paciencia, apoyo incondicional, estímulo e incentivo, para hacer posible la culminación de mis objetivos propuestos y me ha ayudado a superar cada día como persona con valores éticos y morales para poder desenvolver en la sociedad.

Wilson

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS x
ÍNDICE DE ILUSTRACIONES xi
ÍNDICE DE ANEXOS xiv
RESUMEN xv
SUMMARY xvi
INTRODUCCIÓN 1

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA 2
1.1. Planteamiento del Problema 2
1.2. Justificación 3
1.3. Objetivos 4
1.3.1. *Objetivo general* 4
1.3.2. *Objetivos específicos* 4

CAPÍTULO II

2. MARCO TEÓRICO 5
2.1. Fuerzas de arrastre 5
2.2. Cargas de resistencia al movimiento 5
2.2.1. *Carga aerodinámica* 5
2.2.1.1. *Carga de resistencia a la rodadura* 6
2.2.1.2. *Carga de resistencia por la pendiente* 7
2.3. Modelos de consumo de energía para vehículos eléctricos 7
2.4. Componentes de un sistema electrónico de medida 11
2.4.1. *Sensor o transductor* 11
2.4.1.1. *Principios de transducción* 12
2.4.1.2. *Características estáticas del transductor* 13
2.4.1.3. *Características dinámicas del transductor* 14
2.4.2. *Acondicionador de señal* 14
2.4.2.1. *Tipos de acondicionamientos* 15
2.4.3. *Procesamiento de la señal* 16
2.4.3.1. *Microcontrolador* 17

2.4.3.2. *Lenguaje de programación* 18
2.4.4. ***Presentación de la información*** 18
2.5. **Sistemas electrónicos de medida** 19
2.5.1. ***Instrumentos de propósito general*** 19
2.5.2. ***Sistemas de adquisición de datos*** 20
2.5.2.1. *Protocolos de comunicación* 20
2.5.3. ***Instrumentos virtuales*** 22
2.6. **Sistema Global de Navegación por Satélite (GNSS)** 23
2.6.1. ***Protocolo NMEA 0183*** 23
2.7. **Almacenamiento** 24
2.7.1. ***Memoria Secure Digital (SD)*** 25

CAPÍTULO III

3. **MARCO METODOLÓGICO** 26
3.1. **Diagrama de etapas del proyecto** 26
3.2. **Selección de parámetros o magnitudes físicas a medir u obtener y almacenar ...** 27
3.3. **Selección de los dispositivos eléctricos para obtener y almacenar las medidas** ... 28
3.3.1. ***Sensor de presión ambiente*** 28
3.3.2. ***Sensor de temperatura ambiente*** 29
3.3.3. ***Anemómetro y sensor de dirección de viento*** 30
3.3.4. ***Sensor de corriente*** 31
3.3.5. ***Sensor de voltaje*** 31
3.3.6. ***Modulo receptor GNSS*** 32
3.3.7. ***Modulo micro SD*** 33
3.3.8. ***Placas electrónicas con microcontrolador*** 33
3.3.9. ***Interfaz gráfica o pantalla*** 35
3.4. **Diseño del datalogger** 35
3.4.1. ***Diseño del procesamiento de señales directas de sensores con placa principal*** 37
3.4.1.1. *Circuito de sensores con la placa principal* 37
3.4.1.2. *Descripción del algoritmo de la placa principal con sensores directos* 39
3.4.1.3. *Simulación de sensores virtuales para la comprobación del algoritmo* 43
3.4.2. *Diseño para la medición de velocidad y dirección del viento (subcircuito 1)* 45
3.4.2.1. *Circuito para el anemómetro* 45
3.4.2.2. *Descripción del algoritmo para el anemómetro* 46
3.4.2.3. *Simulación del anemómetro del subcircuito uno* 48
3.4.3. ***Diseño para recepción de datos de posicionamiento (subcircuito 2)*** 49

3.4.3.1. *Circuito para el módulo receptor GNSS* 49
3.4.3.2. *Descripción del algoritmo para el módulo GNSS* 50
3.4.3.3. *Simulación del módulo GNSS o del subcircuito dos*..... 52
3.4.4. ***Diseño de la interfaz gráfica*** 52
3.4.4.1. *Circuito para la interfaz gráfica o pantalla* 52
3.4.4.2. *Descripción del algoritmo de la interfaz gráfica*..... 53
3.4.4.3. *Simulación del circuito de la interfaz gráfica* 60
3.4.5. ***Diseño del almacenamiento de las magnitudes físicas***..... 61
3.4.5.1. *Circuito para el módulo micro SD* 61
3.4.5.2. *Descripción del algoritmo para el módulo micro SD*..... 62
3.4.5.3. *Simulación del almacenamiento de los parámetros seleccionados*..... 65
3.4.6. ***Diseño del envío y recepción de datos entre placas Arduino***. 66
3.4.6.1. *Circuito para el envío y recepción de datos*..... 66
3.4.6.2. *Descripción del algoritmo para la comunicación entre placas Arduino* 66
3.4.6.3. *Simulación de la comunicación entre placas Arduino* 68
3.4.7. *Diseño del circuito de alimentación y carga*..... 69
3.4.7.1. *Dimensionamiento de baterías* 69
3.4.7.2. *Circuito de alimentación y carga* 71
3.5. **Construcción de datalogger** 72

CAPÍTULO IV

4. **ANÁLISIS E INTERPRETACIÓN DE RESULTADOS**..... 75
4.1. **Resultados de pruebas de almacenamiento de datos**..... 75
4.2. **Resultados de la presentación de información e interacción con interfaz gráfica**.. 80

CAPÍTULO V

5. **CONCLUSIONES Y RECOMENDACIONES**..... 82
5.1. **Conclusiones**..... 82
5.2. **Recomendaciones**..... 84

BIBLIOGRAFÍA
ANEXOS

ÍNDICE DE TABLAS

Tabla 2-1: Mensajes NMEA 24
Tabla 2-2: Caracteres utilizados para la estructura de sentencias NMEA. 24
Tabla 3-1: Parámetros seleccionados para la medición y almacenamiento 28
Tabla 3-2: Componentes principales del datalogger 35
Tabla 3-3: Circuitos principales para el datalogger 37
Tabla 3-4: Descripción de las variables pare el algoritmo del anemómetro 46
Tabla 3-5: Variables tipo entera para la recolección de datos de módulo GNSS. 51
Tabla 3-6: Variables declaradas para el algoritmo de la interfaz grafica 53
Tabla 3-7: Asignación de frases el Software Nextion Editor. 55
Tabla 3-8: Objetos tipo botón declarados en el software Nextion Editor y variable para algoritmo de la placa principal 55
Tabla 3-9: Acciones a tomar de acuerdo al valor de uno asignado en la variable *lectura*.. 57
Tabla 3-10: Visualización de las opciones para cada sensor 57
Tabla 3-11: Matriz de acciones para cada sensor. 58
Tabla 3-12: Opciones de tiempo de muestro o registro de datos. 58
Tabla 3-13: Visualización de las medidas o parámetros físicos obtenidos. 59
Tabla 3-14: Estimación de consumo de componentes del datalogger. 70
Tabla 4-1: Resultado de mediciones y registro de parámetros seleccionados. 78

ÍNDICE DE ILUSTRACIONES

Ilustración 2-1: Fuerzas y momentos aerodinámicos que actúan sobre el vehículo. 6
Ilustración 2-2: Componentes del peso del vehículo en una pendiente. 7
Ilustración 2-3: Diagrama de los principios de transducción..... 13
Ilustración 2-4: Acondicionamiento por linealización..... 16
Ilustración 2-5: Acondicionamiento por filtrado. 16
Ilustración 2-6: Acondicionamiento por conversión analógica-digital..... 16
Ilustración 2-7: Clasificación de los sistemas electrónicos de medida. 19
Ilustración 2-8: Conexión de maestro-esclavo en comunicación SPI..... 21
Ilustración 2-9: Señales del protocolo de comunicación SPI..... 21
Ilustración 2-10: Señal de protocolo de comunicación UART. 22
Ilustración 2-11: Conexión de maestro y esclavos para la comunicación I2C. 22
Ilustración 2-12: Memoria SD y micro SD..... 25
Ilustración 3-1: Diagrama de flujo de la construcción del Datalogger. 26
Ilustración 3-2: Sensor de presión BMP280. 29
Ilustración 3-3: Sensor de temperatura LM35A. 29
Ilustración 3-4: Anemómetro Davis Vantage Pro 2 6410..... 30
Ilustración 3-5: Sensor de corriente DHAB S/137. 31
Ilustración 3-6: Sensor de voltaje DVC JXDA4U..... 32
Ilustración 3-7: Módulo receptor GNSS NS-HP-GN2 32
Ilustración 3-8: Antena de alta precisión para modulo receptor GNSS 33
Ilustración 3-9: Módulo micro SD y pines de comunicación SPI y alimentación. 33
Ilustración 3-10: Arduino Mega 2560 34
Ilustración 3-11: Arduino Pro Mini 34
Ilustración 3-12: Pantalla NEXTION lNX4827K043_011R. 35
Ilustración 3-13: Diagrama de bloques del sistema datalogger. 36
Ilustración 3-14: Circuito de sensores con Arduino Mega 2560. 38
Ilustración 3-15: Comparación y asignación del valor de corriente medida..... 41
Ilustración 3-16: Definición de pin y variable para los puntos de interés..... 42
Ilustración 3-17: Tipo de lectura e interrupción para el pin D3 de la placa Arduino Mega. ... 42
Ilustración 3-18: Función creada para el conteo de puntos de interés 43
Ilustración 3-19: Resultados de simulación de sensor de temperatura LM35. 43
Ilustración 3-20: Resultados de simulación de sensor de corriente DHAB S/137..... 44
Ilustración 3-21: Resultados de simulación de sensor de voltaje JXDA4U. 44
Ilustración 3-22: Resultados de simulación de sensor de presión absoluta BMP280. 45

Ilustración 3-23: Circuito del anemómetro. 45
Ilustración 3-24: Pines de señal de entrada para el anemómetro. 46
Ilustración 3-25: Variables declaradas en el algoritmo del anemómetro 46
Ilustración 3-26: Procesos generados en el *void setup()* para el algoritmo del anemómetro ... 47
Ilustración 3-27: Inicialización de temporizador para anemómetro. 47
Ilustración 3-28: Conteo de interrupciones del sensor de velocidad del viento 48
Ilustración 3-29: Proceso de la función *isr_tiempo()* para el sensor de velocidad del viento .. 48
Ilustración 3-30: Resultados de simulación de sensor de dirección del viento del 49
Ilustración 3-31: Resultados de simulación de sensor de dirección del viento del 49
Ilustración 3-32: Circuito del módulo receptor GNSS. 50
Ilustración 3-33: Lectura de datos NMEA por interrupción. 50
Ilustración 3-34: Resultados de simulación del módulo GNSS. 52
Ilustración 3-35: Circuito de la interfaz gráfica 52
Ilustración 3-36: Estructura para envío de datos tipo texto, hacia la pantalla Nextion. 54
Ilustración 3-37: Estructura para envío de datos tipo número a la pantalla Nextion. 60
Ilustración 3-38: Declaración y posición de objetos en el software Nextion Editor. 60
Ilustración 3-39: Simulación de interfaz gráfica. 61
Ilustración 3-40: Circuito del módulo de la micro SD. 62
Ilustración 3-41: Función para almacenamiento de datos en tarjeta micro SD. 63
Ilustración 3-42: Inicialización de TIMER4 64
Ilustración 3-43: Algoritmo de la función para el registro de datos 64
Ilustración 3-44: Medición de tiempo de registro de datos para 1 segundo. 65
Ilustración 3-45: Medición de tiempo de registro de datos para 2 segundo. 65
Ilustración 3-46: Resultados de simulación del registro de los parámetros seleccionados. 66
Ilustración 3-47: Circuito para la comunicación entre placas Arduino. 66
Ilustración 3-48: Inicialización de comunicación UART para Arduino Mega 2560. 67
Ilustración 3-49: Inicialización de comunicación UART en Arduino Pro Mini. 67
Ilustración 3-50: Trama de datos para envío de datos de velocidad y dirección del viento. ... 67
Ilustración 3-51: Envió de medida de dirección y velocidad del viento 68
Ilustración 3-52: Recepción de datos del anemómetro y GNSS en la placa principal Arduino Mega. 68
Ilustración 3-53: Simulación de la transmisión y recepción de datos entre placas Arduino 69
Ilustración 3-54: Batería 18650 de Li-ion 71
Ilustración 3-55: Circuito principal de alimentación y carga 72
Ilustración 3-56: Prototipo Datalogger 72
Ilustración 3-57: Diseño de placa PCB 73
Ilustración 3-58: Caja para la implementación del datalogger 74

Ilustración 4-1: Cantidad de datos almacenados respecto al tiempo registro de datos acumulado, para un segundo de muestreo 76
Ilustración 4-2: Cantidad de datos almacenados respecto al tiempo registro de datos acumulado, para dos segundos de muestreo. 76
Ilustración 4-3: Estructura de registro de datos. 79
Ilustración 4-4: Secuencia de presentación de la información en la interfaz Gráfica. 79
Ilustración 4-5: Partes de la presentación de la información en la interfaz gráfica 80
Ilustración 4-6: Botones para la interacción entre pantalla y usuario. 81

ÍNDICE DE ANEXOS

ANEXO A: ESPECIFICACIONES SENSOR DE PRESIÓN BMP280

ANEXO B: ESPECIFICACIONES SENSOR DE TEMPERATURA LM35A

ANEXO C: ESPECIFICACIONES DEL ANEMOMETRO VANTAGE PRO2 6410

ANEXO D: ESPECIFICACIONES DEL SENSOR DE CORRIENTE DHAB S/137

ANEXO E: ESPECIFICACIONES DEL SENSOR DE VOLATJE JXDA4U

ANEXO F: ESPECIFICACIONES DEL RECEPTOR GNSS PX1122R

ANEXO G: ESPECIFICACIONES DE LA ANTENA PARA EL MODULO RECEPTOR GNSS

ANEXO H: ESPECIFICACIONES DE LA PLACA ARDUINO MEGA 2560

ANEXO I: ESPECIFICACIONES DE LA PLACA ARDUINO PRO MINI

ANEXO J: ESPECIFICACIONES DE LA PANTALLA NEXTION lNX4827K043_011R

ANEXO K: DIAGRAMA DE FLUJO DE LA PROGRAMACIÓN DEL DATALOGGER CON ATMEGA 2560

ANEXO L: DIAGRAMA DE FLUJO (1) ANEMÓMETRO, (2) MÓDULO GNSS

ANEXO M: CIRCUITO COMPLETO DE DATALOGGER

ANEXO N: PCB PARA DATALOGGER

ANEXO O: FRAGMENTOS DE RESULTADO DE REGISTRO DE DATOS EN FORMATO .XLSX, PARA INTERVALOS DE REGISTRO DE 1 SEGUNDO

ANEXO P: FRAGMENTOS DE RESULTADO DE REGISTRO DE DATOS EN FORMATO .XLSX, PARA INTERVALOS DE REGISTRO DE 2 SEGUNDOS

ANEXO Q: MANUAL DE USUARIO DE DATALOGGER

RESUMEN

La carrera de ingeniería automotriz no cuenta con un dispositivo capaz de medir y almacenar datos relacionados con variables físicas de fenómenos que influyen en las pérdidas de energía durante el movimiento de un vehículo eléctrico, lo cual es un limitante para la realización de estudios académicos referente a la validación de modelos matemáticos y generación de ciclos de conducción. El presente trabajo de integración curricular tuvo como objetivo construir un datalogger con sensores no invasivos para la recolección de datos de consumo energético de un vehículo eléctrico. Para construir el datalogger se desarrolló una parte física y una virtual. La parte física consta de una caja, una placa PCB, una placa Arduino Mega2560, una placa Arduino Pro-Mini, sensores: BMP280, LM35A, Davis Vantage Pro2-6410, DHAB S/137 y JXDA4U; módulo GNSS, pulsador y pantalla INX4827K043_011R; la parte virtual consta de: algoritmo desarrollado en Arduino ID e interacción con la interfaz gráfica. Se adaptó el datalogger a un vehículo y un pack de baterías, se realizaron pruebas de: registro y almacenamiento de datos e interacción con la interfaz gráfica. Se obtuvo una relación de 1 dato/segundo y 0.5 dato/segundo para 1 segundo y 2 segundos de muestreo respectivamente; los datos se almacenaron en un archivo de texto en una estructura definida con un separador entre cada dato; se pudo interactuar con la pantalla en la selección de las variables a registrar junto con la presentación de la información. De los resultados alcanzados se concluyó que el datalogger es capaz de registrar los datos durante el funcionamiento real del dispositivo con un tiempo de muestreo de 1 o 2 segundos, almacenarlos en archivo con formato de texto en forma ordenada y presentar la información en tiempo real, manteniendo un grado de precisión acorde con las especificaciones de los fabricantes.

Palabras clave: <DATALOGGER>, <CONSUMO ENERGÉTICO>, <ARDUINO ID>, <VARIABLES FÍSICAS>, <CIRCUITOS ELÉCTRICOS>, <PLACA DE CIRCUITOS IMPRESOS>.

SUMMARY

The automotive engineering career does not have a device capable of measuring and storing data related to physical variables of phenomena that influence energy losses during the movement of an electric vehicle, which is a limitation for the realization of academic studies related to the validation of mathematical models and generation of driving cycles. The objective of this curricular integration work was to build a datalogger with non-invasive sensors for the collection of energy consumption data of an electric vehicle. To build the datalogger, a physical and a virtual part was developed. The physical part consists of a box, a PCB board, an Arduino Mega2560 board, an Arduino Pro-Mini board, sensors: BMP280, LM35A, Davis Vantage Pro2-6410, DHAB S/137 and JXDA4U; GNSS module, push button and display INX4827K043_011R; the virtual part consists of: algorithm developed in Arduino ID and interaction with the graphical interface. The datalogger was adapted to a vehicle and a battery pack, tests of data recording and storage and interaction with the graphical interface were performed. A ratio of 1 data/second and 0.5 data/second was obtained for 1 second and 2 seconds of sampling respectively. The data were stored in a text file in a defined structure with a separator between each data; it was possible to interact with the screen in the selection of the variables to be recorded along with the presentation of the information. From the results achieved it was concluded that the datalogger is capable of recording data during the actual operation of the device with a sampling time of 1 or 2 seconds, storing them in a text format file in an orderly manner and presenting the information in real time, maintaining a degree of accuracy according to the manufacturers' specifications.

Keywords: <DATALOGGER>, <ENERGY CONSUMPTION>, <ARDUINO ID>, <PHYSICAL VARIABLES>, <ELECTRIC CIRCUITS>, <PRINTED CIRCUIT BOARD>.

Lic. Sandra Paulina Porras Pumalema Msc
C.I. 0603357062

INTRODUCCIÓN

Un tema fundamental en la movilidad eléctrica en vehículos eléctricos es la energía. La eficiencia de conversión de la energía de un vehículo eléctrico en comparación a un vehículo a combustión interna es muy alta; un vehículo de motor de combustión interna (MCI) a diésel tiene una eficiencia menor al 40%, a gasolina tiene una eficiencia menor al 30% y un vehículo eléctrico tiene una eficiencia del 90%; entre el 60% y 70% de la energía utilizada en vehículos de motor de combustión interna (MCI) se disipa en forma de calor y del 30% al 40% se utiliza para conllevar las cargas en carretera (Gil y Prieto, 2013, pp. 44-45).

La movilidad eléctrica en el país creció en los últimos años de acuerdo con los datos de venta de vehículos eléctricos registrados por la Asociación de Empresas Automotrices del Ecuador (AEADE). En el año 2018 se han vendido 130 vehículos y para el año 2022 se han vendido 405 vehículos, esto quiere decir que en el año 2022 se han vendido 3.1 veces más vehículos eléctricos en comparación al año 2018, desde el año 2017 hasta el año 2022 se tiene en total dentro del país 1215 vehículos eléctricos (AEADE, 2022, p.102).

Realizar estudios y análisis respecto a la energía consumida en la conducción en una ruta bajo ciertas condiciones permite estimar la energía consumida de forma más real y con ello obtener datos más precisos de consumo, para ello se requiere de dispositivos que permitan obtener los datos necesarios para la estimación de la energía consumida en un vehículo, es por ello que el presente proyecto tiene como finalidad construir un dispositivo que facilite el registro de datos relacionados al consumo de energía para vehículos eléctricos.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Planteamiento del Problema

En los últimos años a nivel mundial se ha implementado mecanismos de reducción de contaminación, siendo los vehículos de combustión interna los más representativos. En Ecuador se implementaron leyes que favorecen a la comercialización de vehículos híbridos y eléctricos (Suplemento Registro Oficial, 2019), impulsando a una mayor demanda; siendo la recolección de variables físicas (presión, temperatura, velocidad del viento, posicionamiento, etc.) un aspecto primordial para en el estudio de estos mecanismos.

La falta de información de datos relacionados al consumo de energía en vehículos eléctricos limita el desarrollo de estudios enfocados en la eficiencia, autonomía y puntos estratégicos de recarga, lo cual hace imposible determinar la sostenibilidad de estos vehículos de acuerdo con los recursos energéticos predispuestos en el país. En otros países que tienen estudios relacionados al consumo en vehículos eléctricos se implementan de mejor manera y de forma rápida estrategias para la adaptación y uso de la movilidad eléctrica, a diferencia de Ecuador que no presenta este tipo de estudios lo cual hace que sea muy complicado aplicar adecuadamente una movilidad basada en energías renovables, en especial para el caso de los vehículos eléctricos.

Estimar la cantidad de energía es difícil al no contar con dispositivos específicos para la medición de los fenómenos que interviene en las pérdidas de energía sobre el vehículo eléctrico durante su funcionamiento, esto hace que sea necesario disponer de un dispositivo con la capacidad de medir y registrar los parámetros que permiten cuantificar la energía empleada para mover el vehículo bajo determinadas condiciones como: tráfico, ambiente y vía. En la carrera de ingeniería automotriz no se cuenta con un dispositivo con la capacidad de medir y almacenar datos directamente relacionados con las variables físicas de los fenómenos que influyen en las pérdidas de energía en una ruta, por lo que es difícil académicamente realizar estudios respecto a la movilidad eléctrica por lo cual se ve la necesidad de construir un dispositivo datalogger para futuros estudios académicos relacionados a la recolección de datos para modelos matemáticos y generación de ciclos de conducción.

1.2. Justificación

De acuerdo con los datos de proyecciones de crecimiento en el parque automotor eléctrico, se estima que para el año 2025 en el país se tendrá aproximadamente 10000 vehículos eléctricos, entre los cuales se encontrarían buses públicos, taxis, camiones de carga liviana y vehículos ligeros; para años posteriores al 2025 hasta el año 2040 se estima que la movilidad eléctrica dominará el sector de transporte, dejando atrás los vehículos de combustión interna (Hinicio, 2021, pp.3-4). De acuerdo con la información anterior se estimó que el parque automotor crecerá considerablemente, aunque en los últimos años haya crecido un 12.15 %, en relación con la meta establecida en el año 2025.

Determinar el consumo de energía en los vehículos eléctricos permite estimar la eficiencia, autonomía aproximada, precisar los puntos estratégicos de recargas y la sostenibilidad energética. Para estimar el consumo de energía se debe cuantificar las pérdidas de energía en ruta provocadas por: resistencia del aire, resistencia a la rodadura y pendiente de carretera. Este proyecto está encaminado en el desarrollo de un dispositivo datalogger para la recolección de parámetros requeridos para el cálculo y estimación del consumo energético y con ello posibilitar una mejor estimación de la energía utilizada en cualquier tipo de ruta predeterminada. El desarrollo de este trabajo de integración curricular sirve de base para realizar estudios futuros dentro de la carrera de Ingeniería Automotriz enfocados en la recolección de datos para modelos matemáticos de consumo y ciclos de conducción para vehículos eléctricos. El dispositivo tendrá la capacidad de medir, visualizar y almacenar datos de corriente, voltaje de batería, velocidad de viento, dirección del viento, velocidad del vehículo, presión ambiente, temperatura ambiente, posicionamiento e identificar puntos de interés.

La construcción del datalogger está basada en un dispositivo electrónico de medida con la particularidad de adquirir y almacenar los datos obtenidos por sensores, un pulsador como contador y un receptor GNSS. El dispositivo es definido como datalogger debido a la particularidad de almacenar las variables medidas. Los sensores y el dispositivo receptor GNSS se seleccionarán de acuerdo con el tipo de variable física a medir, disponibilidad dentro del mercado, facilidad de programación e instalación en el vehículo y costo. El diseño del dispositivo parte principalmente de: selección de parámetros físicos a medir o datos a registrar y selección de los dispositivos para obtener y almacenar los parámetros seleccionados. El diseño del datalogger se divide en tres partes: diseño de los circuitos electrónicos, programación y simulación. Luego de tener el diseño del datalogger se realizará la construcción, prueba de almacenamiento y finalmente genera un análisis respecto a los resultados dados en la prueba.

1.3. Objetivos

1.3.1. Objetivo general

Construir un datalogger para recolección de datos de consumo energético de un vehículo eléctrico.

1.3.2. Objetivos específicos

- Recopilar información por medio de una revisión bibliográfica relacionada a principios de funcionamiento de sensores, microcontrolador y programación para la construcción del datalogger.
- Diseñar el circuito electrónico del datalogger mediante Proteus.
- Programar el datalogger mediante el software adecuado para la recolección de datos.
- Desarrollar la PCB mediante herramientas de soldadura para el ensamblaje del datalogger.
- Desarrollar pruebas de funcionamiento del datalogger implementándolo en el vehículo para el correcto registro de datos.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Fuerzas de arrastre

Son aquellas fuerzas y los momentos que intervienen en el desplazamiento del vehículo, siendo la dirección longitudinal (eje x) fuerza positiva, mientras que cuando actúa en dirección negativa (eje -z) es ascendente. Según SAE J670e, define la fuerza normal a la que interviene en dirección hacia abajo y fuerza vertical como negativa de la normal, por ende, dicha fuerza es equivalente a la carga en el neumático.

Es indispensable tener en cuenta el eje dinámico de carga, en el que involucra a la segunda ley de Newton ya que analiza las cargas por eje determinando el esfuerzo de tracción; alterando la aceleración, pendiente, velocidad máxima y esfuerzo, en base al análisis de rendimiento de aceleración y frenado (Gillespie, 1992, pp. 10-11).

2.2. Cargas de resistencia al movimiento

Son aquellas fuerzas que actúan sobre el vehículo y se oponen al movimiento. En general actúan tres cargas importantes: carga aerodinámica, carga de resistencia a la rodadura y carga debido a la pendiente (Gillespie, 1992, pp. 118-119).

2.2.1. Carga aerodinámica

La carga aerodinámica aplica sobre el vehículo arrastre, elevación, momentos de balanceo, fuerzas laterales, ruido y guiñada. La fuerza de arrastre actúa de forma longitudinal, la fuerza lateral actúa de forma lateral y la elevación actúa de forma vertical; en la Ilustración 2-1 se presentan estas fuerzas y momentos aerodinámicos que actúan sobre el vehículo.

La fuerza de arrastre se considera como la fuerza que actúa en mayor medida sobre el vehículo. Para determinar la fuerza de arrastre aerodinámico se recurre a un modelo semi-empírico que permite representar el efecto sobre el vehículo (Gillespie, 1992, pp.87-89). Se utiliza la Ec.(1).

$$D_A = {}^1\!/_2 \cdot \rho_{aire} \cdot A_f \cdot C_d \cdot V_{Tot}^2 = ({}^1\!/_2 \cdot \rho_{aire} \cdot A_f \cdot C_d \cdot (v + v_{viento})^2 \qquad (1)$$

Donde:

D_A = arrastre o carga aerodinámica.

ρ_{aire} = densidad del aire.

A_f = área frontal del vehículo.

C_d = coeficiente de resistencia aerodinámica.

V_{Tot} = velocidad total.

v = velocidad del vehículo.

v_{viento} = velocidad del viento.

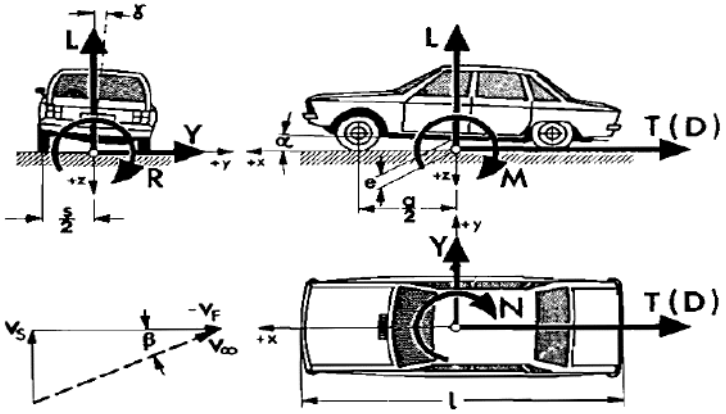


Ilustración 2-1: Fuerzas y momentos aerodinámicos que actúan sobre el vehículo.

Fuente: Guillespie, 1992, p.87.

2.2.1.1. Carga de resistencia a la rodadura

La residencia a la rodadura en terreno llano se relaciona con los neumáticos del vehículo, esta resistencia se presenta desde el momento en que los neumáticos empiezan a girar. Una gran cantidad de la potencia se transforma en energía en forma de calor. La resistencia a la rodadura se determina tomando en cuenta las resistencias en las cuatro ruedas del vehículo (Gillespie, 1992, pp.110-111). La Ec.(2) se aplica para determinar esta carga.

$$R_{rod} = R_{rod\,f} + R_{rod\,r} = f_r \cdot W \qquad (2)$$

Donde:

R_{rod} = resistencia o carga debida a la rodadura.

$R_{rod\,f}$ = resistencia a la rodadura en ruedas frontales.

$R_{rod\,r}$ = resistencia a la rodadura en ruedas traseras.

W = peso del vehículo.

f_r = coeficiente de resistencia a la rodadura.

2.2.1.2. Carga de resistencia por la pendiente

Esta resistencia aparece cuando el vehículo toma trayectos con un grado de inclinación. La fuerza generada por la pendiente se opone al ascenso del vehículo debido a que aparece una componente del peso del vehículo en dirección contraria al movimiento, como se muestra en la **¡Error! No se encuentra el origen de la referencia.** (Rodríguez y Villar, 2017, p.8).

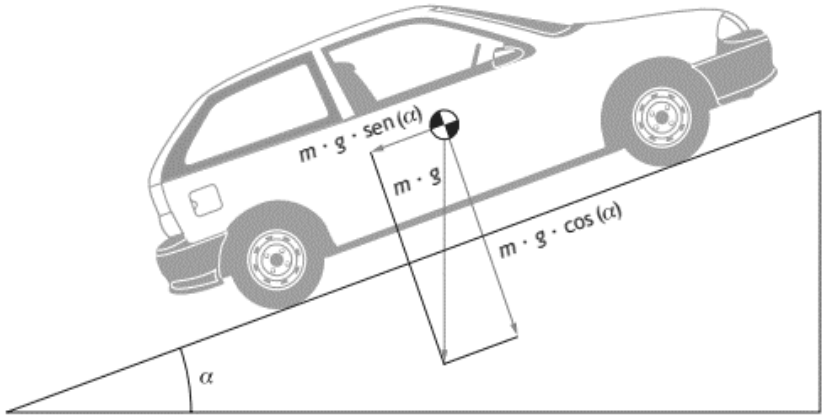


Ilustración 2-2: Componentes del peso del vehículo en una pendiente.

Fuente: Rodríguez y Villar, 2017, p.8.

Cabe decir que en este documento el ángulo debido a la pendiente será θ, por lo tanto:

$$\alpha = \theta \quad (3)$$

Para determinar esta carga se utiliza la Ec. (4).

$$F_{pend} = m \cdot g \cdot \sin(\theta) \quad (4)$$

Donde:

m = masa del vehículo.

g = aceleración gravitacional.

θ = ángulo de la pendiente de la calzada o vía.

2.3. Modelos de consumo de energía para vehículos eléctricos

Es importante analizar los modelos de consumo de energía para vehículos eléctricos, debido a que en estos modelos se aplican ecuaciones que se relacionan entre si con el fin de determinar la energía consumida, para ello cada ecuación contiene parámetros que permiten realizar los cálculos. Describir estos modelos ayuda a establecer que parámetros se pueden medir y precisar la cantidad de energía consumida por el vehículo eléctrico.

De acuerdo con la revisión de la literatura se encontraron algunos modelos para determinar el consumo de energía y el estado de carga de la batería. Estos modelos utilizados tienen como fin determinar el consumo de energía en vehículos eléctricos por medio de la aplicación de ecuaciones.

Se desarrolla un modelo VT-CPEM para estimar el consumo de energía por medio de un ciclo de conducción, la potencia consumida instantánea y el estado de carga de la batería (SOC) (Fiori, Ahn y Rakha, 2016, p.260). Se aplica la Ec.(5) para obtener la potencia en las ruedas por medio de parámetros relacionados a las cargas en carretera.

$$P_{ruedas}(t) = \Big(m \cdot a(t) + m \cdot g \cdot \cos(\theta) \cdot \frac{C_r}{1000}(c_1 \cdot v(t) + c_2) + {}^1\!/_2 \cdot \rho_{aire} \cdot A_f \cdot C_d \cdot v^2(t) + m \cdot g \cdot \sin(\theta)\Big) \cdot v(t) \quad (5)$$

Donde:

P_{ruedas}= potencia en las ruedas.
a = aceleración en el vehículo.
C_r, c_1, c_2 = parámetros relacionados a la resistencia a la rodadura.

En este modelo P_{ruedas} es la potencia en las ruedas [W], la masa en este modelo está en [kg], la aceleración en [m/s^2], la gravedad en [m/s^2], la densidad del aire en [kg/m^3], el área frontal en [m^2] y la velocidad del vehículo en [m/s].

La potencia del motor eléctrico $P_{Motor\ electrico}$ se determina considerando la eficiencia en la línea de transmisión del motor eléctrico a las ruedas $\eta_{tren\ motriz}$, como se muestra en la Ec.(6).

$$P_{Motor\ electrico}(t) = P_{ruedas}(t) \cdot \eta_{tren\ motriz} \quad (6)$$

Se puede estimar por medio de este modelo el estado de carga final de la batería (SOC) aplicando la Ec.(7) y Ec.(8).

$$SOC_{final}(t) = SOC_0 - \sum_{i=1}^{N} \Delta SOC_{(i)}(t) \quad (7)$$

$$\Delta SOC_{(i)}(t) = SOC_{(i-1)}(t) - \frac{P_{Motor\ electrico_{net}}(t)}{3600 * Capacidad_{Bateria}} \quad (8)$$

Donde:

SOC_{final} = estado de carga final de la batería.

SOC_0 = estado de carga inicial de la batería.

$P_{Motor\ electrico_{net}}$ = potencia consumida.

$Capacidad_{Bateria}$ = capacidad de la batería.

Por último, la Ec.(9) permite determinar el consumo de energía.

$$EC = \frac{1}{3600000} \cdot \int_0^t P_{Motor\ electrico_{net}}(t)dt \cdot \frac{1}{d} \quad (9)$$

Donde:

EC = energía consumida.

d = distancia.

Se encontró otro modelo utilizado para determinar el consumo de energía y el rendimiento de la batería de buses eléctricos, se simula utilizando la plataforma Autonomie desarrollado por el Laboratorio Nacional de Argonne con el apoyo del DOE. En este modelo se simula la eficiencia en el consumo de energía de vehículos de transporte pesado de clase 8. En el modelo planteado se toma en cuenta la perdida aerodinámica, de rodadura, pendiente, aceleración de avance y la inercia rotacional; dando como resultado la Ec.(10) (Gao et al. 2017, pp.590-951).

$$P_{ruedas} = m \cdot V \cdot \frac{dv}{dt} + \frac{1}{2}\rho_{aire} \cdot C_d \cdot A_f \cdot v^3 + m \cdot g \cdot f_r \cdot v + m \cdot g \cdot v \cdot \sin(\theta) \quad (10)$$

En cuanto al estado de carga de la batería (SOC) se utiliza la Ec.(11).

$$SOC = 1 - \int_0^t \left(P_{dis}^e - P_{chg}^e\right) dt / E_{batt} \quad (11)$$

Donde:

SOC = estado de carga de la batería.
P_{dis}^e = potencia de descarga de la batería.
P_{chg}^e = potencia de carga de la batería.
E_{batt} = capacidad de almacenamiento de energía de la batería.

En otro documento se establece un método para determinar el consumo de energía de forma experimental junto con la eficiencia de los vehículos eléctricos, se establece una ecuación para determinar la energía en las ruedas por medio de la adaptación de un dispositivo con la capacidad de medir los parámetros relacionados a la ecuación formulada, como se muestra en la Ec.(12) (Paffumi et al., 2015, p.279).

$$P_{ruedas} = \int_{in}^{fin} \Big(m \cdot a(t) + m \cdot g \cdot \sin\theta\,(t) + \mu \cdot m \cdot g \cdot \cos\theta\,(t) + sing \cdot \left(v(t) + v_{viento}(t)\right) \frac{\rho_{aire}}{2} \cdot C_d \cdot A_f \cdot \left(v(t) + v_{viento}(t)\right)^2 \Big) \cdot v(t)\, dt \quad (12)$$

Donde:

$sing$ = función de signo, determina el signo de la suma algebraica ($v + v_{viento}$).

Cabe resaltar que en los tres modelos anteriormente descritos, en la formulación de la ecuación a implementar, se establecen algunos parámetros idénticos, entre estos se encuentran: masa del vehículo, aceleración, aceleración de la gravedad, pendiente de la carretera, densidad del aire, área fontal del vehículo, coeficiente de arrastre aerodinámico y la velocidad del vehículo.

La densidad del aire es la razón entre la masa y el volumen del aire. Esta densidad obtiene a partir de la Ec.(13) relacionada al estado de un gas ideal, se toma en cuenta al aire (Cengel y Boles 2012, p. 138).

$$Presion_{aire} \cdot Volumen_{aire} = m_{aire} \cdot R_{aire} \cdot T_{aire} \quad (13)$$

Donde:

$Presion_{aire}$ = presión del aire o presión ambiental.
$Volumen_{aire}$ = volumen o espacio ocupado por el aire.
m_{aire} = masa del aire.
R_{aire} = constante del aire.
T_{aire} = temperatura del aire o temperatura ambiental.

La densidad se obtiene a partir de la ecuación Ec.(14).

$$\rho_{aire} = \frac{m_{aire}}{Volumen_{aire}} \quad (14)$$

A partir de Ec. (13) y Ec. (14), se obtiene una fórmula matemática que permite obtener la densidad del aire a partir de la presión ambiente, temperatura ambiente y constante del aire.

$$\rho_{aire} = \frac{m_{aire}}{Volumen_{aire}} = \frac{Presión_{aire}}{R_{aire} \cdot T_{aire}} \quad (15)$$

En cuanto al cálculo del SOC de la batería del vehículo eléctrico se toma en cuenta la potencia consumida de la batería. La potencia eléctrica es el producto entre el voltaje y corriente, como se muestra en la Ec. (16).

$$P_{electrica} = V(t) \cdot I(t) \quad (16)$$

Donde:

V = voltaje de la batería.
I = corriente de la batería.

2.4. Componentes de un sistema electrónico de medida

2.4.1. Sensor o transductor

Es un dispositivo que transforma o convierte una magnitud física medida en una señal eléctrica, generalmente la señal de salida es voltaje, intensidad o impedancia. Los transductores son denominados de diferentes formas, por ejemplo, en procesos industriales se denominan transmisores, en otros casos se denominan sensores, detectores, en algunos casos células, etc. (Granda y Mediavilla, 2015, p.207).

2.4.1.1. Principios de transducción

Entre los principios de transducción más utilizados se describen a continuación (Granda y Mediavilla, 2015, pp.210-211).

- Los transductores resistivos transforman un cambio medido de magnitud en uno de resistencia.
- Los transductores potenciométricos convierten el cambio medido de magnitud en uno de tensión, por medio de una variación de un contacto móvil en un elemento resistivo.
- La transducción por galga extensiométricas cambia la magnitud medida en resistencia por medio de la deformación de galgas extensiométricas. Normalmente las galgas extensiométricas se conectan a un puente de Wheatstone con una tensión de alimentación.
- Los transductores reductivos convierten la magnitud medida en una tensión alterna por medio de la reluctancia de un circuito magnético conformado por dos o más devanados, aplicando una excitación alterna en los devanados.
- En el grupo de los transductores especiales se encuentra el sensor Hall, basado en el efecto Hall en el cual se genera un gradiente de potencial en un conductor en el que circula una corriente al aplicarse un campo magnético. Otro es la célula fotoemisiva que se basa en el efecto fotoeléctrico.
- Los transductores piezoeléctricos convierten la magnitud medida en un cambio de tensión, se presenta en ciertos materiales al aplicar un esfuerzo mecánico.
- Los transductores con el principio piezorresistivo se basan en la relación entre la resistencia eléctrica y la deformación, es decir que, cuando un material recibe una deformación su resistividad interna se modifica. La resistividad es distinta a la resistencia de un material, esto ya que la resistividad se relaciona a la forma de cómo están alojados los átomos de un material en una celda unitaria (Corona y Abarca, 2019, pp.18-19).

En la **¡Error! No se encuentra el origen de la referencia.** se presenta un diagrama de los p rincipios de transducción que pueden encontrar en los transductores.

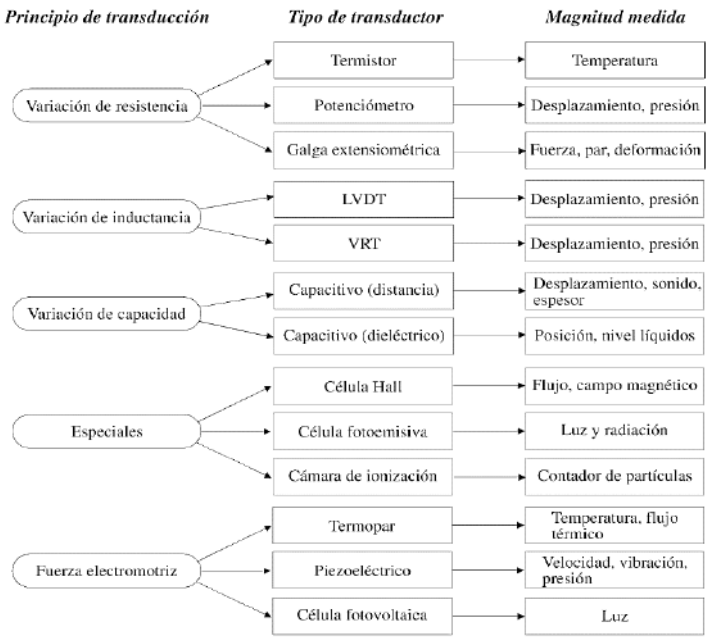


Ilustración 2-3: Diagrama de los principios de transducción.

Fuente: Granda y Elena 2015, pp.211.

2.4.1.2. Características estáticas del transductor

Las características estáticas de los sensores se presentan a continuación (Corona y Abarca, 2019, pp. 18-19).

- La sensitividad es el ingreso o entrada mínima requerida para provocar una salida que sea perceptible. La curva de salida se presenta para representar el cambio de la salida respecto a la entrada, de esta curva la pendiente se conoce como la sensitividad, en el caso de sensores con comportamientos no lineales se realiza una caracterización en el rango de funcionamiento del transductor.
- El rango es el intervalo entre el valor mínimo y máximo del parámetro físico que el transductor puede medir.
- La precisión se basa en la repetibilidad de una medida, es decir, la medida de un parámetro físico debe dar la misma salida en todo momento.
- La exactitud es la diferencia entre la salida vigente o actual y el valor verdadero o real del parámetro medido. Se expresa en porcentaje.
- La linealidad estática depende de factores ambientales, se relaciona con la desviación existente entre la curva del sensor proporcionada por el fabricante en condiciones controladas y la curva de salida actual. La no linealidad se representa en porcentaje, y este valor representa que tanto se separa de la curva ideal con relación al valor máximo a escala completa, como se muestra en la Ec.(17).

$$\%_{no\ linealidad} = \frac{desviación\ máxima}{Valor\ máximo\ a\ escala\ completa} \cdot 100 \quad (17)$$

- El offset es la separación en el eje y de la curva de salida, esta curva tendrá un igual comportamiento en ciertas condiciones, representa una salida del sensor cuando idealmente debería ser cero. Para mejorar esta variación en la curva se aplican correcciones en el eje y de la curva.
- La resolución es el cambio mínimo en el parámetro físico que le hace posible registrar al sensor.
- El error estático es causado por problemas en las lecturas, este error depende de la escala en la que se presenten los datos. Este error es mínimo cuando la curva tiende a ser lineal.

2.4.1.3. Características dinámicas del transductor

En cuanto a las características dinámicas del sensor se describen a continuación (Corona y Abarca, 2019, p. 50).

- El tiempo de respuesta es la duración de tiempo que pasa desde que el parámetro sensado tiene un cambio de estado y el transductor lo registra. El tiempo de reacción o respuesta depende de la magnitud que se mida y del transductor sensado.
- La histéresis es la posibilidad del transductor para copiar o seguir la curva ideal de salida debido a cambios en la variable física. La diferencia que tienen con la linealidad es que la presencia de histéresis se relaciona con el cruce de la salida en ambos sentidos a la curva de salida ideal.
- La linealidad dinámica se define como la capacidad del sensor en seguir de forma correcta la curva ideal o la dada por el fabricante del transductor en los casos donde el parámetro físico sufre cambios repentinos o muy rápidos, es decir, es el grado de desviación que tiene el transductor a la salida causado por cambios muy rápidos del fenómeno sensado.
- El error dinámico se puede generar por cargas inducidas por aparatos de medición, además dependerá de las características propias del sensor y de la forma en como se lo utiliza en el sistema.

2.4.2. Acondicionador de señal

Es un circuito que transforma las variables eléctricas de salida de los transductores en una señal eléctrica de fácil medición como: corriente, voltaje o frecuencia. La unión entre el transductor y

acondicionador es la interfase entre el parámetro físico y la entrada del circuito electrónico que procesara la información (Granda y Mediavilla, 2015, pp. 6-7). Un acondicionador se utiliza para:

- Obtener una magnitud que no es directamente una señal de tensión o intensidad.
- Amplificar la señal hasta un nivel superior al nivel de ruido eléctrico aleatorio.
- Filtrar la señal con el objetivo de eliminar el ruido generado por interferencias eléctricas.
- Situaciones donde la respuesta no sea lineal a los cambios de magnitud medida.

2.4.2.1. Tipos de acondicionamientos

Los acondicionamientos de señal se utilizan para convertir una señal de salida, normalmente se aplican los acondicionamientos para que el sistema de procesamiento pueda leer la señal de una forma más fácil. Se aplican distintos procesos de acondicionamiento los cuales se describen a continuación (Corona y Abarca, 2019, pp. 38-39).

- La amplificación tiene como objetivo aumentar la magnitud de una señal, por ejemplo, amplificar una señal de 5-10mV a una 0-5V.
- La linealización convierte una señal no lineal en una señal linealizada o de comportamiento lineal. En la Ilustración 2-4 se muestra un ejemplo de este tipo de acondicionamiento.
- El filtrado se aplica para separar componentes indeseados de la señal, existen varias configuraciones de filtros que van de acuerdo con los componentes que se desean eliminar o conservar. En la Ilustración 2-5 se representa el acondicionamiento de filtrado.
- En la conversión se aplican convertidores de conversión analógica-digital, de voltaje-frecuencia, frecuencia-voltaje, voltaje-corriente, corriente-voltaje, alterna-directa y directa-alterna. El caso más común es la de convertir una señal continua o analógica en una señal discreta o digital. En la Ilustración 2-6 se representa el acondicionamiento por conversión analógica-digital.
- En el aislamiento eléctrico se corta o interrumpe el paso de la señal entre la entrada y la salida, no existe un conductor físico entre la entrada y la salida. Por lo generar la entrada se transfiere a la salida por medio de una conversión óptica o magnética.
- La excitación en algunos casos los transductores la requieren para su funcionamiento, este es el caso de los transductores de reluctancia variable los cuales requieren de una corriente alterna.

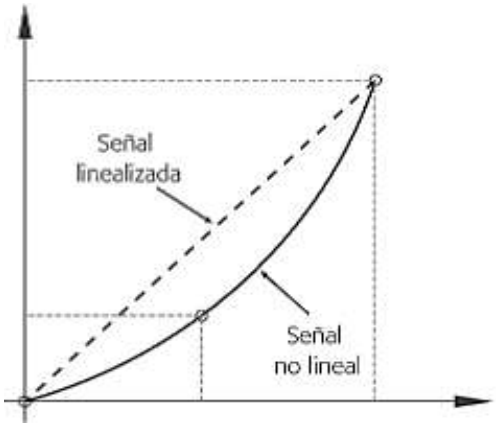


Ilustración 2-4: Acondicionamiento por linealización

Fuente: Corona y Abarca 2019, p.39.

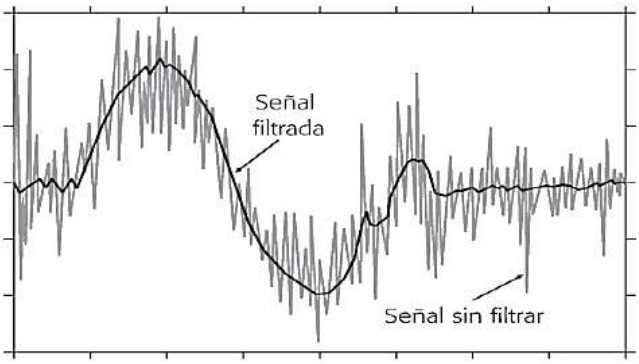


Ilustración 2-5: Acondicionamiento por filtrado.

Fuente: Corona y Abarca 2019, p.39.

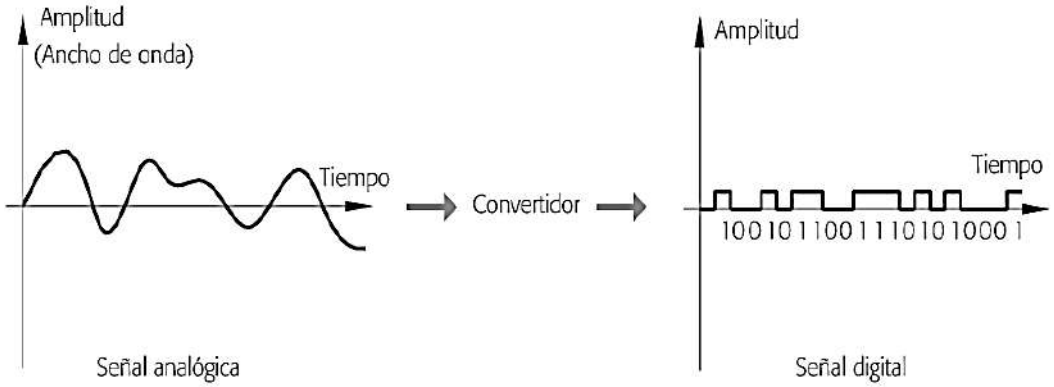


Ilustración 2-6: Acondicionamiento por conversión analógica-digital.

Fuente: Corona y Abarca 2019, p.39.

2.4.3. Procesamiento de la señal

Aquí se incluyen las conversiones a las que se debe someter la señal eléctrica para obtener la información que se busca. Se aplican operaciones lineales, no lineales, de composición de

múltiples señales o de procesamiento digital. Los códigos empleados en el hardware digital son: binario, hexadecimal, octal y BDC. Para el procesamiento de señal se utilizan algunas funciones digitales, para ello se crean chips encaminados a: almacenamiento, sincronización, aritmética, lógica, conversión de códigos, control, procesamiento, encaminamiento y conversión de naturaleza de señal. La electrónica actual requiere que se aplique una relación o puente con las señales análogas, para ello se emplean conversores análogos/digital (A/D) y digital/análogo, actualmente los conversores análogos/digital son los destinados para la instrumentación compleja en la actualidad.

2.4.3.1. Microcontrolador

Es un componente pequeño electrónico con la capacidad de ejecutar automáticamente instrucciones programadas de forma lógica. La característica principal de este dispositivo es que cuenta con todos los circuitos integrados en un solo chip como un sistema cerrado, a diferencia del microprocesador que presenta un sistema abierto con componentes separados del chip principal, es decir, requiere de circuitos externos como: memoria de datos, memoria de programa, reloj externo y dispositivos E/S. El microcontrolador requiere de un conjunto de instrucciones o algoritmo almacenado en una memoria en la que se decodifican y se ejecutan las operaciones. La programación se realiza con un lenguaje de programación (Dogan, 2006, p.2).

Según Moreno y Córcoles (2018), el microcontrolador puede estar conformado por distintos elementos, pero debe contener algunos elementos básicos para considerarse como tal:

- Las entradas y salida (E/S), se encargan de comunicar al microcontrolador con el exterior, estas entradas pueden ser de tipo digitales o de tipo analógicas. Cuenta con pines o conexiones de entrada para sensores, pines de salida para actuadores y pines de entrada/salida para conectar sensores y actuadores.
- La unidad central de procesamiento (CPU) es el componente de mayor importancia, en él se ejecutan las instrucciones del programa, es conocido también como procesador.
- En la memoria se almacena la información de instrucciones o datos a procesar. La CPU requiere de este elemento para acceder a la información y ejecutar las operaciones. Existen dos grupos de memorias: las volátiles y persistentes. Las memorias volátiles almacenan la información hasta que se corta la alimentación, aquí se encuentra la memoria RAM. Las memorias persistentes o no volátiles almacenan la información aun después de no recibir alimentación, en este grupo de memorias se encuentra el programa del microcontrolador.

En cuanto a las memorias no volátiles se encuentra la memoria ROM, EPROM, EEPROM, EEPROM Flash:

- La memoria de solo lectura (ROM) contiene el programa del microcontrolador que se incorpora durante el proceso de fabricación por lo tanto su contenido no puede ser alterado o modificado.
- La memoria de solo lectura programable y borrable (EPROM) es similar a la memoria ROM, pero con la particularidad de ser reprogramable.
- La memoria de solo lectura, programable y borrable eléctricamente (EEROM) son memorias reprogramables y puede almacenar datos de valores mínimos y máximos, datos de identificación, valores de configuración, etc. Normalmente son muy lentas.
- La memoria EEPROM Flash almacena el programa del usuario, su velocidad es alta y es reprogramable.

2.4.3.2. Lenguaje de programación

Existen diferentes lenguajes de programación, los más utilizados son: Basic, C, C++, Objetive-C, Java y Python. Para microcontroladores de Arduino se utiliza el lenguaje C, el cual se utiliza en un Entorno de Desarrollo Integrado (IDE). El lenguaje de programación permite al usuario utilizar comandos específicos destinados al control de un microcontrolador. Se utiliza un Entorno de Desarrollo Integrado (IDE) que abarca utilidades, herramientas y funciones para facilitar el desarrollo de software. El entorno de desarrollo integrado realiza actividades como: crear y modificar proyectos de software, ejecución del programa, implementación de código fuente, depuración del programa y compilación del código fuente. El Entorno de Desarrollo Integrado (IDE) está conformado por algunos componentes básicos para que funcione, estos componentes básicos se describen a continuación (Moreno y Córcoles, 2018, pp.98-99).

- El Editor de texto, es una herramienta útil para escribir el código del programa.
- El Compilador, permite la traducción del código fuente al lenguaje del microcontrolador.
- El depurador, permite experimentar y borrar errores en el código fuente.
- El editor gráfico, posibilita diseñar y crear interfaces gráficas para la interacción del usuario con la aplicación.
- El intérprete, tiene la capacidad de ejecutar el código fuente en el microcontrolador traduciéndolo a su código.

2.4.4. Presentación de la información

La final del procesamiento de la señal se da la información de resultado del proceso de medida, esta información debe ser comprensible al operador. El método más común para presentar la información es por medio de displays alfanuméricos y terminales alfanuméricos y gráficos relacionados a computadores. También puede ser indispensable registrar la señal o señales por medio de un almacenamiento temporal o permanente para su posterior análisis (Granda y Mediavilla, 2015, p. 7).

2.5. Sistemas electrónicos de medida

Un sistema electrónico de medida es simple o complejo. Instrumentos modernos están conformados por sistemas de instrumentación que aplican potencia de cálculo y flexibilidad de ejecución de ordenadores, software e instrumentos programables. La clasificación de los sistemas electrónicos de medida se representa en la Ilustración 2-7 (Granda y Mediavilla, 2015, pp. 8-13).

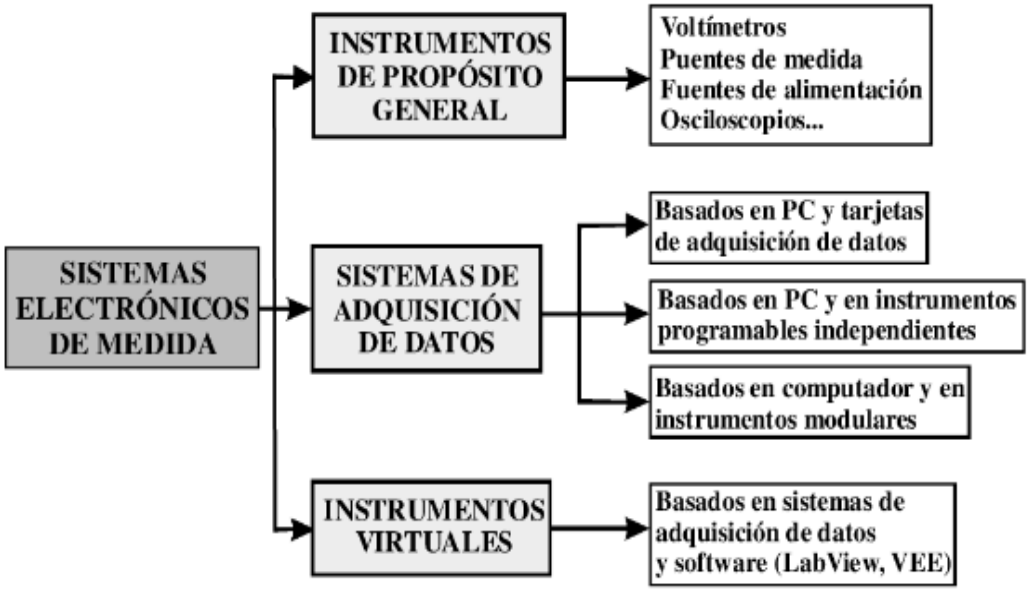


Ilustración 2-7: Clasificación de los sistemas electrónicos de medida.

Fuente: Granda y Elena 2015, p. 8.

2.5.1. Instrumentos de propósito general

Son aquellos instrumentos útiles en una medida, son fabricados independientemente con ciertas características y capacidades de medición dadas por el fabricante. Dentro del instrumento está incorporado el circuito electrónico que procesa la señal a analizar. Algunos ejemplos son los voltímetros, amperímetros, medidores de resistencia, capacidades, inductancias, osciloscopios, etc.

2.5.2. Sistemas de adquisición de datos

Son sistemas electrónicos de medida destinados a adquirir de forma automática la información de un proceso determinado o del estado de una planta (Gunsha, 2018, p.5). Aquellos sistemas basados en un computador son confiables debido a su coste, versatilidad, capacidad de cálculo, almacenamiento y visualización automática de medidas; se recurre a estos sistemas para obtener una gran cantidad de datos, situaciones que conllevan un largo tiempo de registro o cuando se desea una gran objetividad y fiabilidad. Se tiene tres tipos de configuraciones.

- Apoyados en un computador con una o varias tarjetas para obtención de datos.
- Apoyados en instrumentos autónomos de adquisición e independientes conectados a un computador.
- Apoyados en la instrumentación modular.

En algunos casos existe el instrumento en tarjeta que actúa como instrumento autónomo donde la tarjeta contiene la funcionalidad del instrumento. Para medidas en un entorno industrial se implementan métodos de monitorización remota a través de una red de comunicación y mediciones de gran precisión y fiabilidad se aplican sistemas de adquisición de datos basados en instrumentos independientes.

2.5.2.1. Protocolos de comunicación

El Serial Peripheral Interface (SPI) es un estándar de comunicación serie que transmite de forma síncrona mediante cuatro líneas de manera simultánea en los dos sentidos. Las líneas se identifican con las siglas descritas a continuación (Moreno y Córcoles, 2018, pp.53-54).

- SCK: envía la señal del reloj desde el maestro a todos los dispositivos esclavos.
- SS: sincroniza a los dispositivos esclavos y el número de líneas dependerá de la cantidad dispositivos conectados al maestro.
- MOSI: se utiliza para envió datos al esclavo.
- MISO: se utiliza para enviar datos del esclavo al maestro.

En la Ilustración 2-9 se observan las señales generadas en las líneas o puertos de la comunicación SPI, se identifica la sincronización en el envío y recepción de datos del maestro-esclavo por medio de la señal base generada por el reloj del maestro.

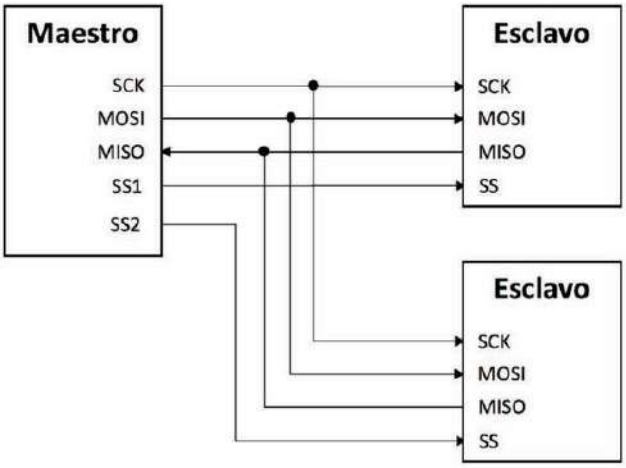


Ilustración 2-8: Conexión de maestro-esclavo en comunicación SPI.

Fuente: Moreno y Córcoles, 2018, p.54.

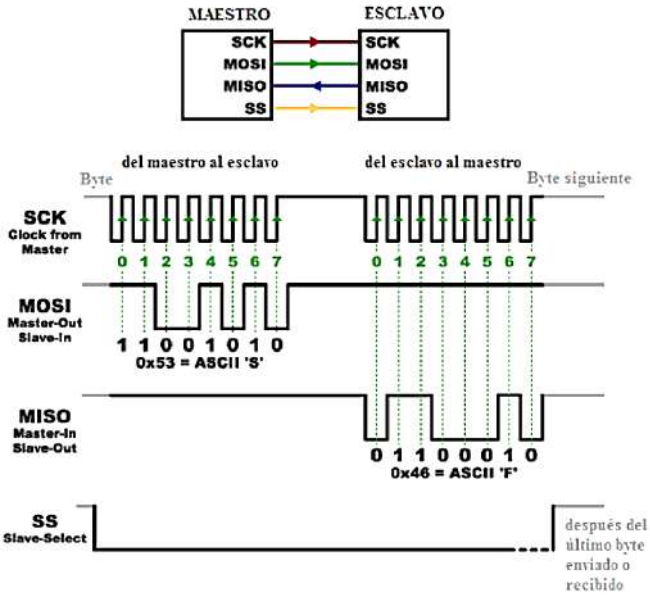


Ilustración 2-9: Señales del protocolo de comunicación SPI.

Fuente: Pérez y González, 2016, p.11.

El Universal Asynchronous Receiver/Transmitter (UART) es un protocolo serial utilizado para la comunicación entre varios procesadores para la trasformación de datos y transmitir a una determinada velocidad de forma asíncrona. Todas las placas de Arduino disponen de puerto UART ubicado en los pines Rx y Tx de las entradas analógicas 0 y 1 respectivamente. Se utiliza una línea para receptar y otra para el envío de datos. Se utiliza un bit en nivel bajo para referencia de inicio de la transmisión, en general se ocupan 8 bits para transmitir la información y para la parada o fin de trama se utiliza un bit en nivel alto, tal como se muestra en la Ilustración 2-10 (Pérez y González, 2016, pp.8-9).

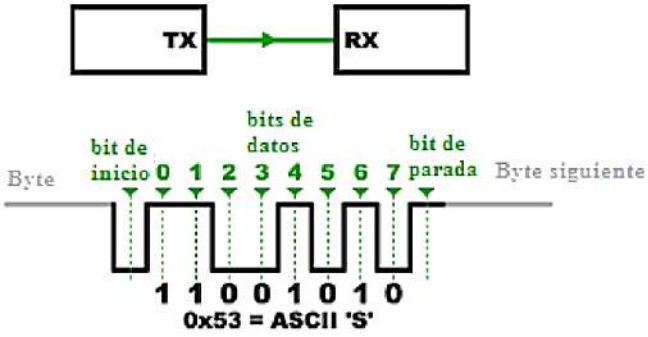


Ilustración 2-10: Señal de protocolo de comunicación UART.

Fuente: Pérez y González, 2016, p.9.

Para la recepción de datos se utiliza Rx y para el envió de datos se utiliza Tx. Para establecer la comunicación se debe tener para Rx y Tx los mismos parámetros de cantidad de bits del dato, cantidad de bits de paro, control de paridad, velocidad de transmisión y niveles lógicos utilizados. Para conectar los módulos a comunicar se debe conectar TX con RX (Herrera, Ilizaliturri y Morales 2006, p.4)

El Inter-Integrated Circuit (I2C) es un bus de datos destinado a la comunicación entre uno o varios dispositivos maestros y uno o varios dispositivos esclavos. La comunicación se realiza por medio de dos cables, necesita de un controlador definido como maestro. Se utiliza dos líneas, la una línea para reloj (SCL) y la otra para datos (SDA), las dos líneas necesitan de una conexión a voltaje CC (V) con una resistencia pull-up seleccionada de acuerdo con la capacitancia en las líneas. Los datos se transfieren cuando el bus está inactivo es decir cuando las líneas están en nivel alto después de una condición de parada (Valez y Becker, 2015, p.3).

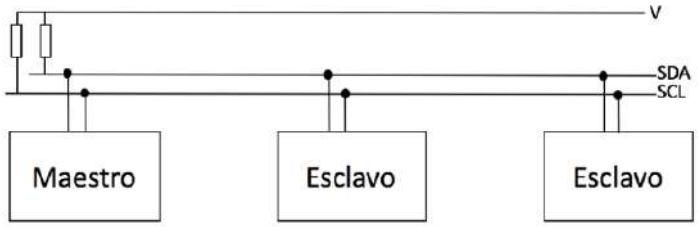


Ilustración 2-11: Conexión de maestro y esclavos para la comunicación I2C**.**

Fuente: Moreno y Córcoles, 2018, p.52.

2.5.3. Instrumentos virtuales

Es un entorno de programación gráfico para control y simulación de algún instrumento remoto o local. Permiten realizar medidas como si fueran de un instrumento real, actúa formando y recogiendo señales como lo haría su homólogo físico. Se utilizan programas como LabVIEW, VEE y MATLAB para crear instrumentos virtuales con el fin de simplificar y reducir el tiempo de desarrollo de aplicaciones (Granda y Mediavilla 2015, pp.14-15).

2.6. Sistema Global de Navegación por Satélite (GNSS)

Este sistema comprende a los Sistemas de Navegación por Satélite que brindan posicionamiento geoespacial y ayudan en la navegación y tiempo con cobertura global, que se logra por medio de constelaciones nominales de entre 27 a 30 satélites. El GNSS engloba los sistemas: GPS, GLONNAS, BeiDou, GALILEO, QZSS, IRNSS, entre otros sistemas de aumentación.

Todos los sistemas que forman parte del GNSS están conformados por tres componentes, los cuales son (Berné, Garrido y Capilla, 2019, pp.105-108):

- Segmento espacial: conformado por satélites en órbita, estos satélites se encargan de transmitir señales de medición en diferentes bandas de frecuencia.
- Segmento de control: compuesto por varios centros de control y antenas para la comunicación con los satélites en orbitan para GNSS. Se encarga de mantener en correcto funcionamiento del sistema y envía mensajes y datos de navegación.
- Segmento de usuario: conformado por los usuarios que tienen equipos de recepción de señales en tierra, mar o aire para posicionamiento o navegación.

2.6.1. Protocolo NMEA 0183

Es un estándar que permite la transmisión de datos entre instrumentos marítimos, equipos de comunicación y equipos de navegación. Es utilizado en la conexión de dispositivos con GNSS y es compatible con la totalidad de los dispositivos en el mercado. Los datos que permite intercambiar son básicos, entre estos esta la velocidad, posición, orientación, tiempo, altitud, constelación de satélites y separación del geoide.
En la transmisión de datos se hace uso de tres sentencias NMEA:

- Sentencias de envío de datos (Talker Sentences)
- Sentencias de origen de equipo (Proprietary Sentences)
- Sentencias de consulta (Query Sentences)

Para realizar la transmisión de datos se hace uso de las sentencias con caracteres ASCII delimitados por comas. Las comas tienen la función de marcas que ayudan a identificar el tipo de información a transmitir, además, ayudan a generar la estructura de la sentencia NMEA (Berné, Garrido y Capilla, 2019, pp. 502-503).

Tabla 2-1: Mensajes NMEA

Mensaje	Descripción
GGA	Tiempo, posición y datos de tipo fijo
GLL	Latitud, longitud, hora UTC de fijación de posición y estado
GSA	Modo de funcionamiento del receptor GPS, satélites, utilizados en la solución de posición y valores DOP
GSV	Número de satélites GPS a la vista, valores de elevación, azimut, valores SNR
MSS	Relación señal-ruido, intensidad de la señal, frecuencia y tasa de bits de un receptor de radio-beacon.
RMC	Tiempo, fecha, posición, rumbo y velocidad
VTG	Información de rumbo y velocidad relativa al suelo
ZDA	Mensaje de temporización PPS (sincronización con PPS)

Fuente: SiRF Technology, 2007.

Realizado por: González A., y Ortiz W., 2023.

En la Tabla se muestran las diferentes tramas NMEA que se pueden recibir un módulo receptor GNSS.

Tabla 2-2: Caracteres utilizados para la estructura de sentencias NMEA.

Caracteres	HEX	Descripción
"$"	24	Inicia la sentencia
Aaccc		Campo de dirección. "aa" es el identificador del hablante. "ccc" identifica el tipo de sentencia.
","	2C	Delimitador de campo
C-c		Bloque de oraciones de datos
"*"	2A	Delimitador de suma de comprobación
Hh		Campo de suma de comprobación
<CR><LF>	0D0A	Final de oración

Fuente: SkyTraQ, 2021.

Realizado por: González A y Ortiz W., 2023

Para la escritura de la sentencia se inicia con el carácter "$", luego "aa" es el identificador hablante seguido del identificador del tipo de sentencia "ccc", se separa con un delimitador de campo, se ingresa en el bloque de datos los datos a transmitir separados por el delimitador de campo, luego se agrega el delimitador de suma de comprobación, se añade el campo de suma de comprobación y por último está el final de la oración, como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

2.7. Almacenamiento

El almacenamiento de datos es un método para guardar información en forma de imágenes, texto, video, etc., de tal forma que esta información se mantenga a largo plazo. Para poder realizar el

almacenamiento se requiere de dispositivos con una memoria no volátil como discos duros, disco compacto o CD, DVD, tarjetas de memoria flash, etc. (Ramos, Ramos y Viñas, 2022, pp.90-101).

2.7.1. Memoria Secure Digital (SD)

Es una memoria flash portátil de tipo EEPROM de tipo no volátil que almacena la información aun cuando no esté alimentada o en funcionamiento. Su peso y tamaño es reducido y su capacidad de almacenamiento puede llegar a ser muy elevada. Esta memoria posee un formato de menor tamaño denominado micro SD útil en dispositivos reducidos como reproductores de audio, teléfonos móviles, etc. Para la comunicación se utiliza el protocolo SPI para la transferencia de datos en el almacenamiento.

Ilustración 2-12: Memoria SD y micro SD.

Fuente: Kingston, 2020.

CAPÍTULO III

3. MARCO METODOLÓGICO

3.1. Diagrama de etapas del proyecto

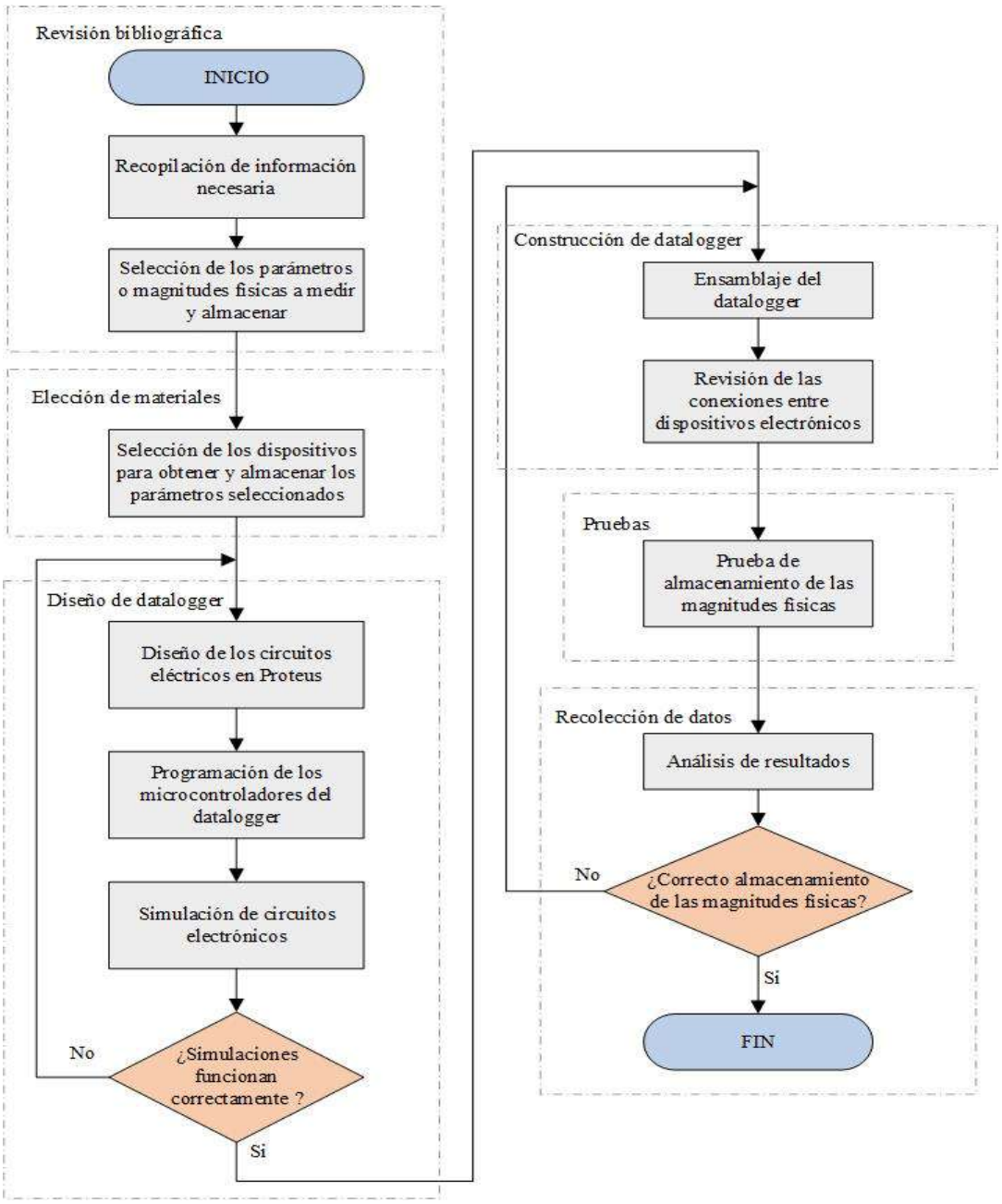


Ilustración 3-1: Diagrama de flujo de la construcción del Datalogger.

Realizado por: González A. y Ortiz W., 2023.

El diagrama de flujo presentado en la Ilustración , es una representación del proceso para la construcción del datalogger. El diagrama de flujo va en la dirección de las flecas, los rectángulos

representan los procesos a seguir y los rombos representan las decisiones a tomar. La recopilación de la información se realizó en el CAPITULO II; la selección de los parámetros físicos a medir se determinó por medio de los modelos matemáticos descritos en el CAPITULO II y se definieron estos parámetros en el CAPITULO III; la selección de los dispositivos para medir, obtener y almacenar las magnitudes físicas y datos se realizó de acuerdo a la selección de los parámetros, estos dispositivos se especificaron en el CAPITULO III. La simulación virtual de los circuitos electrónicos requirió del diseño de los circuitos y de algoritmos para su funcionamiento dentro de la simulación, estos procesos se describieron en el CAPITULO III. Al final de cada simulación se verificó que los resultados sean los correctos, en el caso donde los resultados no fueron los correctos se regresó al punto del diseño de los circuitos electrónicos, y en el caso donde los resultados fueron los correctos se prosiguió con el proceso de la construcción. Para la construcción se tomaron todos los circuitos eléctricos diseñados, la unión de los algoritmos en uno solo, los dispositivos seleccionados y la estructura para el montaje de los dispositivos eléctricos y electrónicos; además, se realizó una revisión de las conexiones entre dispositivos. El análisis de los resultados se realizó en base al funcionamiento del dispositivo construido junto con los sensores conectados, esta parte se especificó en el CAPITULO IV. Finalmente, una vez comprobado el funcionamiento se finalizó el proyecto, si el funcionamiento no era el correcto se retornó al proceso de construcción.

3.2. Selección de los parámetros o magnitudes físicas a medir u obtener y almacenar

De acuerdo, a los modelos de consumo de energía en vehículos eléctricos descritos en el anterior capítulo, mismos que se expresan en las ecuaciones Ec.(5), Ec.(10) y Ec.(12); se eligió cuatro parámetros o variables físicas: ángulo de la pendiente, densidad del aire, velocidad del viento y velocidad del vehículo, para las mediciones. Tomando como referencia la Ec.(15) relacionada con la densidad del aire se eligió la presión y temperatura ambiente como los parámetros a medir por medio de sensores. Mediante la Ec.(11) vinculada al SOC de la batería y la Ec.(16) asociada a la potencia eléctrica, se determinó los parámetros a medir: voltaje y corriente de la batería.

El anemómetro fue utilizado para medir la velocidad del aire y dirección del viento. Adicionalmente se obtuvo datos referentes al posicionamiento del vehículo, por medio de la latitud, longitud y altitud; y un dato referente a la identificación de puntos de interés. La altitud es indispensable para obtener el ángulo de la pendiente de la vía. En la Tabla se especifican los parámetros a obtener por medio de sensores y el módulo receptor GNSS.

Tabla 3-1: Parámetros seleccionados para la medición y almacenamiento

Parámetro	Unidad de medida/Formato	Dispositivo de medición
Presión ambiente	hPa	Sensor o transductor
Temperatura ambiente	°C	Sensor o transductor
Velocidad del viento	m/s	Sensor o transductor
Dirección del viento	Grados sexagesimales (°)	Sensor o transductor
Corriente de batería	Amperios (A)	Sensor o transductor
Voltaje de batería	Voltios (V)	Sensor o transductor
Latitud	Formato: ddmm.mmmmmmm / Grados Decimales (DD)	Módulo GNSS
Longitud	Formato: ddmm.mmmmmmm / Grados Decimales (DD)	Módulo GNSS
Altura	m	Sensor o transductor
Velocidad del vehículo	m/s	Módulo GNSS
Número de puntos de interés	Adimensional	Pulsador

Realizado por: González A y Ortiz W., 2023.

3.3. Selección de los dispositivos eléctricos para obtener y almacenar las medidas

El sistema de medición y almacenamiento o datalogger está conformado por dos placas electrónicas: una placa principal y otra para un proceso específico. El sistema esta subdividido en dos partes: la primera parte se encarga del procesamiento principal de sensores, módulos, almacenamiento y visualización con una placa electrónica principal y la segunda parte está destinada únicamente para procesar las señales del anemómetro con una placa electrónica.

De acuerdo a la Tabla se requirieron seis sensores, un módulo GNSS y un pulsador. Los parámetros seleccionados fueron escogidos de acuerdo a: los modelos de consumo de energía, la magnitud física a medir, información útil relacionada al funcionamiento, rango de medición, grado de complejidad de medición en el vehículo, precisión y disponibilidad en el mercado de los dispositivos junto con el costo.

3.3.1. Sensor de presión ambiente

El sensor de presión ambiente se seleccionó en base al tipo de medición a obtener, se distinguió dos tipos de sensores de presión: sensor de presión relativa y sensor de presión barométrica. Según la Ec.(15), la presión a medir es la del ambiente es decir una presión absoluta para ello se seleccionó un sensor que mida presión absoluta. Según el Ministerio de Turismo (2021, p.1) uno de los puntos más altos del ecuador se encuentra a 4800msnm en las faldas del monte Chimborazo, a esta altura se tiene aproximadamente 0.55atm que corresponde a 557.29hPa; para la presión más

alta a medir se tomó como referencia el nivel del mar que corresponde a 1 atm que equivale a 1013.25hPa (Sánchez, 2016, p.1). Con el rango de la presión a medir, entre 557.29 y 1013.25hPa, se seleccionó el sensor BMP280. Las especificaciones del sensor se presentan en el ANEXO A. Una característica importante de este sensor se relaciona con la información respecto a la presión absoluta, que la envía de forma directa por medio de una interface de comunicación I2C o SPI. Este sensor funciona bajo el principio de transducción piezorresistivo. Funciona en un rango de temperatura de entre -40° y 85°C y el rango de medición de la presión absoluta barométrica está entre 300 y 1100hPa. Adicional a la presión se puede determinar la altitud o altura con respecto a un punto específico, es decir se lo puede utilizar como altímetro.

Ilustración 3-2: Sensor de presión BMP280.

Fuente: MOUSER, 2023.

3.3.2. Sensor de temperatura ambiente

Se escogió el sensor de temperatura de acuerdo a las condiciones ambientales en el Ecuador, se tomó en cuenta la temperatura más baja y alta en el país como referencia para la selección del sensor de temperatura. Según la publicación de INAMHI (2012. pp.1-19) en el Ecuador se registró una temperatura mínima de -1.4°C y máxima de 35.9°C. En base a estos datos, se seleccionó el sensor LM35A que brinda un amplio rango de medición que se encuentra entre -55 a 150°C. Este transductor es de tipo resistivo, el voltaje de salida es linealmente proporcional a la temperatura en °C. En el ANEXO B se presentan las especificaciones de este sensor.

Ilustración 3-3: Sensor de temperatura LM35A.

Fuente: GAIMC, 2023.

3.3.3. Anemómetro y sensor de dirección de viento

El transductor para medir la velocidad del viento se seleccionó en base a las condiciones de velocidad permitida en carretera y velocidad del viento. Para vehículos livianos se estableció como velocidad máxima, en rectas de carreteras, 100km/h, es decir, 27.78m/s (ANT, 2021, p.89). En base al valor máximo de velocidad permitida en carreta se determina que se necesita de un sensor de velocidad de viento o anemómetro que permita medir más de 27.78m/s. De acuerdo a (Buckley et al., 1976, p.2761), se determinó que la velocidad del viento en forma de brisas es de 3mph o 1.34m/s y la velocidad del viento en forma de ráfagas es de 16mph o 7.15m/s durante el movimiento del vehículo. Se seleccionó un anemómetro que permita medir velocidades mayores a la del vehículo, esto debido a que la velocidad del viento es mayor a la velocidad del vehículo. Al considerar la velocidad máxima del vehículo permitida en carretera y velocidad máxima del viento en forma de ráfagas se determinó medir una velocidad mayor a 34.93m/s.

El anemómetro se seleccionó de acuerdo a: rango de medición, la disponibilidad en el mercado, la existencia de una ficha técnica dada por el fabricante y que presente la característica de ser montado en ambientes externos. El anemómetro elegido fue el Vantage Pro2 6410, este sensor presenta la ventaja de tener, adicionalmente, un sensor de dirección del viento; el tipo de transducción para determinar la velocidad del viento es de tipo magnética bajo el principio de efecto Hall y para la dirección del viento se aplica una transducción potenciométrica. En el ANEXO C se presentan las características del anemómetro seleccionado.

Ilustración 3-4: Anemómetro Davis Vantage Pro 2 6410.
Fuente: DAVIS, 2023.

3.3.4. Sensor de corriente

En base a la información relacionada a la descarga máxima que se puede realizar en una batería de alto voltaje de un vehículo eléctrico, se encontró una corriente de descarga máxima cerca de 1000A para el caso del VW ID3, VW ID4 y Tesla Model 3 (EVShop, 2023, p.1). Otro aspecto que se consideró es la dirección del flujo de corriente, en el caso del vehículo eléctrico la batería se puede descargar y cargar, es por ello que el flujo de corriente es en dos sentidos, es decir, existe corriente de descarga y corriente de carga, debido a esto se optó por un transductor de corriente bidireccional.

Se eligió un transductor de corriente directa DC bidireccional DHAB S/137 que permite medir entre -1000 a 1000A, en el ANEXO D se presentan las especificaciones de este transductor.

Ilustración 3-5: Sensor de corriente DHAB S/137.
Fuente: LEM, 2023.

3.3.5. Sensor de voltaje

Se buscó información referente al voltaje máximo que una batería de alto voltaje para un vehículo eléctrico. De acuerdo a la información encontrada una batería puede llegar a tener una estructura cerca de 800VDC como es el caso de KIA EV6, Porsche Taycan, BYD TANG y Audi e-tron GT RS (Electric Vehicle Database, 2023, p.1-320).

De acuerdo a la información anterior se eligió el transductor de voltaje JXDA4U que permite realizar mediciones desde 30 a 1000VDC, se consideró este transductor con este rango de medición debido a que no se encontró un transductor específico para mediciones hasta los 800VDC. En el ANEXO E se presentan las especificaciones de este traductor de voltaje.

Ilustración 3-6: Sensor de voltaje DVC JXDA4U.

Fuente: LEM, 2022.

3.3.6. Modulo receptor GNSS

El módulo receptor GNSS se seleccionó en base a las características de precisión para la latitud, longitud y velocidad. De acuerdo a la información encontrada, el módulo que presenta una buena precisión es el NS-HP-GN2 que posee el receptor PX1122R, para el posicionamiento tiene una precisión de 1.5m y en la velocidad tiene una precisión de 0.05m/s. Este módulo receptor GNSS requiere de una comunicación UART. Las especificaciones del receptor GNSS se presentan en el ANEXO F. Se requirió de una antena para recibir la información, la antena se seleccionó en base a la ganancia y el ruido que puede aceptar el módulo GNSS, el módulo receptor PX1122R acepta una ganancia de hasta 40dB y un factor de ruido menor a 2dB, la antena seleccionada tiene 2dB con un factor de ruido menor a 1.5dB. En el ANEXO G se presenta las especificaciones de la antena.

Ilustración 3-7: Módulo receptor GNSS NS-HP-GN2

Fuente: NavSpark, 2023.

Ilustración 3-8: Antena de alta precisión para modulo receptor GNSS

Fuente: NavSpark, 2023.

3.3.7. Modulo micro SD

En cuanto al módulo para la micro SD se eligió uno propiamente para placas electrónicas Arduino, este módulo fue útil para guardar la información de las magnitudes físicas y datos, el módulo sirve como un medio en la escritura de datos. El módulo requiere una comunicación SPI para transmitir los datos a la tarjeta micro SD.

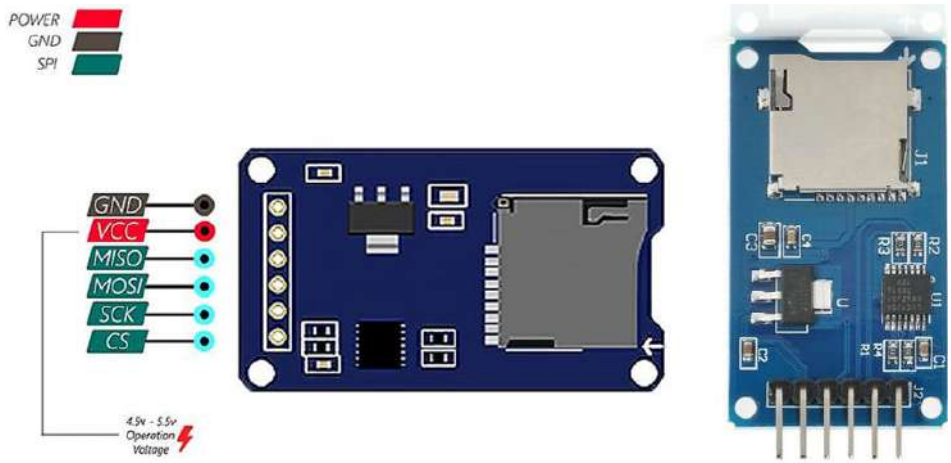


Ilustración 3-9: Módulo micro SD y pines de comunicación SPI y alimentación.

Fuente: DIGIZONE, 2023.

3.3.8. Placas electrónicas con microcontrolador

Se escogieron dos placas de microcontrolador en base a la disponibilidad dentro del mercado, la facilidad para adquirirla y el costo. Se eligió entre las placas de microcontrolador de la marca Arduino, para este trabajo se utilizó una placa de microcontrolador de procesamiento principal y una placa de microcontrolador para un subprocesamiento específico.

Para la placa de microcontrolador de procesamiento principal se necesitó principalmente de tres puertos de comunicación UART, un puerto para comunicación I2C y un puerto de comunicación SPI. En general la mayoría de las placas de microcontrolador Arduino presentas estos puertos de comunicación, pero hay una limitante en cuestión a la cantidad de los puertos UART, es por ello que se eligió la placa Arduino Mega 2560 la cual presenta cuatro puertos UART, un puerto I2C y un puerto SPI. En el ANEXO H se presentan las especificaciones de la placa Arduino Mega 2560.

Para la placa de microcontrolador para el subprocesamiento específico, se eligió un Arduino Pro Mini, la cual se encontraba entre la más económica, de esta placa solo se necesitó de un único puerto de comunicación UART por lo cual no existió la necesidad de utilizar una placa que disponga de más de un puerto de comunicación y de mayor costo. La placa Arduino Pro Mini se utilizó para procesar y enviar la información medida por el anemómetro tanto de la dirección como de la velocidad del viento. Las especificaciones de la placa Arduino Pro Mini se encuentran en el ANEXO I.

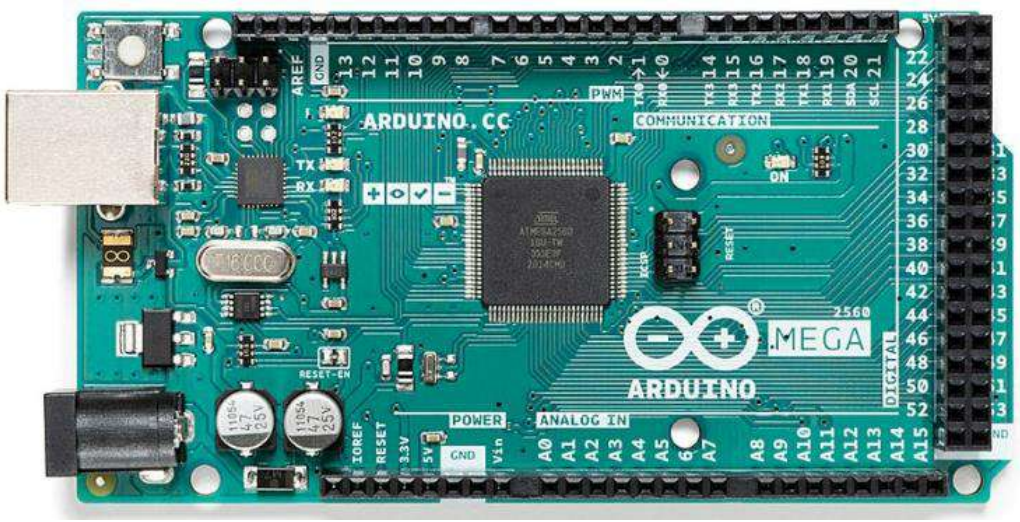


Ilustración 3-10: Arduino Mega 2560

Fuente: Arduino, 2023.

Ilustración 3-11: Arduino Pro Mini

Fuente: Indiamart, 2023.

3.3.9. Interfaz gráfica o pantalla

Para la interfaz gráfica se seleccionó una pantalla para visualizar todos los datos requeridos y que tenga la característica de ser táctil, esto último para seleccionar las opciones disponibles en el almacenamiento de los datos. Se determinó una pantalla de 4.3 pulgadas, de acuerdo a la disponibilidad en el mercado se seleccionó una pantalla de la marca NEXTION con un tamaño de 4.3 pulgadas, el modelo de la pantalla es NX4827K043_011R, y cuenta con la característica de ser táctil, la transmisión de datos es por protocolo UART. Las especificaciones de esta pantalla se presentan en el ANEXO J.

Ilustración 3-12: Pantalla NEXTION lNX4827K043_011R.

Fuente: NEXTION, 2022.

3.4. Diseño del datalogger

El datalogger está constituido principalmente por los elementos presentados en la **¡Error! No se encuentra el origen de la referencia.**. Cada dispositivo presenta una ficha técnica en la cual se especifican las características junto con la ecuación, ganancia o proceso de transmisión de datos.

Tabla 3-2: Componentes principales del datalogger

Componentes	Modelo	Referencia
Sensor de presión y altímetro	BMP280	(MOUSER, 2023)
Sensor de temperatura	LM35A	(GAIMC, 2023)
Sensor de velocidad de viento (anemómetro)	Vantage Pro2 6410	(DAVIS, 2023)
Sensor de corriente	DHAB S/137	(LEM 2016)
Sensor de voltaje	JXDA4U	(AliExpres2023)
Módulo GNSS NS-HP-GN2	PX1122R	(NavSpark 2023)
Pulsador para puntos de interés	ZB2 (NO)	(microlog, 2023)

Módulo Micro SD	Genérico	(DIGIZONE 2023)
Placa principal	Arduino Mega 2560 (ATmega2560)	(Arduino 2023)
Placa subcircuito	Arduino Pro Mini (ATmega328P)	(Indiamart 2023)
Pantalla Nextion	lNX4827K043_011R	(NEXTION 2023)

Realizado por: González A y Ortiz W., 2023.

En la **¡Error! No se encuentra el origen de la referencia.** se esquematiza las conexiones del d atalogger mediante sus pines y protocolos de comunicación, tanto para medición como almacenamiento de datos.

En el ANEXO K se presenta el diagrama de flujo del algoritmo para el microcontrolador ATMega 2560; en el ANEXO L se presentan un diagrama de flujo del algoritmo para el microcontrolador ATMega 328P para el anemómetro.

El circuito electrónico general del datalogger se dividió en seis circuitos, estos se describen en la **¡Error! No se encuentra el origen de la referencia.**, se realizó de esta forma con el fin de d escribir de mejor manera cada sensor y modulo que forman parte del datalogger.

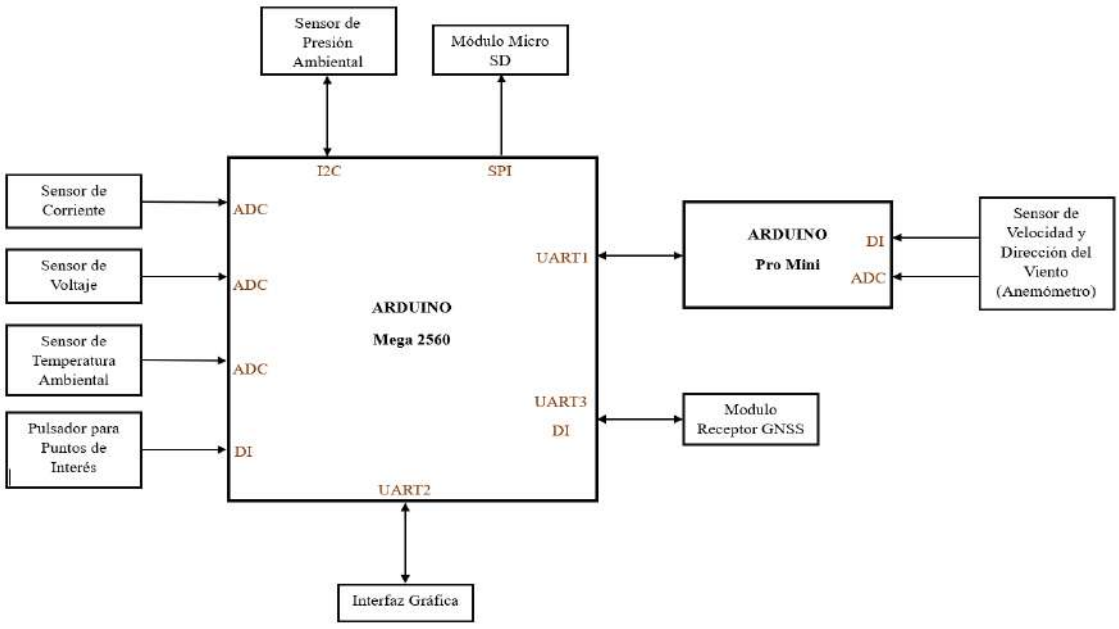


Ilustración 3-13: Diagrama de bloques del sistema datalogger.

Realizado por: González A. y Ortiz W., 2023.

Tabla 3-3: Circuitos principales para el datalogger

Nro.	Circuitos electrónicos principales Datalogger	Descripción
1	Circuito de sensores con la placa principal	Incluye sensor de presión, temperatura, corriente, voltaje y pulsador.
2	Circuito para el anemómetro	Incluye sensor de velocidad del viento y dirección del viento
3	Circuito para el módulo receptor GNSS	Destinado para modulo receptor GNSS
4	Circuito para la interfaz gráfica o pantalla	Destinado para pantalla Nextion
5	Circuito para el módulo micro SD	Destinado para el registro de los parámetros seleccionados
6	Circuito para el envío y recepción de datos	Destinado para la transmisión y recepción de datos de velocidad y dirección del viento
7	Circuito de alimentación y carga	Destinado para la alimentación de todo el sistema y carga de baterías

Realizado por: González A. y Ortiz W., 2023.

Para cada circuito se describió el algoritmo para el funcionamiento de los sensores y módulos que forman parte del datalogger. Se presentaron los resultados de la simulación virtual en cada uno de los circuitos para corroborar el funcionamiento del algoritmo para los microcontroladores. La simulación virtual se realizó con la ayuda del software Proteus, el mismo que se utiliza para el desarrollo de diagramas de los circuitos electrónicos, a excepción de la pantalla Nextion que se utilizó el software Nextion Editor para la configuración de la pantalla y su simulación virtual.

3.4.1. Diseño del procesamiento de señales directas de sensores con placa principal

3.4.1.1. Circuito de sensores con la placa principal

Los cálculos de las medidas de los sensores analógicos se realizaron utilizando la resolución del microcontrolador que es de 0.004888V/bit, este valor es útil para convertir el valor digital a un valor de voltaje. En esta parte del diseño del datalogger solo se representan los sensores conectados directamente a la placa principal de procesamiento o Arduino Mega 2560. En este circuito se encuentran: sensor de temperatura ambiental, sensor de presión ambiental, sensor de corriente y sensor de voltaje. Adicionalmente se añade la conexión del pulsador para identificar puntos de interés.

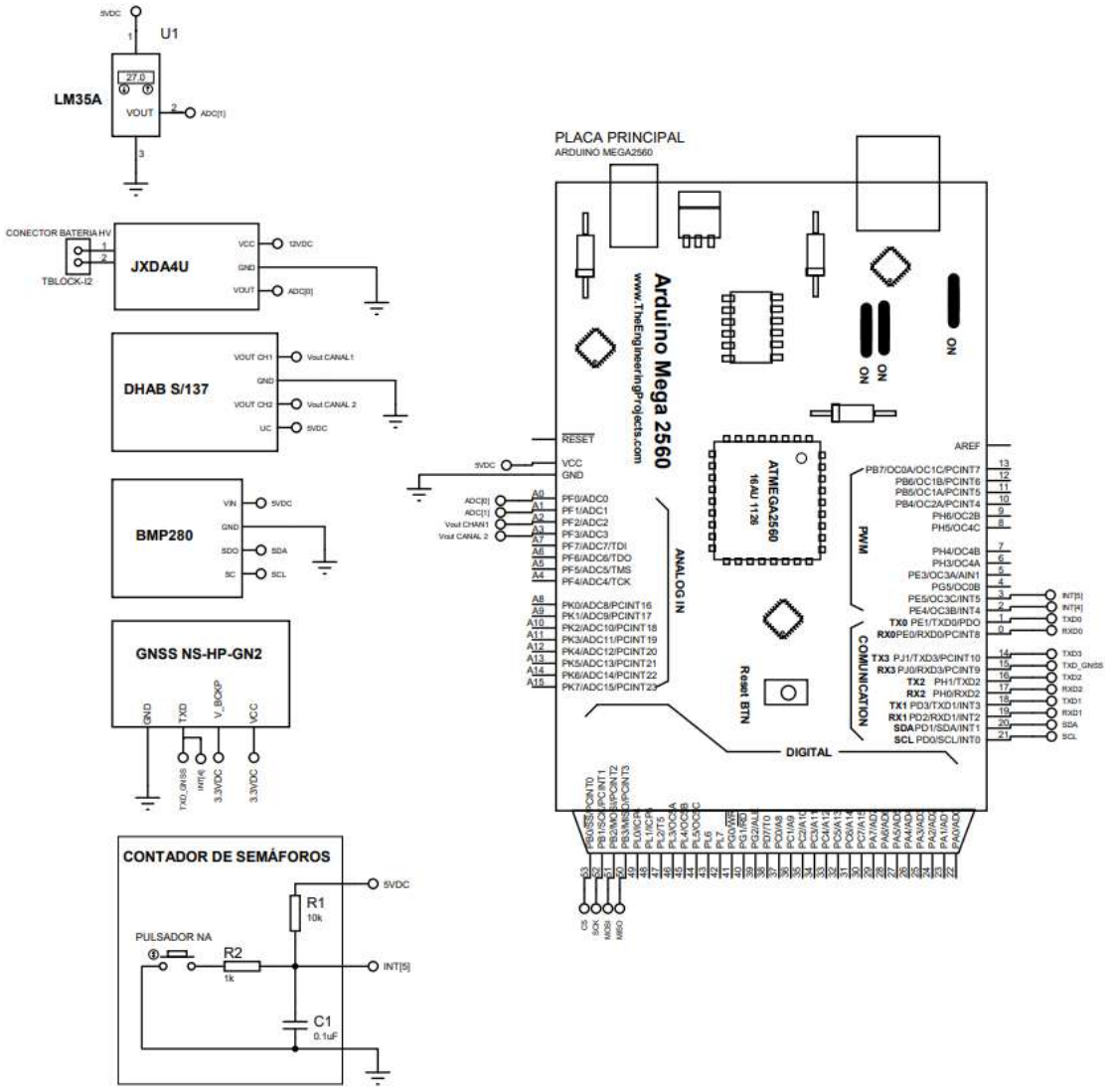


Ilustración 3-14: Circuito de sensores con Arduino Mega 2560.

Realizado por: González A. y Ortiz W., 2023.

En el circuito representado en la **¡Error! No se encuentra el origen de la referencia.** se observan l as conexiones de los sensores y de un pulsador. La señal de salida del sensor de temperatura ambiental, sensor de corriente y sensor de voltaje es de tipo analógica, es decir, la señal de salida varia su amplitud o voltaje de acuerdo a la variación de la magnitud física medida, el voltaje de la señal se encuentra entre 0 y 5VDC, las señales de voltaje ingresan a los pines que presentan un conversor análogo-digital (ADC) en la placa Arduino Mega 2560. Para el sensor de corriente se añade algunos capacitores y resistencias para protección y ruido recomendado por el fabricante, revisar ANEXO D y ANEXO M.

El módulo del sensor de presión BMP280 envía una señal de tipo digital, el módulo envía valores medidos de presión absoluta por medio de la interfaz de comunicación I2C. El módulo del sensor

de presión se conectó a los pines SDA y SCL del Arduino Mega 2560 que corresponden a la comunicación I2C.

Para identificar los puntos de interés en una ruta se utiliza un pulsador, el pulsador sirve para generar interrupciones que ingresan al Arduino por medio del pin D2. Para el funcionamiento del conteo de los semáforos con el algoritmo se necesitó que la entrada de la señal digital sin presionar el pulsador sea de 5VDC, para ello se utilizó una resistencia R2 de 10KΩ como Pull-Up, que permite la lectura de un estado en reposo, con esta resistencia el microcontrolador lee un estado de aproximadamente 5VDC. Se conectó un capacitor cerámico de 0.1uF en paralelo con el pulsador para ayudar a eliminar el revote en la señal al momento presionar el pulsador y con ello evitar falsas lecturas de tipo interrupción. La resistencia R1 se conectó en serie con el pulsador para proteger al condensador durante la descarga y para protegerlo durante la carga se utiliza la resistencia R2 de 10KΩ.

3.4.1.2. Descripción del algoritmo de la placa principal con sensores directos

En el algoritmo del sensor de temperatura se declararon dos variables: *LecA_LM35* y *Dato_LM35*; se definió el pin de entrada analógica A1 como *pinLM35*, todo esto antes del *void loop()* y *void setup()*. Una vez definidas las variables y pines, se realizó la lectura digital como se muestra en la Ec.(18); se obtiene la medida de temperatura en °C, para ello el valor calculado de la medida se asignó a la variable *Dato_LM35*, el cálculo de la medida se realizó con la Ec.(19).

$$LecA\_LM35 = analogRead(pinLM35) \quad (18)$$

$$Dato\_LM35 = \frac{LecA\_LM35 \cdot 5000.0}{1023.0} \cdot Ganancia_{temp} \quad (19)$$

Donde:

$LecA\_LM35$ = lectura digital de señal de entrada del sensor de temperatura.
$Dato\_LM35$ = cálculo de medida de temperatura en °C.
$Ganancia_{temp}$ = ganancia del sensor LM35A, 0.01V/°C.

En el algoritmo del sensor de corriente se declararon ocho variables: *LecA_S137CH1*, *Dato_S137CH1*, *G1_S137*, *LecA_S137CH2*, *Dato_S137CH2*, *G2_S137*, *Voff_S137* y *Dato_S137*; se definió el pin de entrada analógica A2 como *pinS137CH1* y el pin A3 como *pinS137CH2*. En el *void loop()* se realizó la lectura analógica, con la función *analogRead()*, para

los pines *pinS137CH1* y *pinS137CH2*, luego se asignó un valor de lectura digital como se muestra en la Ec.(20)**.** y Ec.(21), respectivamente. Luego se obtuvo el valor calculado de la corriente medida por medio de las lecturas de señal analógica, definidas para el canal uno y canal dos, el canal uno solo permite mediciones entre el rango de -75 a 75A y el canal dos se ocupa para la medición entre -1000 a -75A y de 75 a 1000A; las variables asignadas para el cálculo de la medición son *Dato_S137CH1* y *Dato_S137CH2*, para la medición del canal uno y canal dos, respectivamente. Para el cálculo de la medida se utilizó la fórmula dada por el fabricante del sensor de corriente, la fórmula se muestra en la Ec.(22) y Ec.(23) para el canal uno y canal dos respectivamente. Finalmente, una vez calculada la medida de corriente en cada canal se realizó una comparación en un rango de valores de entre -75 y 75A, si se determina que la medida se encuentra en este rango el valor de la corriente medida almacenado en *Dato_S137CH1* y se asigna a la variable *Dato_S137*, caso contrario *Dato_S137* toma el valor almacenado en *Dato_S137CH2*, como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

$$LecA\_S137CH1 = analogRead(pinS137CH1) \quad (20)$$

$$LecA\_S137CH2 = analogRead(pinS137CH2) \quad (21)$$

$$Dato\_S137CH1 = \frac{((LecA\_S137CH1 * 0.004888) - Voff\_S137)}{G1\_S137} \quad (22)$$

$$Dato\_S137CH2 = \frac{((LecA\_S137CH2 * 0.004888) - Voff\_S137)}{G2\_S137} \quad (23)$$

Donde:

$LecA\_S137CH1$ = lectura digital de señal de entrada del canal uno del sensor de corriente.
$LecA\_S137CH2$ = lectura digital de señal de entrada del canal dos del sensor de corriente.
$Dato\_S137CH1$ = cálculo de medida de corriente del canal uno en A.
$Dato\_S137CH2$ = cálculo de medida de corriente del canal dos en A.
$Voff\_S137$ = voltaje de compensación de 2.5V.
$G1\_S137$ = sensibilidad del canal uno de 26.67mv/A.
$G2\_S137$ = sensibilidad del canal uno de 2mv/A

```
...
if((Dato_S137CH1> -75) && (Dato_S137CH2< 75)){
  Dato_S137 = Dato_S137CH1;
}else{
  Dato_S137 = Dato_S137CH2;
}
...
```

Ilustración 3-15: Comparación y asignación del valor de corriente medida.

Realizado por: González A. y Ortiz W., 2023.

En el Algoritmo del sensor de voltaje al inicio se declararon dos variables: *LecA_ JXDA4U* y *Dato_ JXDA4U*; se define el pin A0 como *pin JXDA4UP*. Luego en el *void loop()* se realizó una lectura analógica del pin *pinJXDA4UP* como se muestra en la Ec.(24), el valor leído de forma digital se almacena en *LecA_ JXDA4U*. Finalmente se realizó el cálculo de la medida de voltaje con la Ec.(25).

$$LecA\_JXDA4U = analogRead(pinJXDA4U) \quad (24)$$

$$Dato\_JXDA4U = \frac{LecA\_JXDA4U \cdot 5.0}{1023.0} \cdot Ganancia_{volt} \quad (25)$$

Donde:

$Dato\_JXDA4U$ = medida de voltaje calculada en V.
$LecA\_JXDA4U$ = lectura digital de señal de entrada del sensor de voltaje.
$Ganancia_{volt}$ = ganancia del sensor JXDA4U, 200VP/VS.

Para el sensor de presión, un sensor con interfaz de comunicación I2C, primero se inicializaron las librerías para la comunicación I2C y para el sensor de presión BMP280; para la comunicación se inicializa la librería *Whire.h* y para el sensor BMP280 se inicializan las librerías *Adafruit_Sensor.h* y *Adafruit_BMP280.h*. Luego se creó un objeto *bmp* de tipo *Adafruit_BMP280*, junto con las variables a utilizadas en la obtención de los datos de presión y altura denominadas *Dato_BMP* y *Dato_Alt* respectivamente. Para obtener el dato de la presión, se utilizó la función *bmp.readPressure ()*, la medida se almacena en la variable *Dato_BMP* como se muestra en la Ec.(26). Para determinar la altura se utiliza la función *bmp.readAltitude()* como se muestra en la Ec.(27), se incluye el valor de presión con respecto al punto inicial de la medición, en este caso la medida se realiza con respecto al nivel del mar por lo que se incluye la presión a nivel del mar en hPa.

$$Dato\_BMP = bmp.readPressure()/100 \quad (26)$$

Donde:

Dato_BMP = medida de presión absoluta en hPa.

$$Dato\_Alt = bmp.readAltitude(1013.25) \quad (27)$$

Donde:

$Dato\_Alt$ = medida de altura en m.

Para el contador de semáforos en la señal de entrada hacia el Arduino se tiene aproximadamente 5VDC cuando no se presione el pulsador y aproximadamente 0VDC cuando se presione, de esta forma el programa para el conteo determinara el momento en que el voltaje baje a cerca de 0VDC y con ello se suma un valor de uno cada vez que se presione el pulsador para la identificación de los puntos de interés. Primero se definió el pin utilizado para la interrupción y la variable para determinar y almacenar la cantidad de interrupciones, como se muestra en la **¡Error! No se e ncuentra el origen de la referencia.**.

```
...
const byte Pin_Puntos = 3;
int Cont_Puntos;
...
```

Ilustración 3-16: Definición de pin y variable para los puntos de interés.

Realizado por: González A. y Ortiz W., 2023.

Luego se especificó el tipo de lectura digita y el tipo de interrupción, como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

```
pinMode(Pin_Puntos, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(Pin_Puntos), Puntos_Interes, FALLING);
```

Ilustración 3-17: Tipo de lectura e interrupción para el pin D3 de la placa Arduino Mega.

Realizado por: González A. y Ortiz W., 2023.

Se creó una función denominada *void Puntos_interes()*, en la cual se genera el conteo de puntos de interés por medio de un contador al que se le suma una unidad al momento de detectar una interrupción el pin D3, como se muestra en la **¡Error! No se encuentra el origen de la r eferencia.**.

```
...
void Puntos_Interes() {
  Cont_Puntos = Cont_Puntos + 1;
}
...
```

Ilustración 3-18: Función creada para el conteo de puntos de interés

Realizado por: González A. y Ortiz W., 2023.

3.4.1.3. Simulación de sensores virtuales para la comprobación del algoritmo

Para la simulación fue necesario cargar el algoritmo en el software Proteus en la placa de microcontrolador del circuito. Los resultados de la simulación del sensor de temperatura se representan en la **¡Error! No se encuentra el origen de la referencia.**.

Ilustración 3-19: Resultados de simulación de sensor de temperatura LM35.

Realizado por: González A. y Ortiz W., 2023.

Los resultados de la simulación del sensor de corriente se muestran en la **¡Error! No se encuentra el origen de la referencia.**.

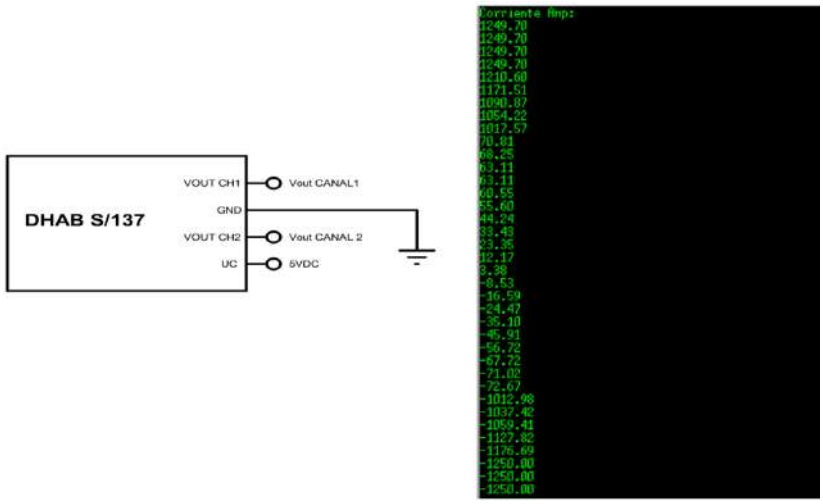


Ilustración 3-20: Resultados de simulación de sensor de corriente DHAB S/137.

Realizado por: González A. y Ortiz W., 2023.

Los resultados de la simulación del sensor de voltaje se presentan en la **¡Error! No se encuentra e l origen de la referencia.**.

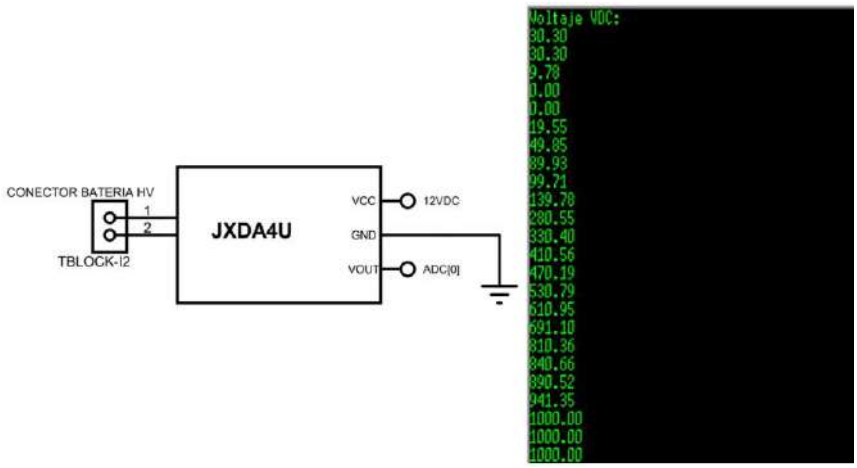


Ilustración 3-21: Resultados de simulación de sensor de voltaje JXDA4U.

Realizado por: González A. y Ortiz W., 2023.

Los resultados del sensor de presión se presentan en la **¡Error! No se encuentra el origen de la r eferencia.**.

Ilustración 3-22: Resultados de simulación de sensor de presión absoluta BMP280**.**

Realizado por: González A. y Ortiz W., 2023.

3.4.2. Diseño para la medición de velocidad y dirección del viento

3.4.2.1. Circuito para el anemómetro

Para el anemómetro se tiene una placa Arduino Pro Mini como placa para un subcircuito destinado al procesamiento las señales del anemómetro. Se procesan dos señales, una analógica que corresponde al sensor de dirección del viento y una periódica o de frecuencia correspondiente al sensor de velocidad del viento. El anemómetro tiene su propia placa debido a que el algoritmo del sensor funciona por medio de interrupciones lo cual hace que exista problemas en el registro y medición de los demás datos en el caso de conectarlo a la placa principal de forma directa.

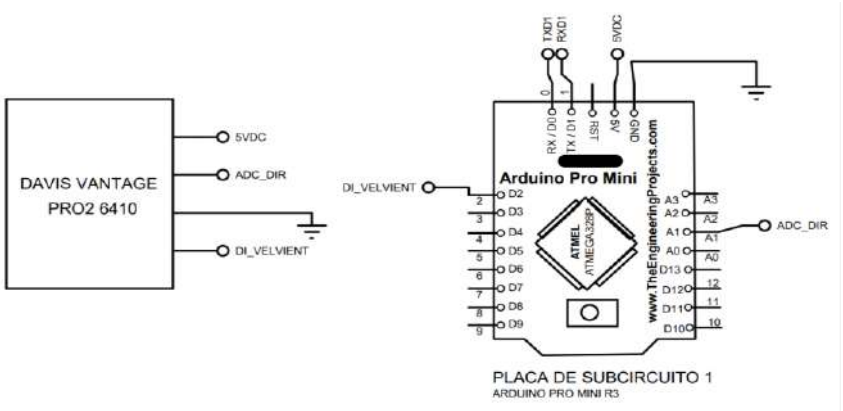


Ilustración 3-23: Circuito del anemómetro.

Realizado por: González A. y Ortiz W., 2023.

La señal del sensor de dirección del viento ingresa a la placa electrónica por medio del pin A1, y la señal de la velocidad del viento ingresa por el pin D2.

3.4.2.2. Descripción del algoritmo para el anemómetro

En el algoritmo primero se inicializó la librería *math.h* como se observa en el ANEXO L, se definieron los pines para la lectura de las señales del anemómetro como se muestra en la **¡Error! No se encuentra el origen de la referencia.**. Se utilizó el pin D2 para leer la señal del sensor de viento y el pin A1 para la señal de la dirección del viento.

```
...
#define PinVel 2
#define PinDir A1
...
```

Ilustración 3-24: Pines de señal de entrada para el anemómetro.

Realizado por: González A. y Ortiz W., 2023.

Se declararon variables tipo: *int, float* y *volatile unsigned long*; como se muestra en la **¡Error! No se encuentra el origen de la referencia.**. En la **¡Error! No se encuentra el origen de la referencia.** se describe la función de cada variable dentro del algoritmo.

```
...
int LecA_Direc;
float Dato1_Direc;
float Dato1_Velmps;
volatile unsigned long Tiempo_Rebote;
volatile unsigned long Cont_Int;
volatile unsigned long Pulsos_Seg;
...
```

Ilustración 3-25: Variables declaradas en el algoritmo del anemómetro

Realizado por: González A. y Ortiz W., 2023.

Tabla 3-4: Descripción de las variables pare el algoritmo del anemómetro

Variable declarada	Función
Tipo int	
LecA_Direc	Lectura digital de la señal de entrada del sensor de dirección
Tipo float	
Dato1_Direc	Cálculo de dirección del viento
Dato1_Velmps	Cálculo de la velocidad del viento
Tipo volatile unsigned long	
Pulsos_Seg	Variable utilizada para almacenar el conteo de interrupciones
Cont_Int	Variable como contador por medio de interrupciones
Tiempo_Rebote	Variable utilizada para evitar falsas lecturas, tipo interrupción, en el sensor de velocidad del viento

Realizado por: González A. y Ortiz W., 2023.

Se inicializó la función *attachInterrupt()* para las interrupciones, en la cual se definió: pin de lectura de interrupción, la función para la interrupción y modo de interrupción; como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

```
...
attachInterrupt(digitalPinToInterrupt(PinVel), Rotaciones_Int, FALLING);
...
```

Ilustración 3-26: Procesos generados en el *void setup()* para el algoritmo del anemómetro

Realizado por: González A. y Ortiz W., 2023.

Se consideró el conteo de las interrupciones cada segundo, para ello se necesitó de un temporizador que encere el contador segundo a segundo, en la ilustración se muestra la inicialización de este temporizador.

```
...
TCCR1A = 0;
TCCR1B = 0;
TCCR1B |= (1<<CS10)|(1 << CS12);
TCNT1 = 0xC2F8;
TIMSK1 |= (1 << TOIE1);
...
```

Ilustración 3-27: Inicialización de temporizador para anemómetro.

Realizado por: González A. y Ortiz W., 2023.

Para determinar la dirección del viento se realiza una lectura digital en el pin A1 denominado *LecA_Direc* como se muestra en la Ec. (28). En la Ec. (29) se presenta la ecuación para determinar la dirección del viento.

$$LecA\_Direc = analogRead(PinDir) \quad (28)$$

$$Dato1\_Direc = \frac{LecA\_Direc \cdot 5.0}{1023.0} \cdot Ganancia_{Dir} \quad (29)$$

Donde:

$Ganancia_{Dir}$ = ganancia del sensor de dirección del viento, 72°/V.

La lectura digital de $PinDir$ y el cálculo de la dirección se generan en una función denominada ANEM_*DIR()*, la cual se ejecuta cada segundo.

Para el cálculo de la velocidad del viento se creó la función *Rotaciones_Int()* que funciona junto a un temporizador, en esta función se realiza el conteo de interrupciones por medio de la variable

Cont_Int, adicionalmente se añadió un comparador con la función *millis()* para evitar la lectura de falsas interrupciones, como se muestra la **¡Error! No se encuentra el origen de la referencia.**.

```
void Rotaciones_Int() {
  if((millis() - Tiempo_Rebote) > 15 )
    Cont_Int++;;
    Tiempo_Rebote = millis();
  }
}
```

Ilustración 3-28: Conteo de interrupciones del sensor de velocidad del viento.

Realizado por: González A. y Ortiz W., 2023.

Para determinar el tiempo de duración del conteo de las interrupciones, se utilizó un temporizador (TIMER1). El temporizador se configuró para un segundo, cada segundo se accede a la función ISR*(TIMER1_OVF_vect)* en el cual se almacena el valor de *Cont_Int* en la variable *Pulsos_Seg* y luego se encera *Cont_Int* para reiniciar el conteo desde cero, como se muestra en la **¡Error! No s e encuentra el origen de la referencia.**.

```
...
ISR(TIMER1_OVF_vect)
{
  TCNT1 = 0xC2F7;
  Pulsos_Seg = Cont_Int;
  ANEM_DIR();
  ANEM_VEL();
  Cont_Int = 0;
}
...
```

Ilustración 3-29: Proceso de la función *isr_tiempo()* para el sensor de velocidad del viento

Realizado por: González A. y Ortiz W., 2023.

El cálculo de la velocidad del viento se genera en la una función denominada *ANEM_VEL()*, en la cual se realiza el cálculo por medio de la Ec.(30).

$$Dato1\_Velmps = \frac{2.25 \cdot Pulsos\_Seg}{1.0} * 0.447 \qquad (30)$$

3.4.2.3. Simulación del anemómetro del subcircuito

En la simulación se obtuvo datos de dirección y velocidad del viento. El valor de frecuencia es muy similar al valor de velocidad en m/s, se compara la frecuencia con la velocidad calculada. En la **¡Error! No se encuentra el origen de la referencia.** se puede observar los resultados de la m edición de la dirección del viento. En la **¡Error! No se encuentra el origen de la referencia.** se presenta los resultados de la simulación del sensor de velocidad.

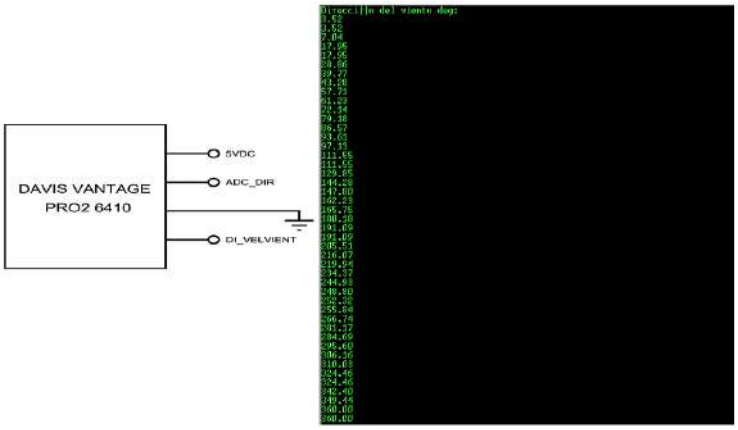


Ilustración 3-30: Resultados de simulación de sensor de dirección del viento del anemómetro Davis Vantage Pro 2 6410.

Realizado por: González A. y Ortiz W., 2023.

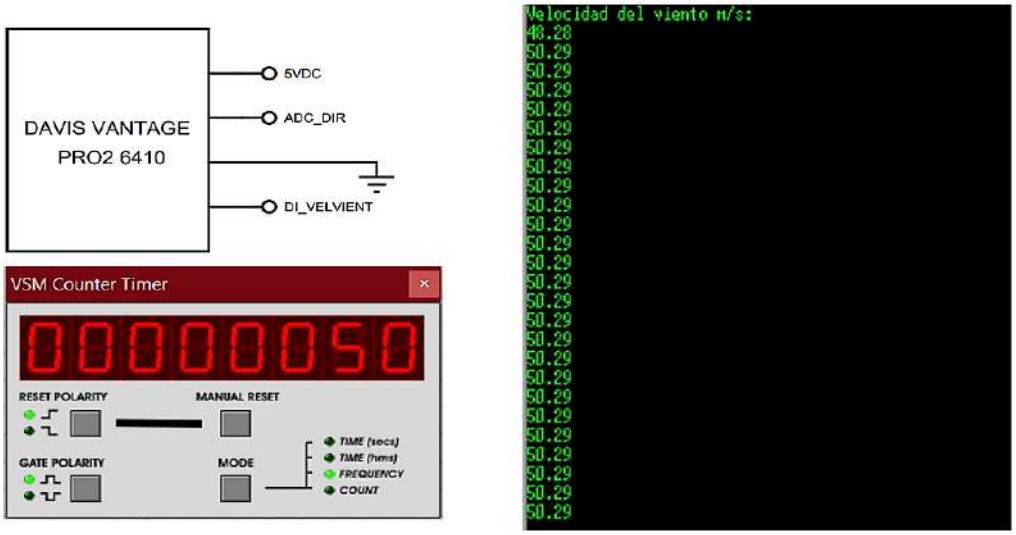


Ilustración 3-31: Resultados de simulación de sensor de dirección del viento del anemómetro Davis Vantage Pro 2 6410.

Realizado por: González A. y Ortiz W., 2023.

3.4.3. Diseño para recepción de datos de posicionamiento (subcircuito 2)

3.4.3.1. Circuito para el módulo receptor GNSS

El módulo receptor GNSS se conectó a la placa principal. Se conectó el pin RXD3 del Arduino Mega con el pin TX del módulo GNSS, junto con el pin D2 que permite identificar la entrada de datos al microcontrolador. La transmisión de los datos a la placa principal Arduino Mega 2560 se realiza por medio de la conexión RXD3. El módulo receptor se alimenta con 3.3VDC de placa principal.

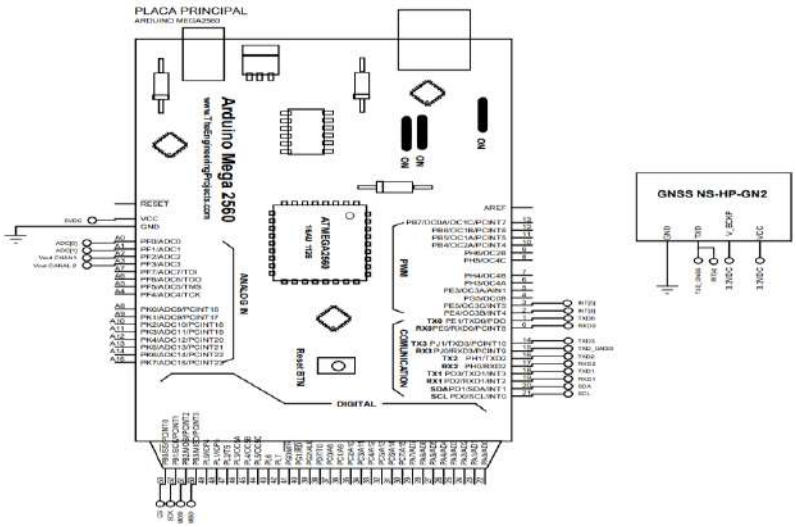


Ilustración 3-32: Circuito del módulo receptor GNSS.

Realizado por: González A. y Ortiz W., 2023.

3.4.3.2. Descripción del algoritmo para el módulo GNSS

Para establecer la comunicación entre el Arduino Mega y el módulo GNSS se realizó la inicialización del puerto serial con una velocidad de 115200 baudios.

Para la lectura y procesamiento de los datos se utilizó la librería *SkyTraqNmeaParser.h* y la librería *SoftwareSerial.h*. La librería funciona con un puerto serial y un pin para determinar una interrupción. El algoritmo realiza la lectura de datos en el puerto UART3 en el instante que ingresa una interrupción en el pin D2 de la placa Arduino, como se muestra en la Ilustración 3-33.

```
volatile boolean inService = false;
void serial3Interrupt()
{
if (inService)
{
  return;
}
inService = true;

interrupts();
while(!Serial3.available());
parser.Encode(Serial3.read());

inService = false;
}
```

Ilustración 3-33: Lectura de datos NMEA por interrupción.

Realizado por: González A. y Ortiz W., 2023.

El procesamiento de los datos de las tramas NMEA se realiza en tres funciones: *void ShowSatellites(const SatelliteInfo* si), bool GnssUpdated(U32 f, const char* buf, SkyTraqNmeaParser::ParsingType type), bool Display(U32 f, const char* buf, SkyTraqNmeaParser::ParsingType type).* La primera función permite determinar la cantidad de

satélites, la segunda función permite obtener datos actualizados del módulo receptor GNSS y la última función permite leer los datos de posicionamiento y velocidad dados por el módulo receptor GNSS. Estas tres funciones funcionan junto con la librería *SkyTraqNmeaParser.h* dada por el fabricante del módulo receptor.

En la **¡Error! No se encuentra el origen de la referencia.** se presentan las variables en las cuales s e almacenan los datos de: fecha, tiempo, posicionamiento y velocidad; junto con la función que permite obtener cada uno de estos datos.

Tabla 3-5: Variables tipo entera para la recolección de datos de módulo GNSS.

Parámetro	Función/Funciones	Variables	Descripción
Fecha	*gnss.GetYear(), gnss.GetMonth(), gnss.GetDay()*	*next_Fecha*	En la variable se almacena la fecha que se presenta en la pantalla
Tiempo	*gnss.GetHour(), gnss.GetMinute(), gnss.GetSecond(),*	*next_Hora*	En la variable se almacena el tiempo que se presenta en la pantalla
Latitud	*gnss.GetLatitude()*	*Latitud, next_Lati*	Primera variable: almacena el dato para guardar en la micro SD. Segunda variable: almacena el dato a visualizar en la pantalla.
Longitud	gnss.GetLongitude()	*Longitud, next_Longi*	Primera variable: almacena el dato para guardar en la micro SD. Segunda variable: almacena el dato a visualizar en la pantalla.
Velocidad	gnss.GetSpeedInKnots()	*Velocidad_Veh Velocidad_Vehknot, Velocidad_Vehmps Str_VelocidadVehmps next_VelVeh*	Primera variable: almacena la velocidad en knots dada por la trama NMEA Segunda variable: conversión de velocidad de la trama NMEA de cadena a flotante. Tercera variable: conversión de velocidad a m/s Cuarta variable: conversión de velocidad en m/s en una cadena, el valor se almacena en la micro SD Cuarta variable: almacena el dato a visualizar en la pantalla.

Realizado por: González A. y Ortiz W., 2023.

3.4.3.3. Simulación del módulo GNSS o del subcircuito dos

Los resultados de la simulación para la recepción de datos de posicionamiento y velocidad se presentan en la **¡Error! No se encuentra el origen de la referencia.**.

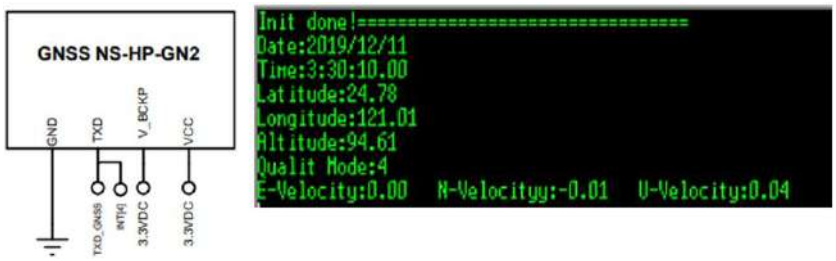


Ilustración 3-34: Resultados de simulación del módulo GNSS.

Realizado por: González A. y Ortiz W., 2023.

3.4.4. Diseño de la interfaz gráfica

3.4.4.1. Circuito para la interfaz gráfica o pantalla

Se representa el circuito de conexión entre la placa principal Arduino Mega y la pantalla Nextion modelo lNX4827K043_011R. Este circuito sirve para la visualización y selección de las opciones de los parámetros a registrar. Tiene cinco pines: VCC alimentación a 5VDC, GND, TX y RX. Los dos últimos pines son para el puerto de comunicación, para transmitir como recibir información. En la placa principal se asignó el puerto UART0: RX0 y TX0.

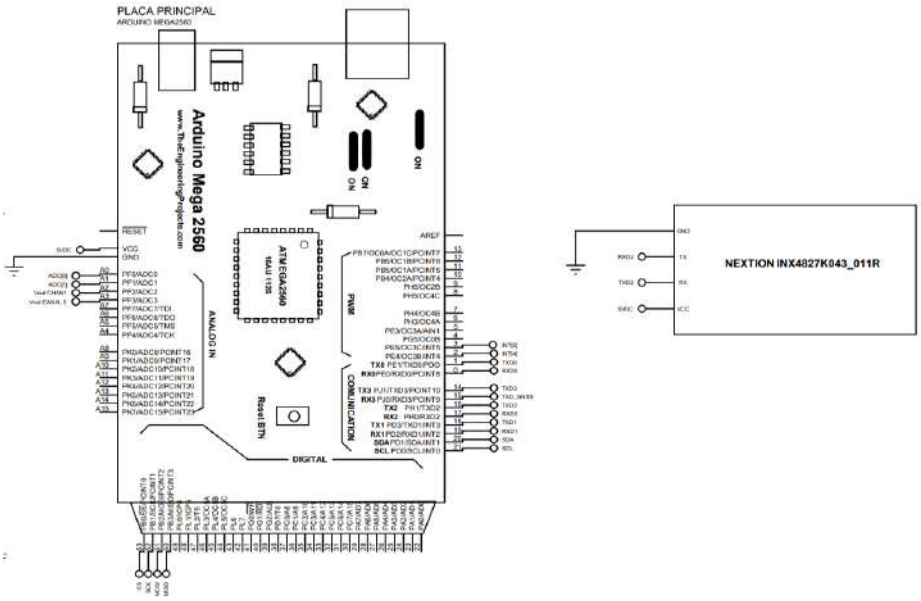


Ilustración 3-35: Circuito de la interfaz gráfica

Realizado por: González A. y Ortiz W., 2023.

En la Ilustración 3-16 se presenta el circuito de la pantalla conectada a la placa principal, de los pines de la pantalla TX y RX se envían y reciben los datos respectivamente. En el Arduino Mega 2560 se envían los datos por medio del pin TX0 y se reciben datos por el pin RX0. La conexión para la comunicación UART es cruzada, el pin RX se conecta con el TX.

3.4.4.2. Descripción del algoritmo de la interfaz gráfica

El algoritmo de la interfaz gráfica es referente a la placa principal; los objetos declarados, para la visualización y transmisión de datos hacia la placa principal, se utilizaron para el algoritmo de la placa principal, pero primero se requirió que estos objetos se asignen y posicionen en software Nextion Editor. El software Nextion Editor se utilizó para: crear el algoritmo de la pantalla Nextion y como simulador.

Primero se inicializó la comunicación serial, para ello se utilizó la función *Serial.begin()*, en este caso, la velocidad de transmisión es de 9600 baudios. En el algoritmo de la pantalla para la visualización e interacción táctil se declararon las variables, representadas en la **¡Error! No se e ncuentra el origen de la referencia.**.

Se inicializó la comunicación UART2 con una velocidad de 9600 baudios, y se empieza a enviar el texto de presentación hacia la interfaz gráfica. Para realizar la transmisión de los datos se utilizó la función *Serial.print()* y *Serial.write()*, la primera función se utilizó para enviar el contenido a visualizar en la pantalla y la segunda función se utilizó para escribir la terminación para que la pantalla identifique el contenido y pueda mostrarlo, luego de cada *Serial.print()* se ingresan tres *Serial.write(0xff)* para confirmar el envío y un *delay* de 1500ms de espera. En la **¡Error! No se e ncuentra el origen de la referencia.** se presenta la forma de ingresar los datos en el *Serial.print()*.

Tabla 3-6: Variables declaradas para el algoritmo de la interfaz grafica

Variable declarada	Valor asignado	Función que realiza
Tipo int		
Val_Confirmar	0	Se utiliza para iniciar el almacenamiento de las mediciones o datos.
Val_Detener	0	Se utiliza para detener el almacenamiento de las mediciones o datos
Val_reiniciar		Se utiliza para la selección y almacenamiento de las mediciones o datos
Val_menu	1	Se utiliza para navegar entre sensores, módulo GNSS y tiempos de registro.
Crear_Arch	0	Se utiliza para crear un nuevo archivo para almacenar las mediciones o datos
Opc_regist	0	Se utiliza para la opción de almacenar la medición en cada sensor o módulo GNSS

Caso_Pres	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de presión.
Caso_Temp	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de temperatura.
Caso_Corr	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de corriente.
Caso_Volt	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de voltaje.
Caso_VelVient	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de viento.
Caso_Dirvient	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el sensor de dirección de viento.
Caso_Gnss	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para el módulo GNSS.
Caso_PInt	0	Se utiliza para almacenar un valor de acuerdo a la opción seleccionada para los puntos de interés.
lectura_1	1	Se utiliza para cambiar de sensor.
lectura_2	1	Se utiliza crear un nuevo archivo de formato .txt.
lectura_3	1	Se utiliza para seleccionar la opción de almacenar.
lectura_4	1	Se utiliza para confirmar toda la selección de los sensores.
lectura_5	1	Se utiliza para detener el almacenamiento de datos.
lectura_6	1	Se utiliza para reiniciar el registro de datos.
Tipo Char		
Puls_Nex	n/a	Se utiliza para almacenar el dado enviado desde la pantalla por medio de la comunicación Serial.

Realizado por: González A. y Ortiz W., 2023.

```
Serial.print("objeto_nextion.txt=\"Texto_Visualizar\"")
```

Ilustración 3-36: Estructura para envío de datos tipo texto, hacia la pantalla Nextion.

Realizado por: González A. y Ortiz W., 2023.

En la estructura del *Serial.print()* se tiene el *objeto_nextion*, que se refiere al objeto declarado en el algoritmo del software Nextion Editor como un identificador para visualizar y posicionar el texto en la interfaz gráfica. El *.txt* hace referencia a que el objeto en la interfaz gráfica es de tipo texto. En la parte de *Texto_Visualizar* se ingresa el texto que se desea visualizar en la interfaz gráfica. Con las funciones anteriores se realizó el envío de las frases de inicio, esto en el v*oid setup()*, cada frase se presenta en la pantalla de acuerdo a lo requerido en el algoritmo de cada sensor o módulo y de acuerdo al menú de opciones, las frases a enviar se presentan en la **¡Error! No se encuentra el origen de la referencia.**.

Tabla 3-7: Asignación de frases el Software Nextion Editor.

Identificador Nextion	Texto Asignado	Tiempo de espera (delay) [ms]
inicio.txt	DATALOGGER	1500
inicio2.txt	ESPOCH_AUTOMOTRIZ	1500
iniciogen.txt	Iniciando BMP280:	1500
iniciogen.txt	BMP280 no encontrado...	1500
iniciogen.txt	Revisar BMP280…	1500
iniciogen.txt	Iniciando Tarjeta Micro SD ...	1500
iniciogen.txt	¡Fallo en inicialización de Micro SD!	1500
iniciogen.txt	Tarjeta Micro SD no conectada	1500
iniciogen.txt	Inicio de Micro SD correcto	1500
iniciogen.txt	Acceso correcto en el archivo	1500
archivo.txt	*Arch_txt*	0
iniciogen.txt	¡Error al abrir el archivo!	1500
iniciogen.txt	¡Límite máximo de archivos creados!	1500
archivo.txt	Limpiar memoria de tarjeta Micro SD	0

Realizado por: González A. y Ortiz W., 2023.

En el *void loop()*, se realizó la transmisión de información entre de la pantalla y la placa Arduino. La pantalla además de ser útil para la visualización, presenta la característica de ser táctil y con ello enviar datos desde la pantalla hacia la placa principal. Se asignaron seis objetos de tipo botón táctil que al momento de presionar y dejar de presionar se envía un dato, en la **¡Error! No se e ncuentra el origen de la referencia.** se presentan los objetos utilizados para en el envío desde la pantalla y la variable para la recepción en la placa principal, y el valor asignado.

Tabla 3-8: Objetos tipo botón declarados en el software Nextion Editor y variable para algoritmo de la placa principal.

Objeto tipo botón declarado en el software Nextion Editor	Variable asignada en el algoritmo de la placa principal	Valor que toma la variable al presionar el botón táctil	Valor que toma la variable al dejar de presionar el botón táctil	Función del botón táctil
bmenu	*Puls_Nex*	*a*	*b*	Se utiliza para cambiar de sensor, módulo GNSS y tiempos de registro.
barch	*Puls_Nex*	*c*	*d*	Se utiliza para crear un nuevo archivo de extensión ".txt".
balm	*Puls_Nex*	*e*	*f*	Se utiliza para seleccionar la opción de almacenar, y selección de tiempo de registro.

bconfir	*Puls_Nex*	*g*	*h*	Se utiliza para confirmar toda la selección de opciones en cada sensor y tiempo de registro.
bdetener	*Puls_Nex*	*i*	*j*	Se utiliza para detener la visualización y registro de datos.
breiniciar	*Puls_Nex*	*k*	*l*	Se utiliza para reiniciar selección de sensores, módulo GNSS, tiempo de registro, almacenamiento y creación de un nuevo archivo.

Realizado por: González A. y Ortiz W., 2023.

Primero se realizó la lectura de los datos enviados desde la pantalla, para ello se aplicó una decisión para determinar si existen datos a leer en la comunicación serial por medio de la función *Serial.available()*; en el caso de no existir datos se sigue a la siguiente instrucción. En el caso de existir datos a leer, el valor leído se almacena en la variable denominada *Puls_Nex*, desde la pantalla se envían las siguientes letras: *a, b, c, d, e, f, g, h, i, j, k, l*; y estas se asignan de la siguiente forma:

- Si el *dato* es "*a*" *lectura1* toma el valor de 0.
- Si el *dato* es "*b*" *lectura1* toma el valor de 1.
- Si el *dato* es "*c*" *lectura2* toma el valor de 0.
- Si el *dato* es "*d*" *lectura2* toma el valor de 1.
- Si el *dato* es "*e*" *lectura3* toma el valor de 0.
- Si el *dato* es "*f*" *lectura3* toma el valor de 1.
- Si el *dato* es "*g*" *lectura4* toma el valor de 0
- Si el *dato* es "*h*" *lectura4* toma el valor de 1.
- Si el *dato* es "*i*" *lectura5* toma el valor de 0,
- Si el *dato* es "*j*" *lectura5* toma el valor de 1,
- Si el *dato* es "*k*" *lectura6* toma el valor de 0,
- Si el *dato* es "*l*" *lectura6* toma el valor de 1,

Las acciones que se aplican al momento de que cada variable *lectura* tome el valor de uno se presentan en la **¡Error! No se encuentra el origen de la referencia.**.

Tabla 3-9: Acciones a tomar de acuerdo al valor de uno asignado en la variable *lectura*.

Variable	Valor que tome la variable	Acción
lectura_1	1	Se suma en uno la variable *Val_menu*.
lectura_2	1	Se asigna el valor de uno a la variable *Crear_Arch*.
lectura_3	1	Se asigna el valor de uno a la variable *Opc_regist*.
lectura_4	1	Se asigna el valor de uno en la variable *Val_Confirmar*.
lectura_5	1	Se asigna el valor de uno en la variable *Val_Detener*.
lectura_6	1	Se asigna el valor de uno en la variable *Val_reiniciar*.

Para visualizar las opciones para cada sensor, se envía hacia la pantalla un dato que describe la opción de guardar, asignando el objeto declarado a cada frase, como se muestra en la **¡Error! No s e encuentra el origen de la referencia.**. Para el caso de selección del tiempo de muestreo o de registro de datos se puede utilizar la opción guardar.

Tabla 3-10: Visualización de las opciones para cada sensor

Identificador Nextion	Texto Asignado
opcion2.txt	Presione guardar

Realizado por: González A. y Ortiz W., 2023.

Para la interacción y presentación del menú en la pantalla se realizó un algoritmo que implica utilizar un contador con la variable *Val_menu* que puede tomar un valor de hasta diez unidades, en el caso de llegar a diez se reinicia el contador a cero. El contador se sumará en uno al momento en que la variable *lectura_1* tome un valor de cero. Para cada valor de *Val_menu* se asigna un caso por medio de la función *switch()*, cada caso representa a un sensor, módulo GNSS o tiempo de registro; se tiene en total diez casos y en cada caso se selecciona la opción guardar. De los diez casos, ocho se asignaron para sensores y módulo GNSS, los otros dos casos restantes se utilizaron para seleccionar las opciones de tiempo de registro. Para crear un nuevo archivo para el registro de datos se utilizó la variable *Crear_Arch*, para crear un nuevo archivo la variable debe tomar el valor de uno al momento en que la variable *lectura_2* tome el valor de cero, *Crear_Arch* toma el valor de cero una vez que se haya creado el archivo. Para el caso de las opciones se utilizó un variable *Opc_Tiempo* para la opción guardar, la variable toma el valor de uno al momento de que *lectura_3* tome un valor de cero, en la se muestra las acciones a tomar de acuerdo al valor asignado en *Opc_Tiempo*, para el caso de módulos y sensores, al final de cada selección estas dos variables se reinician a cero, esto de forma independiente para cada sensor de acuerdo a cada caso en la función *switch()*.

Tabla 3-11: Matriz de acciones para cada sensor.

Sensor/Módulo	Variable	Valores	
		0	**1**
Sensor de presión	*Caso_Pres*	No se almacena medida de presión.	Se almacena medida de presión.
Sensor de temperatura	*Caso_Temp*	No se almacena medida de temperatura.	Se almacena medida de Temperatura
Sensor de corriente	*Caso_Corr*	No se almacena medida de corriente.	Se almacena medida de corriente.
Sensor de voltaje	*Caso_Volt*	No se almacena medida de voltaje.	Se almacena medida de voltaje
Sensor de velocidad de viento	*Caso_VelVient*	No se almacena medida de velocidad del viento.	Se visualiza medida de velocidad del viento.
Sensor de dirección del viento	*Caso_Dirvient*	No se almacena medida de dirección del viento.	Se almacena medida de dirección del viento.
Modulo GNSS	*Caso_Gnss*	No se almacena datos de posición y velocidad de vehículo	Se almacena datos de posición y velocidad de vehículo.
Pulsador para identificación de puntos de interés	*Caso_PInt*	No se almacena la cantidad de puntos de interés.	Se almacena datos de puntos de interés.

Realizado por: González A. y Ortiz W., 2023.

En la pantalla se visualiza la opción seleccionada para cada sensor en el objeto *resultmenu.txt* declarado en el Software Nextion.

Para la selección del tiempo de muestreo o registro de datos se tiene las opciones presentadas en la **¡Error! No se encuentra el origen de la referencia.**. El tiempo seleccionado se presenta en el o bjeto *resultmenu.txt.*

Tabla 3-12: Opciones de tiempo de muestro o registro de datos.

Opciones de tiempo de muestreo	Variable
Muestreo de un segundo	switTiempo
Muestreo de dos segundos	switTiempo

Realizado por: González A. y Ortiz W., 2023.

Tabla 3-13: Visualización de las medidas o parámetros físicos obtenidos.

Objeto Nextion	Variable declarada a visualizar
t6.txt	*next_TiempoTot*
x10.val	*next_Pres*
x11.val	*next_Temp*
x7.val	*next_Corr*
x6.val	*next_Volt*
x8.val	*next_VelVient*
x9.val	*next_DirVient*
t0.txt	*next_Fecha*
t1.txt	*next_Hora*
t3.txt	*next_Lati*
t2.txt	*next_Longi*
t5.txt	*next_Alti*
t4.txt	*next_VelVeh*
n0.val	*next_Puntos*

Realizado por: González A. y Ortiz W., 2023.

Para confirmar toda la selección para los sensores y módulo GNSS la variable *Val_Confirmar* tiene que tomar un valor de uno, esto se realiza por medio de una función condicional en la cual se asigna un valor de uno a la variable *Val_Confirmar* al momento en que *lectura_4* tome un valor de cero. Si la variable *Val_Confirmar* toma el valor de uno, se realiza el almacenamiento de las variables para cada sensor, de acuerdo a lo seleccionado en el menú de sensores, módulo GNSS y tiempo de registro. En la pantalla se visualizan las medidas o datos en todo momento, en una posición asignada con la ayuda de un objeto declarado en el software Nextion Editor. En la ¡Error! N o se encuentra el origen de la referencia. se presenta la variable a mostrar de acuerdo al objeto declarado.

Para enviar datos numéricos, se declaró en el software Nextion Editor un objeto de tipo número, el envío se realiza hacia la pantalla por medio de función *Serial.print()*. Para confirmar él envió de cada valor numérico se ingresan tres *Serial.write(0xff)*. La estructura para el envío de datos numéricos se presenta en la ¡Error! No se encuentra el origen de la referencia..

```
Serial.print("objeto_nextion.val")
Serial.print(Variable_Visulaizar)
```

Ilustración 3-37: Estructura para envío de datos tipo número a la pantalla Nextion.

Realizado por: González A. y Ortiz W., 2023.

En el software Nextion Editor se cargó la imagen a utilizar y los objetos para realizar el envió y posicionamiento de la información. En la **¡Error! No se encuentra el origen de la referencia.** s e presentan los objetos declarados para para la transmisión y visualización de la información.

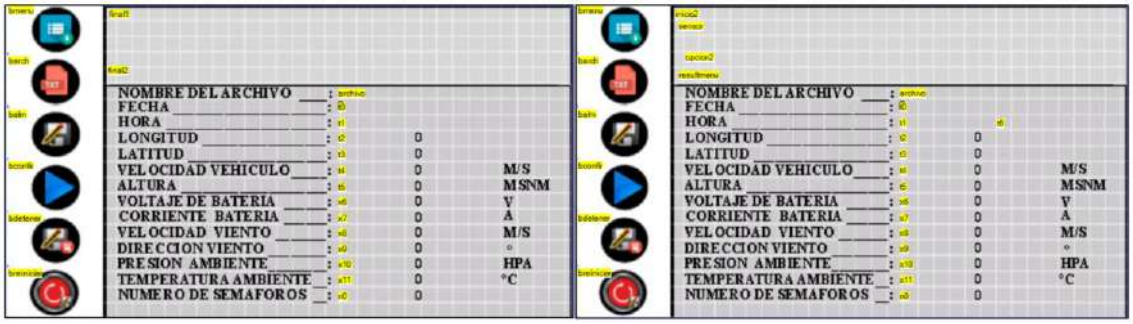


Ilustración 3-38: Declaración y posición de objetos en el software Nextion Editor.

Realizado por: González A. y Ortiz W., 2023.

Cuando el botón *bdetener* sea presionado la variable *detener* toma el valor de uno y todo el registro de datos se detendrá, demás, se presentarán dos frases*:*

- *final1:* "*REGISTRO FINALIZADO"*
- *final2:* "*Desea reiniciar el registro de datos*?"

Para que se genere una acción al presionar el botón *breiniciar*, primero se debe presionar el botón *bdetener*. Al presionar el botón *breiniciar* la variable *reiniciar* toma el valor de uno y con ello se reinicia el programa para seleccionar de nuevo la opción en cada sensor y módulo GNSS, seleccionar el tiempo de muestro y se debe crear un nuevo archivo para el registro de datos.

3.4.4.3. Simulación del circuito de la interfaz gráfica

Para la simulación se utilizó el software Nextion Editor. En la **¡Error! No se encuentra el origen d e la referencia.** se presentan los resultados.

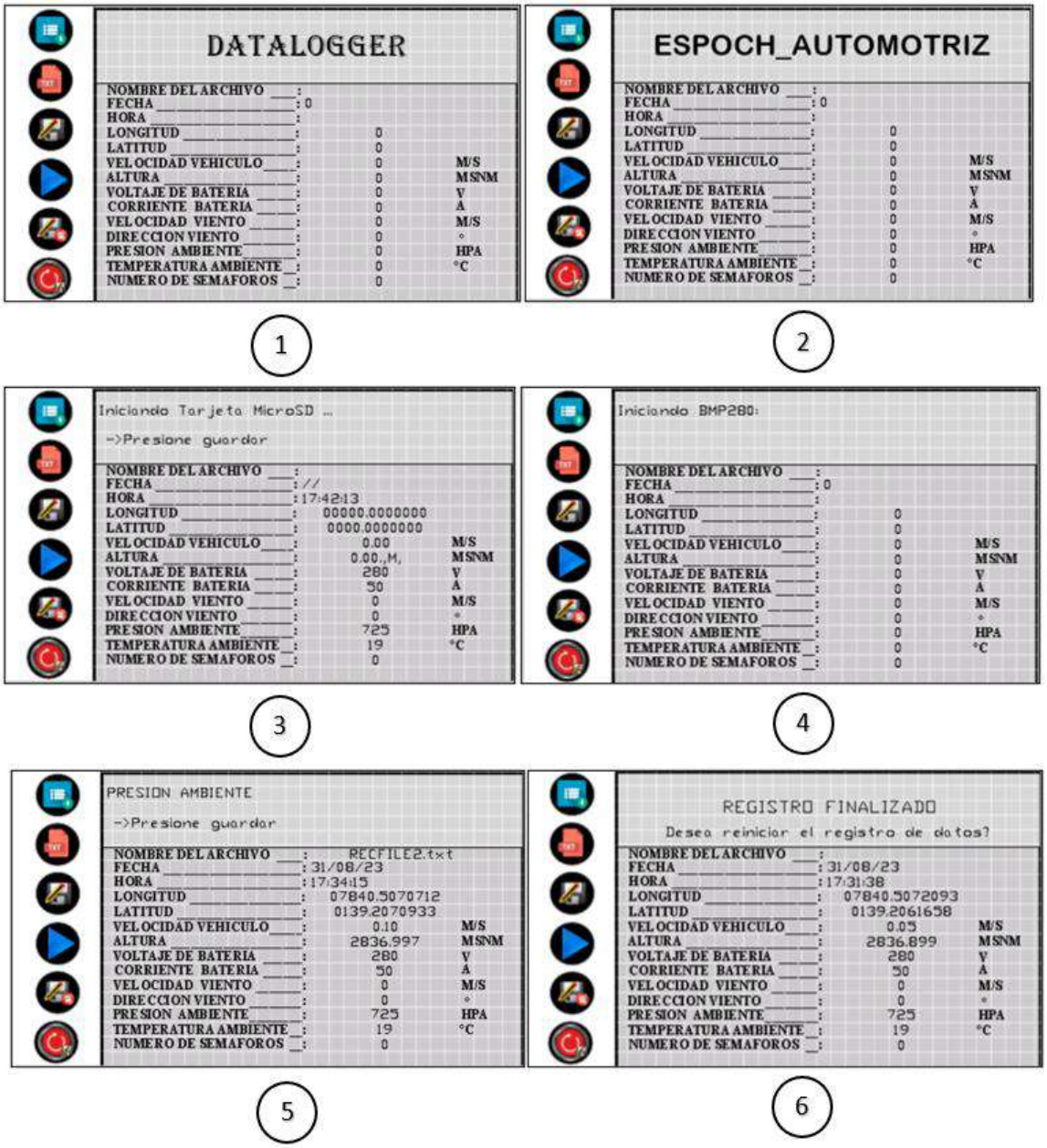


Ilustración 3-39: Simulación de interfaz gráfica.

Realizado por: González A. y Ortiz W., 2023.

3.4.5. Diseño del almacenamiento de las magnitudes físicas

3.4.5.1. Circuito para el módulo micro SD

El módulo de la micro SD requirió de una conexión SPI para la transmisión de los datos y con ello escribir en la tarjeta micro SD las mediciones. Tanto en el módulo como en la placa Arduino Mega 2560 se conectaron los mismos pines para la comunicación SPI.

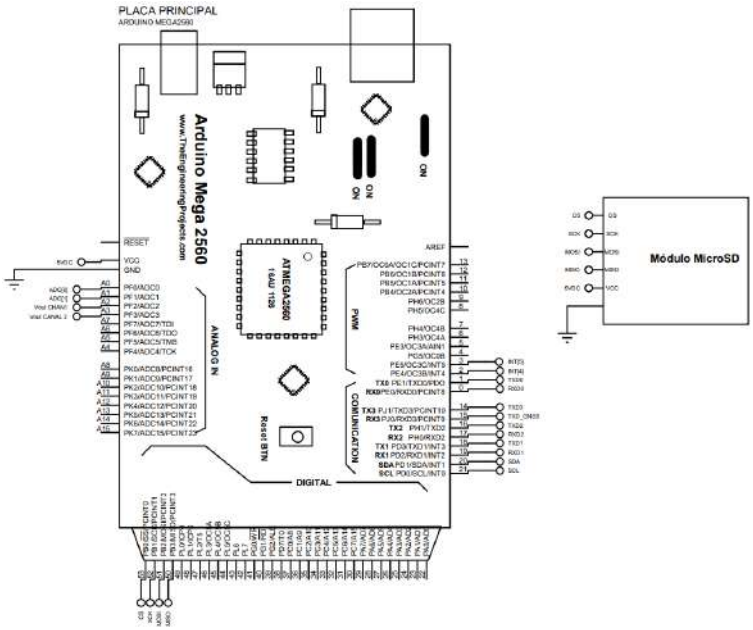


Ilustración 3-40: Circuito del módulo de la micro SD.

Realizado por: González A. y Ortiz W., 2023.

En la **¡Error! No se encuentra el origen de la referencia.** se representa el circuito de conexión e ste módulo, en el módulo utilizado para el datalogger se aplica 5V de corriente directa debido a que presenta un regulador de voltaje de 5V a 3.3V, el voltaje que entrada al módulo no puede ser de 3.3V.

3.4.5.2. Descripción del algoritmo para el módulo micro SD

El algoritmo empezó con la inicialización de las librerías para la comunicación SPI y para la micro SD: *SPI.h* y *SD.h;* luego se definió el pin para la selección del módulo de la micro SD, en este caso se ingresó el número del pin D53 al cual se lo denominó *Chip_Selec*. Luego se creó una variable tipo *String* denominada *Arch_txt*, esta variable fue utilizada para nombrar los archivos creados para la escritura en la tarjeta. Luego se creó un objeto de tipo File, denominado *Recfile_txt*.

Se creó una función denominada Archivo_SD() en la cual se inicializa la tarjeta micro SD por medio de la comprobación del pin D53, normalmente si la tarjeta no se conecta se genera un error en la inicialización. En el caso de que la tarjeta esté conectada correctamente se crea un archivo con la ayuda de un contador declarado como *count*, una variable declarada como tipo *bool* denominada *file* definida como *false*, una función *SD.exists()* y una función *while()*. La variable *count* sirve para crear un nuevo nombre para el nuevo archivo, la variable *file* sirve para salir de la función *while()* luego de crear el archivo, y la función *SD.exists()* sirve para comprobar si existe un archivo en la tarjeta. Luego se abre el archivo creado por medio del objeto de tipo *File*, y se

comprueba que se haya accedido al archivo; se escribirá el nombre de los parámetros a almacenar por medio de la función *myFile.print()*, y luego se cierra el archivo por medio de la función *myFile.close()* con el fin de asegurar el almacenamiento de los datos; en el caso de que el archivo no se abra, se visualiza un error al abrir el archivo, si esta última parte sucede, lo normal es que la cantidad de archivos creados haya llegado al límite.

Se creó otra función denominada *registro_Datos()* en la cual se realiza la escritura de los datos medidos u obtenidos por medio de los sensores y módulo GNSS. La forma de almacenar los datos se presenta en la **¡Error! No se encuentra el origen de la referencia.**.

```
...
Recfile_txt.print(Dato_LM35);
Recfile_txt.print(";");
...
```

Ilustración 3-41: Función para almacenamiento de datos en tarjeta micro SD.

Realizado por: González A. y Ortiz W., 2023.

Para el tiempo de registro se crearon las variables:

- *Tiempo_Reg*
- *Opc_Tiempo*

Por defecto a la variable *Opc_Tiempo* se le asignó un valor de uno para un registro de cada un segundo, esta variable sirve para asignar un valor para el tiempo de registro a seleccionar, con un valor de uno se tiene un registro de cada un segundo y para un valor de 2 se tiene un registro de cada dos segundos. La variable *Tiempo_Reg* se creó para un contador que permite comparar el tiempo de registro seleccionado para ejecutar la función *registro_Datos()*.

Se utilizó los registros del TIMER4 del microcontrolador ATMEGA2560. Primero, se configuró el TIMER4 en el registro TCCR4A junto con el preescalador a través del registro TCCR4B, luego se inicializa el TIMER4 para 1.0 segundo antes del desbordamiento utilizando el registro TCNT4 y finalmente se habilitó la interrupción para el TIMER4, todo esto debe ser configurado en la función *void setup()*, en la **¡Error! No se encuentra el origen de la referencia.** se presenta esta i nicialización.

```
TCCR4A = 0;
TCCR4B = 0;
TCCR4B |= (1<<CS40)|(1 << CS42);
TCNT4 = 0xE17D;
TIMSK4 |= (1 << TOIE4);
```

Ilustración 3-42: Inicialización de TIMER4

Realizado por: González A. y Ortiz W., 2023.

Para registrar el tiempo cada segundo o cada dos segundos se añadió la función *ISR(TIMER4_OVF_vect),* en la cual se ejecuta la función *registro_Datos()* para el almacenamiento o escritura de los parámetros de sensores y módulo GNSS sobre la micro SD. En esta función se debe cumplir dos condiciones principales, la primera se relaciona con la confirmación del registro de datos y la segunda hace referencia al tiempo de registro seleccionado. Para que el tiempo de registro sea el correcto, al registro TCNT4 se le asignó el valor del tiempo de interrupción antes del desbordamiento, en este caso para 1.0 segundos, junto con un contador para configurar el tiempo de registro, este contador se reinicia una vez ejecutada la función *ISR(TIMER4_OVF_vect)*. En la **¡Error! No se encuentra el origen de la referencia.** se presenta e sta parte del algoritmo.

```
...
ISR(TIMER4_OVF_vect){
  TCNT4 = 0xC2F8;
  if((Val_Confirmar == 1) && (Val_Detener != 1)){
    Tiempo_Reg++;
    if(Tiempo_Reg == Opc_Tiempo){
      Registro_Datos();
      Tiempo_Reg=0;
    }
  }
}
...
```

Ilustración 3-43: Algoritmo de la función para el registro de datos

Realizado por: González A. y Ortiz W., 2023.

Los tiempos de muestreo se verificaron con la ayuda de un osciloscopio. Los tiempos medidos de ejecución del registro de datos fueron exactamente 1000 ms para un segundo y 2000 ms para dos segundos, como se muestra en la **¡Error! No se encuentra el origen de la referencia.** e **¡Error! No se encuentra el origen de la referencia.** respectivamente.

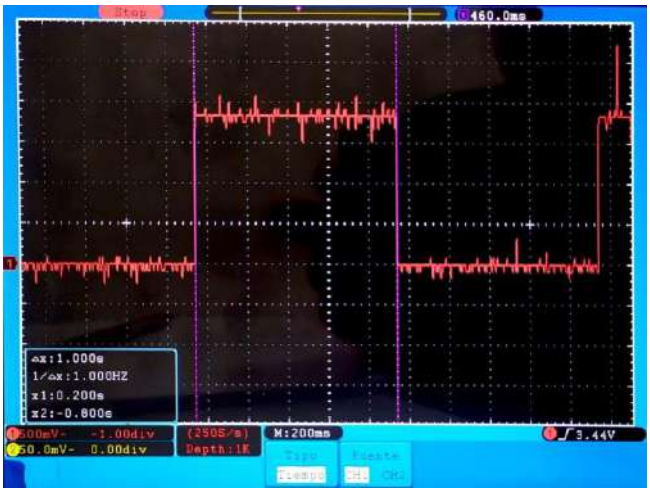


Ilustración 3-44: Medición de tiempo de registro de datos para 1 segundo.

Realizado por: González A. y Ortiz W., 2023.

Ilustración 3-45: Medición de tiempo de registro de datos para 2 segundo.

Realizado por: González A. y Ortiz W., 2023.

3.4.5.3. Simulación del almacenamiento de los parámetros seleccionados

La simulación se realizó por medio de la creación de un archivo y la escritura de los datos de los parámetros físicos seleccionados, el archivo generado es de formato *.txt*. En la **¡Error! No se encuentra el origen de la referencia.** se presentan los resultados obtenidos, se puede observar que cada dato está separado con un punto y coma, esto servirá al momento de importar los datos al software Excel como identificador en la separación de datos.

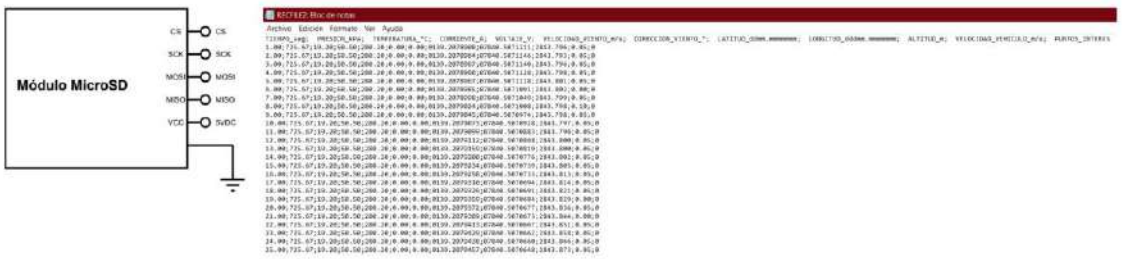


Ilustración 3-46: Resultados de simulación del registro de los parámetros seleccionados.

Realizado por: González A. y Ortiz W., 2023.

3.4.6. Diseño del envío y recepción de datos entre placas Arduino.

3.4.6.1. Circuito para el envío y recepción de datos

En este circuito se encuentra el Arduino Mega 2560 y el Arduino Pro Mini, donde, la placa Arduino Mega 2560 es la encargada de detectar y recibir los datos o información enviada desde la placa Arduino Pro Mini. Se utilizó la comunicación UART para las dos placas Arduino; en la placa Arduino Mega 2560 se tiene cuatro puertos UART de los cuales el puerto UART1 se destinó para el Arduino Pro Mini del anemómetro. En la **¡Error! No se encuentra el origen de la r eferencia.** se muestra las conexiones entre las placas.

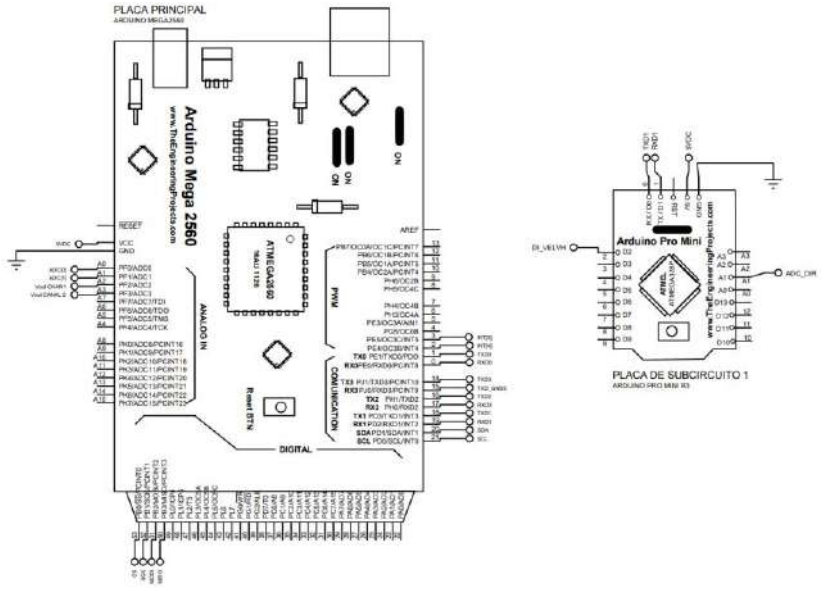


Ilustración 3-47: Circuito para la comunicación entre placas Arduino.

Realizado por: González A. y Ortiz W., 2023.

3.4.6.2. Descripción del algoritmo para la comunicación entre placas Arduino

Hay que considerar que se utilizaron dos placas Arduino:

- Placa principal Arduino Mega 2560, recibe datos.
- Placa Arduino Pro Mini para anemómetro (placa de microcontrolador de subcircuito), transmite datos.

Cada placa de microcontrolador tiene un algoritmo propio, en los tres algoritmos se inicializa la comunicación UART a una velocidad de 9600 baudios, como se muestra en la **¡Error! No se encuentra el origen de la referencia.** e **¡Error! No se encuentra el origen de la referencia.**.

```
...
Serial1.begin(115200);
...
```

Ilustración 3-48: Inicialización de comunicación UART para Arduino Mega 2560.

Realizado por: González A. y Ortiz W., 2023.

```
...
Serial.begin(115200);
...
```

Ilustración 3-49: Inicialización de comunicación UART en Arduino Pro Mini.

Realizado por: González A. y Ortiz W., 2023.

Para el algoritmo de transmisión de datos se utilizó la función *Serial.print()*, para recibir los datos se utilizó la función *Serial.read()*.

La placa del subcircuito 1, transmite los datos de velocidad y dirección, el envió de datos se realizó de carácter en carácter, para ello las mediciones de velocidad y dirección se convirtieron en *String* y luego en carácter; los caracteres se almacenaron en dos vectores de siete posiciones: *charvelocidad* para velocidad y *chardireccion* para dirección. Para el envío de datos se creó una trama de datos la cual se estructuró como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

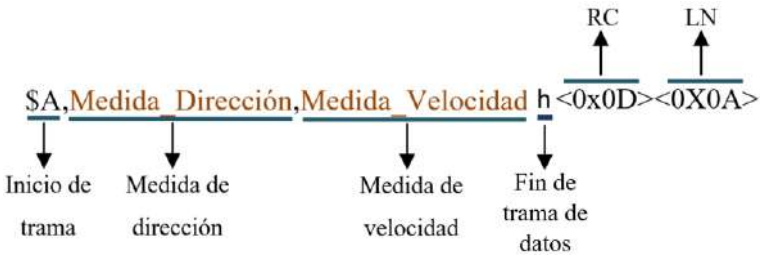


Ilustración 3-50: Trama de datos para envío de datos de velocidad y dirección del viento.

Realizado por: González A. y Ortiz W., 2023.

```
...
for (int i = 0; i <= 3; i = i + 1)
{
  Serial.print(chardireccion[i]);
}
...
for (int i = 0; i <= 4; i = i + 1)
{
  Serial.print(charvelocidad[i]);
}
...
```

Ilustración 3-51: Envió de medida de dirección y velocidad del viento

Realizado por: González A. y Ortiz W., 2023.

El envío de la medida de la dirección y velocidad se realiza de carácter a carácter por medio de un vector como se muestra en la **¡Error! No se encuentra el origen de la referencia.**.

La placa Arduino Mega 2560 se encarga de recibir la información transmitida por la placa Arduino Pro Mini. Los datos transmitidos por el puerto UART1 se almacenaron carácter a carácter por medio de un vector, como se muestra en la **¡Error! No se encuentra el origen de la r eferencia.**. Para recibir los datos se comprueba si existen datos disponibles para leer, para ello se utilizó la función *Serial.available()*.

```
...
for (j = 0; j <= 12; j = j + 1){
  while (!Serial1.available());
  ANEM[j] = Serial1.read();
}
...
```

Ilustración 3-52: Recepción de datos del anemómetro y GNSS en la placa principal Arduino Mega.

Realizado por: González A. y Ortiz W., 2023.

3.4.6.3. Simulación de la comunicación entre placas Arduino

Los resultados de la simulación para la transmisión y recepción de datos entre el Arduino Pro Mini y Arduino Mega 2560 se presentan en la **¡Error! No se encuentra el origen de la r eferencia.**.

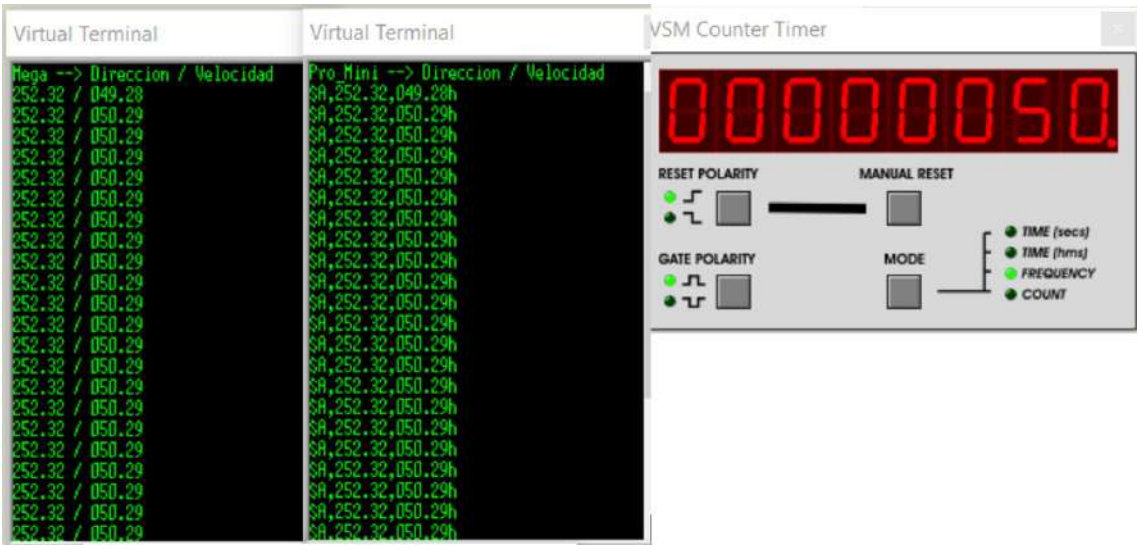


Ilustración 3-53: Simulación de la transmisión y recepción de datos entre placas Arduino.

Realizado por: González A. y Ortiz W., 2023.

3.4.7. Diseño del circuito de alimentación y carga

3.4.7.1. Dimensionamiento de baterías

Para la alimentación del sistema se ha empleado cuatro celdas de Li-ion. El cálculo de potencia teórica consumida se presenta en la **¡Error! No se encuentra el origen de la referencia.**. De a cuerdo al dato de consumo de potencia teórica total se determinó la cantidad de energía requerida para el funcionamiento del datalogger. Se requiere que el datalogger funcione como mínimo tres horas. Para cálculo de la potencia a suministrar, se consideraron cuatro baterías Li-ion 18650 conectado en serie. En total la energía proporcionada de las baterías se calculó en 71.04 Wh, el cálculo se realizó con la Ec.(31).

$$P_{baterias} = 4 \cdot I_{18650} \cdot V_{18650} \quad (31)$$

$$P_{baterias} = 4 \cdot 4.8\ Ah \cdot 3.7\ V$$

$$P_{baterias} = 71.04 Wh$$

Donde:

$P_{baterias}$ = capacidad de la fuente de alimentación en Wh.

I_{18650} = capacidad de batería 18650 Li-ion en Ah.

V_{18650} = voltaje nominal de batería 18650 Li-ion.

Tabla 3-14: Estimación de consumo de componentes del datalogger.

Componente	Modelo	Entrada de voltaje[V]	Corriente [mA]	Corriente [A]	Potencia [W]
Sensor de presión barométrica	BMP280	3.3	4.2	0.0042	0.01386
Sensor de Temperatura	LM35a	5	10	0.01	0.05
Sensor de velocidad de viento relativa al vehículo	Anemómetro Davis Vantage Pro2 6410	5	25	0.025	0.125
Sensor de corriente de batería	LEM DHAB S/137	5	20	0.02	0.1
Sensor de voltaje de batería	JXDA4U	12	25	0.025	0.3
GNSS	PX1122R - Modulo receptor GNSS	3.3	100	0.1	0.33
Antena GNSS	Antena de Multifrecuencia	3.3	50	0.05	0.165
Microcontrolador	Arduino Mega 2560	5	360	0.36	1.8
	Arduino Pro Mini - ATmega328P	10	160	0.16	1.6
	Módulo Micro SD	5	100	0.1	0.5
Diodo LED	Diodo LED	1.2	20	0.02	0.024
Interfaz gráfica o pantalla	NEXTION INX4827K043-011R	5	410	0.41	2.05
BMS	BMS 4S HX A01	16.8	0.3	0.0003	0.00504
Regulador de voltaje	HW-613	12	0.85	0.00085	0.0102
	HW-613	5	0.85	0.00085	0.00425
Resistencias	Resistencias	-	-	-	10
				Total	17.07735

Realizado por: González A. y Ortiz W., 2023.

Teniendo el dato de la potencia de las baterías y el dato de la potencia teórica de consumo, se obtuvo el tiempo aproximado de funcionamiento por medio de la Ec.(32).

$$t_{funcionamiento} = \frac{P_{baterias}}{P_{teorica}} \quad (32)$$

$$t_{funcionamiento} = \frac{71.04Wh}{17.08\,W}$$

$$t_{funcionamiento} = 4.15h$$

Donde:

$t_{funcionamiento}$ = tiempo estimado de alimentación.

En teoría el tiempo de funcionamiento del datalogger con cuatro baterías de Li-ion 18650 conectadas en serie sería de aproximadamente 4.15h.

Ilustración 3-54: Batería 18650 de Li-ion

Fuente: UltraFire, 2023.

3.4.7.2. Circuito de alimentación y carga

En la alimentación se ocupa un fusible para proteger todo el circuito del datalogger y un interruptor para el encendido, las placas Arduino se alimentan con 9V por medio de un regulador de voltaje HW-613 de 14.8V a 9V; la pantalla Nextion, sensores y módulo micro SD ocupan 5V regulados por medio de un regulador de voltaje HW-613 de 9V a 5V. Se utiliza un regulador de voltaje HW-613 de 14.8V a 12V para alimentar el sensor de voltaje. El módulo receptor GNSS se alimenta con 3.3V regulados desde la placa principal. En la **¡Error! No se encuentra el origen de la referencia.** se presenta el circuito principal de alimentación y carga. El circuito de carga consta de un conector tipo Jack, un regulador de voltaje de 20V a 16.8V, un BMS 4S, un módulo de nivel de carga 4S 18650 y 4 baterías Li-ion 18650. Para la carga se utiliza un cable con alimentación de 12V, esta tensión se eleva a 20V por medio de un convertidor DC-DC, debido a que el BMS 4S acepta un voltaje de carga de 16.8V.

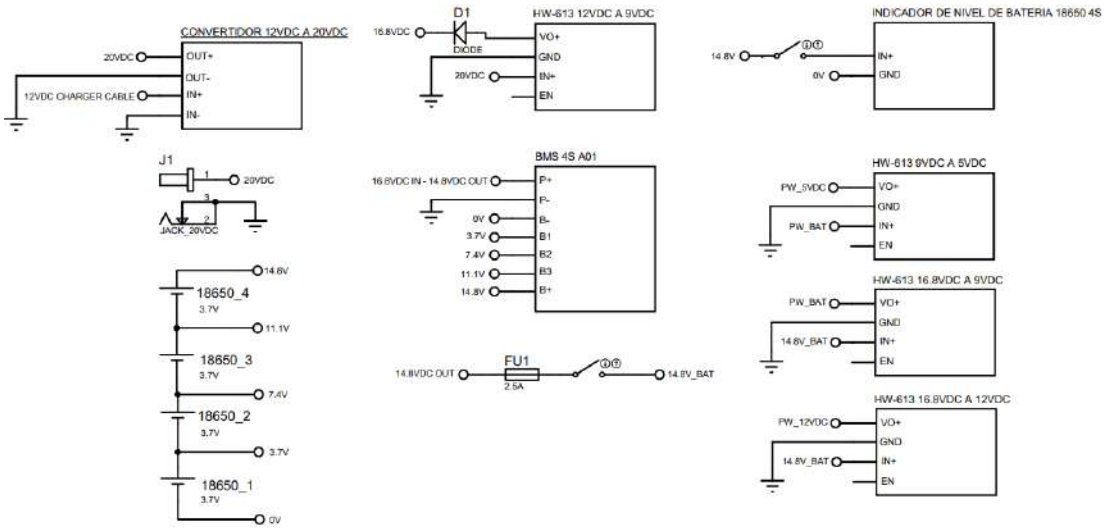


Ilustración 3-55: Circuito principal de alimentación y carga.

Realizado por: González A. y Ortiz W., 2023.

3.5. Construcción de datalogger

Al inicio se realizó un prototipo, que se presenta en la Ilustración 3-62, para verificar las conexiones y funcionamiento, de acuerdo al diagrama del circuito eléctrico realizado en el software Proteus y los algoritmos.

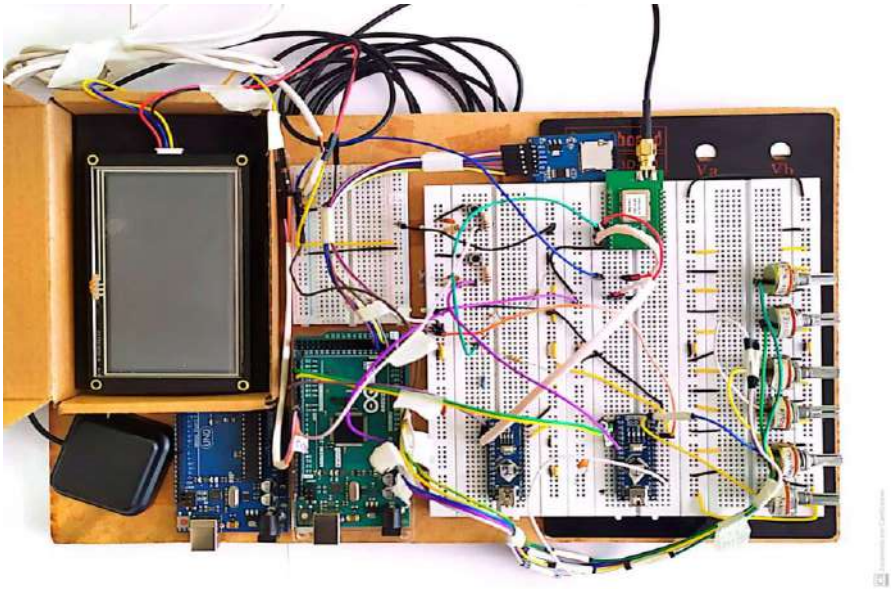


Ilustración 3-56: Prototipo Datalogger

Realizado por: González A. y Ortiz W., 2023.

La placa PCB, diseñada en el software Proteus, y el esquema de impresión se muestra en la **¡Error! No se encuentra el origen de la referencia.** y ANEXO N.

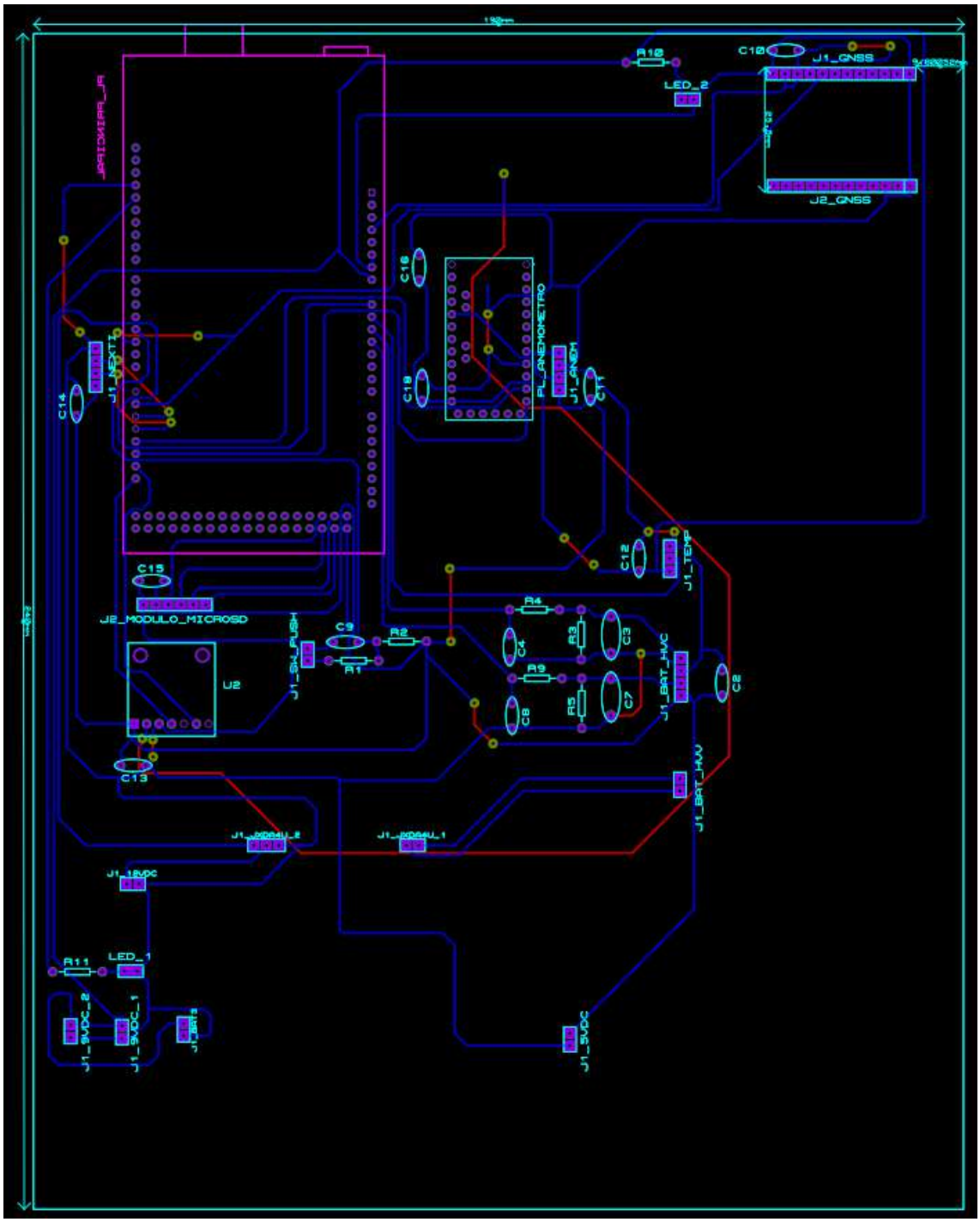


Ilustración 3-57: Diseño de placa PCB

Realizado por: González A. y Ortiz W., 2023.

Las partes del datalogger como: PCB, fuente de alimentación, placas Arduino, sensor de presión, módulo GNSS, módulo micro SD y pantalla fueron ensambladas de forma proporcional y ordenada en el interior de una caja con las dimensiones presentadas en la Ilustración 3-60. Se tomó en cuenta el requerimiento de portabilidad del equipo, para realizar la toma de datos en ruta.

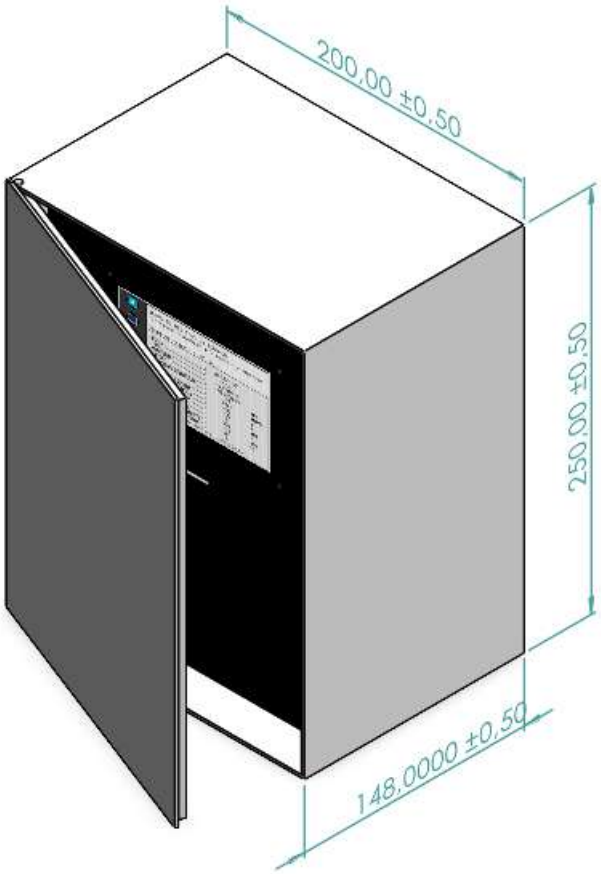


Ilustración 3-58: Caja para la implementación del datalogger

Realizado por: González A. y Ortiz W., 2023.

CAPÍTULO IV

4. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

Los datos obtenidos se presentan en la **¡Error! No se encuentra el origen de la referencia.** y en el ANEXO O y ANEXO P . Para pantalla se presenta el resultado final de la secuencia de visualización en la **¡Error! No se encuentra el origen de la referencia.**. En la en Tabla se p resenta el resultado de las mediciones para cada transductor y módulo GNSS.

4.1. Resultados de pruebas de almacenamiento de datos

Los datos registrados se almacenaron en orden en un archivo de texto, los valores de los datos almacenados presentaron una relación con cada magnitud física medida. Se realizaron dos registros, para 1 y 2 segundos de muestreo, en el primer registro se tomaron mediciones de presión, temperatura, corriente, velocidad del viento, dirección del viento, datos GNSS, altura y puntos de interés; y en el segundo registro se tomaron mediciones únicamente de voltaje.

Se tomó como tiempo de registro 10 minutos para todas las pruebas, es decir 600 segundos. En esta parte de los resultados se relaciona la cantidad de los datos registrados respeto del tiempo total de registro, para ello se aplica la ecuación Ec. (33).

$$m = \frac{Y_2 - Y_1}{X_2 - X_1} \qquad (33)$$

Donde:

m = pendiente

Y = variable dependiente, cantidad de datos.

X = variable independiente, tiempo.

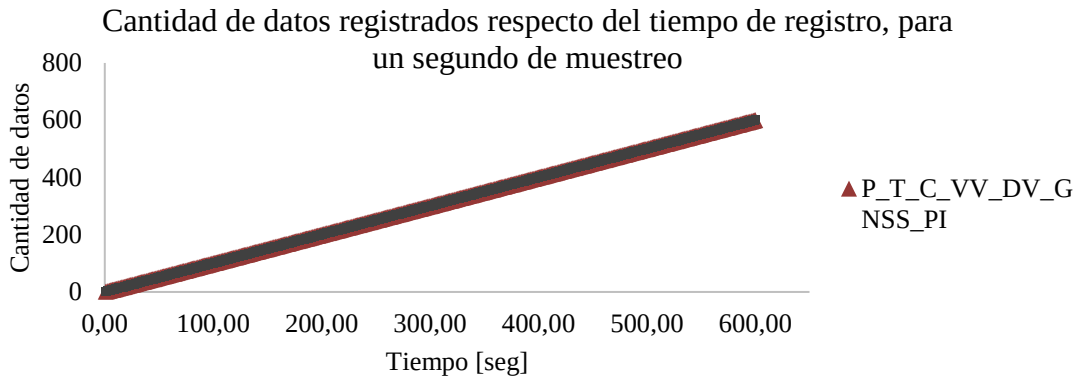


Ilustración 4-1: Cantidad de datos almacenados respecto al tiempo registro de datos acumulado, para un segundo de muestreo

Realizado por: González A. y Ortiz W., 2023.

Para el intervalo de muestreo de un segundo, de acuerdo a la Ilustración , se determinó 600 datos en 600 segundos de registro, dando como resultado 1 dato/segundo.

$$m_{1\ seg} = \frac{dato_2 - dato_1}{tiempo_2 - tiempo_1}$$

$$m_{1\ seg} = \frac{(600-1)[dato]}{(600-1)[segundo]}$$

$$m_{1\ seg} = 1\, dato/segundo$$

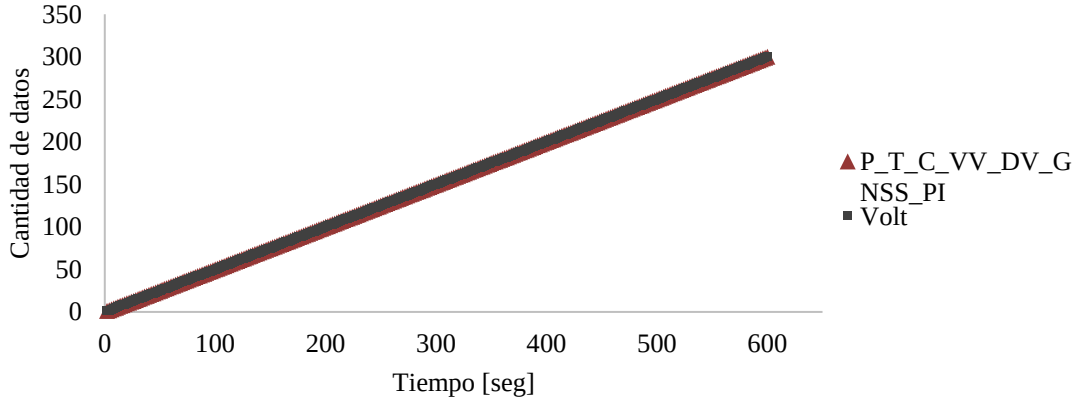


Ilustración 4-2: Cantidad de datos almacenados respecto al tiempo registro de datos acumulado, para dos segundos de muestreo.

Realizado por: González A. y Ortiz W., 2023.

Para el intervalo de muestreo de dos segundos, de acuerdo a la **¡Error! No se encuentra el origen de la referencia.**, se determinó 300 datos en 600 segundos de registro, dando como resultado 0.5 datos/segundo o 2 segundos/dato.

$$m_{2\ seg} = \frac{dato_2 - dato_1}{tiempo_2 - tiempo_1}$$

$$m_{2\ seg} = \frac{(300 - 1)[dato]}{(600 - 2)[segundo]}$$

$$m_{2\ seg} = 0.5\ dato/segundo$$

En la **¡Error! No se encuentra el origen de la referencia.** se presenta el resultado de la estructura d el registro de datos en el archivo de texto, en la cual se obtuvo una estructura ordenada que coincide con los valores obtenidos, esta ilustración se complementa con la Tabla .

Tabla 4-1: Resultado de mediciones y registro de parámetros seleccionados.

Resultado de sensores							
Nro.	**Parámetros seleccionados**	**Precisión**	**Rango de medición**		**Valor registrado** (revisar ANEXO O y ANEXO P)	**Unidad**	**Descripción**
			Min.	**Max.**			
1	Presión ambiente	±1.0	300	1100	732.77	hPa	Presión barométrica o absoluta
9	Altura	1.5 m	-	-	2842.763	m	Altura con respecto al nivel del mar
2	Temperatura ambiente	±0.5	-55	150	13.69	°C	Temperatura relativa del ambiente
3	Corriente de batería	1.30Amp @ ±75Amp 20Amp @ ±1000Amp	-1000	1000	19.32	A	Sensor de efecto hall, no invasivo
4	Voltaje de batería	0.5%	30	1000	93.84	V	Sensor por divisor de voltaje, requiere contacto físico en punto de prueba
5	Velocidad del viento	±1.0	0.5	89	2.1	m/s	Sensor de efecto hall
6	Dirección del viento	±3	0	360	225.60	°	Sensor de tipo potenciómetro
11	Puntos de interés	-	0	-	0	n/a	La identificación del punto de interés se determina por cambio de valor o interrupciones.
Resultado modulo GNSS							
Nro.	**Parámetros seleccionados**	**Precisión**			**Valor registrado** (revisar ANEXO O y ANEXO P)	**Unidad**	**Descripción**
7	Latitud	1.5 m			0139.207176	Formato: ddmm.mmmmmmm	Requiere de antena para inicio de registro de dato, y un tiempo de espera de 1 minuto.
8	Longitud	1.5 m			07840.5105923	Formato: ddmm.mmmmmmm	
10	Velocidad del vehículo	0.05			0.05	m/s	

Realizado por: González A. y Ortiz W., 2023.

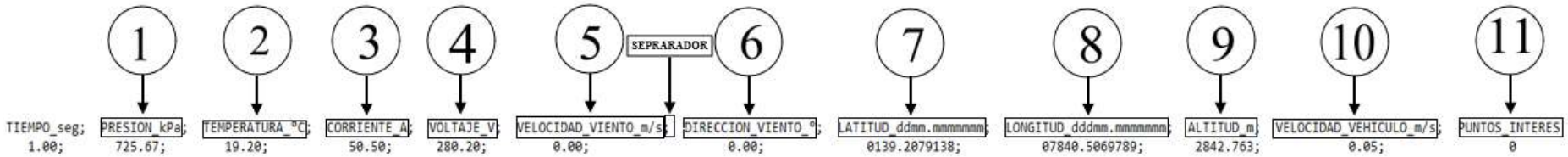


Ilustración 4-3: Estructura de registro de datos.

Realizado por: González A. y Ortiz W., 2023.

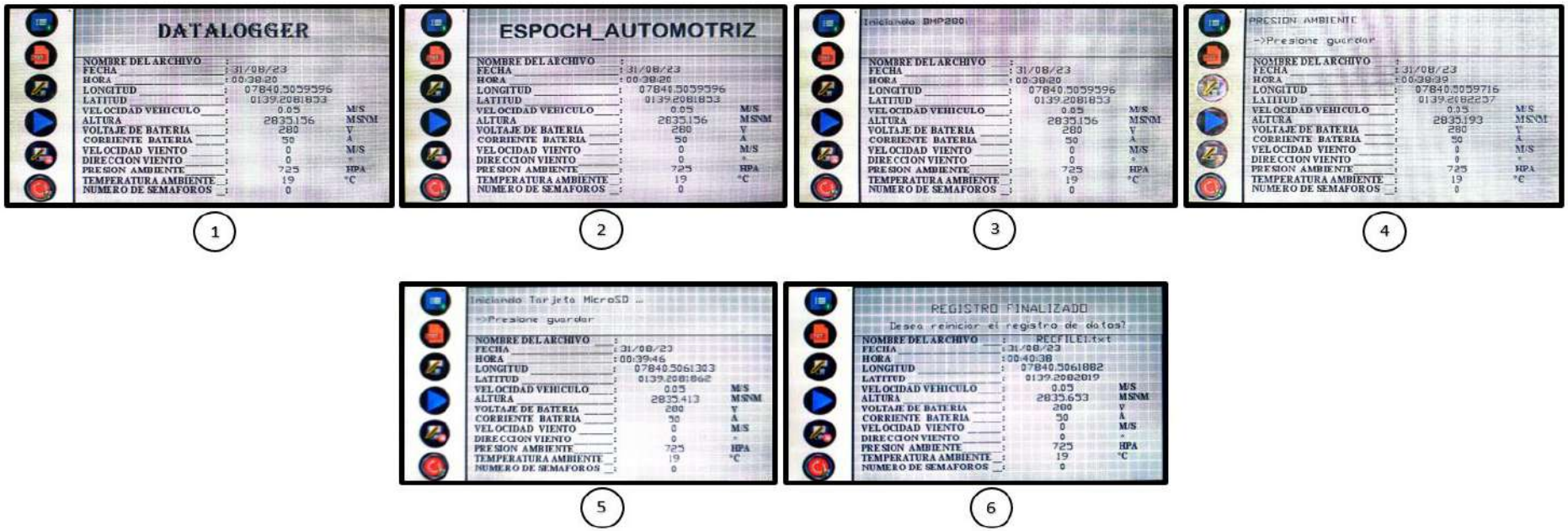


Ilustración 4-4: Secuencia de presentación de la información en la interfaz Gráfica.

Realizado por: González A. y Ortiz W., 2023

4.2. Resultados de la presentación de la información e interacción con la interfaz gráfica

La presentación de la información en la pantalla Nextion se muestra en la **¡Error! No se e ncuentra el origen de la referencia.**, se muestra la secuencia de la presentación de la información enviada desde el microcontrolador de la placa principal. Este resultado está de acuerdo a una secuencia y una lógica de interacción, la presentación de la información es fácil de comprender y manejar.

En la **¡Error! No se encuentra el origen de la referencia.**, en la parte de la inicialización de s ensor de BMP280 y de la tarjeta micro SD, se presenta un mensaje adicional para cada uno de estos dos componentes, los mensajes se relacionan con la conexión de estos dispositivos, para la tarjeta micro SD se presenta un mensaje adicional relacionado al acceso del archivo para el registro de datos. Cabe mencionar que el dispositivo tiene la capacidad de realizar registros en nuevos archivos.

En la **¡Error! No se encuentra el origen de la referencia.** se presentan las partes en las cual se d ividió la presentación de la información. La información fue estructurada de forma ordenada, en la parte A se encontrara la parte de interacción táctil en la cual se puede generar acciones de acuerdo a como se presione el botón táctil; en la parte B se visualizan las acciones tomadas para sensores, modulo GNSS y tiempo de registro; en la parte C se presentan las medidas de las magnitudes físicas y datos de posicionamiento en tiempo real; y en la parte D se presenta el nombre del archivo creado junto con un mensaje adicional que aparece en el caso de haber llegado al límite de archivos creados. En la **¡Error! No se encuentra el origen de la referencia.** se p resentan los botones para configurar el registro de datos.

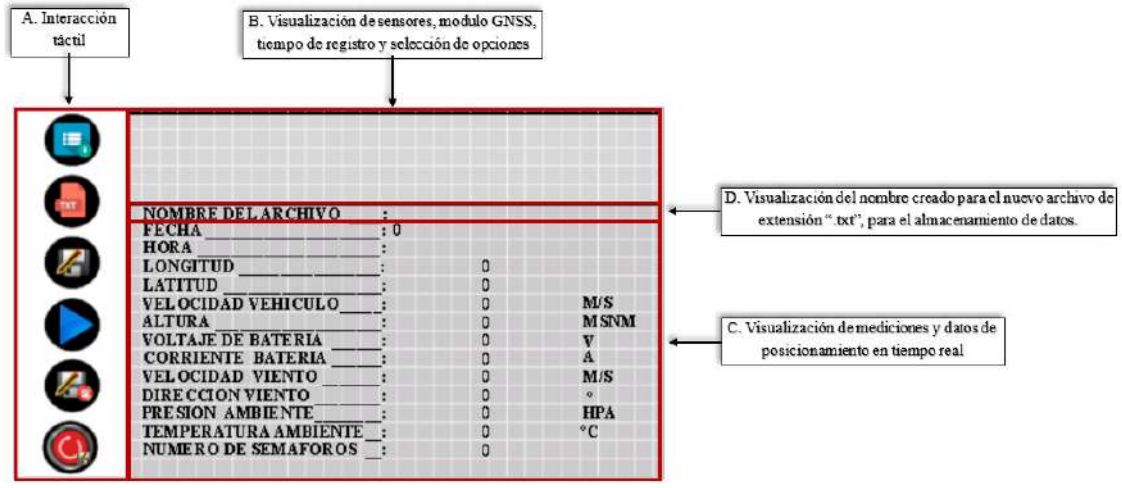


Ilustración 4-5: Partes de la presentación de la información en la interfaz gráfica

Realizado por: González A. y Ortiz W., 2023.

Ilustración 4-6: Botones para la interacción entre pantalla y usuario.

Realizado por: González A. y Ortiz W., 2023.

El datalogger permite crear hasta máximo nueve archivos de extensión ".txt". En el caso de haber llegado al límite de archivos creados se presenta un mensaje que avisa al usuario que se llegó al límite de archivos creados, este mensaje se presenta en la parte D de la **¡Error! No se encuentra e l origen de la referencia.**. Toda la información presentada en la interfaz gráfica y la selección del registro de datos y tiempo de registro por medio de los botones se relacionó con la información registrada en el archivo creado en la micro SD. Los menajes de aviso para el sensor de presión BMP280, tarjeta micro SD y creación de archivos no presentaron problemas, los mensajes se presentaron exactamente en la situación que deberían aparecer.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La recopilación de información bibliográfica permitió seleccionar sensores de: presión (BMP280), temperatura (LM35A), velocidad y dirección del viento (Davis Vantage Pro2 6410), corriente (DHAB S/137), voltaje (JDXDA4U) y módulo GNSS (NS-HP-GN2); considerando: modelos matemáticos de consumo energético en vehículos eléctricos, fundamentos relacionados al sistema electrónico de medida, principio de funcionamiento de transductores, tipo de señal, protocolos de comunicación y especificaciones de funcionamiento; permitiendo proyectar un modelo capaz de obtener datos en tiempo real.

En el diseño del circuito se consideraron los pines de entrada para señales analógicas y digitales de los transductores elegidos. Se tomó en cuenta tres tipos de puertos de comunicación: un puerto SPI, un puerto I2C y tres puertos UART. Adicionalmente se incorporaron pines para: alimentación e interruptores.

El algoritmo se desarrolló en lenguaje C, debido a la compatibilidad de las placas de microcontrolador Arduino con este lenguaje, incluyendo librerías para el manejo de módulos y algunos sensores. Para los sensores analógicos como: temperatura, corriente, voltaje, y dirección de viento; se consideró la resolución del microcontrolador para el cálculo de las magnitudes físicas. Para los sensores digitales como: presión y velocidad del viento, se aplicó una lectura digital. El registro de datos en la memoria micro SD se realiza por protocolo SPI, la transmisión de datos entre las placas de microcontrolador, pantalla Nextion INX4827K043_011R y módulo GNSS se realiza por protocolo UART y el BMP280 utiliza el protocolo I2C.

Para el desarrollo de la PCB se tomó en cuenta el diseño de los siete circuitos electrónicos. Se consideraron los pines utilizados de las placas de microcontrolador y conexiones: transductores, interfaz gráfica, módulos, alimentación e interruptores; se tomó en cuenta el tamaño de cada componente que se conecta de forma directa con la PCB, y con ello se dimensionó la caja que soporta la PCB.

Durante las pruebas realizadas en el registro de datos del vehículo se verificó que el datalogger funcionara en una secuencia ordenada durante la presentación de la información, creación del archivo de formato “.txt” y selección de los parámetros a registrar. El dispositivo fue capaz de

crear un archivo de extensión “.txt” para almacenar las mediciones y posicionamiento; permitió seleccionar el tiempo de registro de 1 o 2 segundos; en el archivo de texto los datos se almacenan de forma ordenada y estructurada lo cual hace que sea posible importar el archivo de texto a una hoja cálculo de Excel. Y se comprobó que los datos de las magnitudes físicas no se pierdan y se almacenen completamente.

5.2. Recomendaciones

La información adquirida mediante el datalogger puede ser útil para futuros trabajos relacionados a ciclos de conducción para estimar el consumo de energía en un vehículo eléctrico.

Seguir el manual presentado en el ANEXO Q para la manipulación del datalogger y evitar problemas relacionados al almacenamiento de las magnitudes físicas.

No manipular los elementos internos del datalogger, con el fin de evitar problemas de funcionalidad.

Evitar rociar los sensores y el datalogger con agua o con otros líquidos ya que al poseer en su estructura partes metálicas estas se oxidarían siendo perjudicial para su funcionamiento.

Realizar pruebas de medición y registro de datos con otro datalogger similar para confirmar la funcionalidad del dispositivo, esto debido a que los resultados se basaron en la precisión de los microcontroladores, sensores y módulos, de acuerdo a las especificaciones de los fabricantes.

BIBLIOGRAFÍA

1. **AEADE,** *Anuario* [en línea]. 2022. Quito: Gráficas Iberia. Disponible en: https://www.aeade.net/wp-content/uploads/2023/03/ANUARIO-AEADE_2022_comp.pdf.

2. **ALIEXPRESS,** Sensor de voltaje JXDA4U DC 10mV/20mV/75mV/1V/5V/10V/50V/100V/300V/500V/1000V/1500V/V transductor - AliExpress. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://es.aliexpress.com/i/32959852858.html.

3. **ARDUINO,** Arduino Mega 2560 Rev3 — Arduino Official Store. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://store.arduino.cc/products/arduino-mega-2560-rev3.

4. **BERNÉ, J., GARRIDO, N. & CAPILLA, R.,** *GNSS: GPS, Galileo, Glonass, Beidou: fundamentos y métodos de posicionamiento* [en línea]. Valencia: Editorial de la Universidad de Valencia. ISBN 9788490487785. Disponible en: https://elibro.net/es/ereader/espoch/111750.

5. **CENGEL, Y. & BOLES, M.,** *Termodinámica*. Séptima. México: McGrawHill.

6. **CORONA, L. & ABARCA, G.,** *Sensores y actuadores: aplicaciones con arduino* [en línea]. Ciudad de México: Grupo Editorial Patria. ISBN 9786075501222. Disponible en: https://elibro.net/es/ereader/espoch/121284.

7. **DAVIS,** Estacionesdavis.es. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.estacionesdavis.es/es/sensores/39-anemometro-para-vantage-pro2.html.

8. **DIGIZONE,** Modulos archivos | Digizone. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://digizone.com.ve/categoria-producto/modulos-y-boards/modulos/.

9. **DOGAN, I.,** *Programación de microcontroladores PIC* [en línea]. Barcelona: MARCOMBO. ISBN 9781449209599. Disponible en: https://elibro.net/es/ereader/espoch/45918.

10. **FIORI, C., AHN, K. & RAKHA, H.,** Power-based electric vehicle energy consumption model: Model development and validation. *Applied Energy* [en línea], vol. 168, ISSN

03062619. DOI 10.1016/j.apenergy.2016.01.097. Disponible en: http://dx.doi.org/10.1016/j.apenergy.2016.01.097.

11. **GAIMC,** LM35 Temperature Sensor- GAIMC. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.gaimc.com/products/LM35-temperature-sensor/LM35_Temperature_Sensor.html?gclid=Cj0KCQjwzdOlBhCNARIsAPMwjbw2pzfEcwoinPtjR58sOriXIT_uBDLqwoEWKc2xFjhQQh4q2C0sOngaAilGEALw_wcB.

12. **GAO, Z., LIN, Z., LACLAIR, T.J., LIU, C., LI, J.M., BIRKY, A.K. & WARD, J.,** Battery capacity and recharging needs for electric buses in city transit service. *Energy* [en línea], vol. 122, ISSN 03605442. DOI 10.1016/j.energy.2017.01.101. Disponible en: http://dx.doi.org/10.1016/j.energy.2017.01.101.

13. **GIL, S. & PRIETO, R.,** Los autos eléctricos: ¿hacia un transporte más sustentable? *2015 E-Health and Bioengineering Conference, EHB 2015* [en línea], DOI 10.1109/EHB.2015.7391473. Disponible en: https://www.fisicarecreativa.com/papers_sg/papers_sgil/Gas/AutorElectricos_Petrot_2013.pdf.

14. **GILLESPIE, T.D.,** *Fundamentals of Vehicle Dynamics*. Warrendale: Society of Automotive Engineers.

15. **GRANDA, M. & MEDIAVILLA, E.,** *Instrumentación electrónica: transductores y acondicionadores de señal* [en línea]. Santander: Universidad de Cantabria. ISBN 9788481027471. Disponible en: https://elibro.net/es/ereader/espoch/53391.

16. **GUNSHA, F.,** *Análisis comparativo del desempeño de controladores discretos PID y difuso evaluados en un sistema de transferencia de calor* [en línea]. Guayaquil: ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL. [consulta: 7 agosto 2023]. Disponible en: http://www.dspace.espol.edu.ec/handle/123456789/52728.

17. **HERRERA, J., ILIZALITURRI, I. & MORALES, M.,** Unidad Básica de Comunicación Serial en un Microcontrolador. *Polibits* [en línea], vol. 33, ISSN 1870-9044. DOI 10.17562/pb-33-1. Disponible en: https://www.redalyc.org/pdf/4026/402640446001.pdf.

18. **HINICIO,** Estrategia Nacional de Electromovilidad para el Ecuador. [en línea]. S.l.: [consulta: 2 febrero 2023]. Disponible en: https://varusecuador.com/wp-content/uploads/2021/05/Estrategia_Nacional_de_Electromovilidad_Ecuador.pdf.

19. **INDIAMART,** Arduino Pro Mini at Rs 200/piece | Girgaon | Mumbai | ID: 7339107562. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.indiamart.com/proddetail/arduino-pro-mini-7339107562.html.

20. **LEM,** Principle of DHAB family. [en línea]. S.l.: Disponible en: www.lem.com.

21. MICROLOG, 2023. pulsador para circuito impreso. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.micro-log.com/interruptor-conmutador/2437-533-pulsador-para-ci.html#/27-cantidad-1_unidad.

22. **MORENO, A. & CÓRCOLES, S.** *Arduinio. Curso Práctico* [en línea]. S.l.: RA-MA Editorial. ISBN 9788499647821. Disponible en: https://elibro.net/es/ereader/espoch/106517.

23. **MOUSER,** BMP280 Herramientas de desarrollo de sensores – Mouser Ecuador. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.mouser.ec/c/?tool%20is%20for%20evaluation%20of=BMP280.

24. **NAVSPARK,** NavSpark : Arduino Compatible Development Board with GPS / GNSS. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://navspark.mybigcommerce.com/.

25. **NEXTION,** Amazon.com: Nextion Mejorado 4.3 pulgadas NX4827K043 Pantalla táctil resistiva HMI LCD 480x272 para Arduino Raspberry Pi : Electrónica. [en línea]. [consulta: 16 julio 2023]. Disponible en: https://www.amazon.com/-/es/Mejorado-pulgadas-NX4827K043-resistiva-Raspberry/dp/B07BL3D5DW.

26. **PAFFUMI, E., DE GENNARO, M., MARTINI, G., MANFREDI, U., VIANELLI, S., ORTENZI, F. & GENOVESE, A.,** Experimental Test Campaign on a Battery Electric Vehicle: On-Road Test Results (Part 2). *SAE International Journal of Alternative Powertrains*, vol. 4, no. 2, ISSN 21674205. DOI 10.4271/2015-01-1166.

27. **PÉREZ, R. & GONZÁLEZ, O.,** *Prototipo de adquisición de señales biológicas utilizando Arduino* [en línea]. Ciudad de la Habana: La Editorial Universitaria. ISBN 9789591631916. Disponible en: https://elibro.net/es/ereader/espoch/71660.

28. **RAMOS, A., RAMOS, J. & VIÑAS, S.,** Dispositivos de almacenamiento. En: MCGRAW-HILL INTERAMERICANA DE ESPAÑA S.L. (ed.), *CEO - Montaje y mantenimiento de equipos GM* [en línea]. 2da. Madrid: s.n., pp. 90-101. [consulta: 10 mayo 2023]. vol. 2ed. ISBN 84-481-7078-4. Disponible en: https://www.mheducation.es/bcv/guide/capitulo/8448180364.pdf.

29. **RODRÍGUEZ, J. & VILLAR, J.,** *Sistemas de transmisión y fenado*. Primera. S.l.: MACMILLAN. ISBN 84-16653-88-7.

30. **SUPLEMENTO REGISTRO OFICIAL, N.,** *Ley Orgánica de eficiencia energética* [en línea]. 19 marzo 2019. S.l.: Editora Nacional. [consulta: 26 abril 2023]. 449. Disponible en: www.registroficial.gob.ec.

ANEXOS

ANEXO A: ESPECIFICACIONES SENSOR DE PRESIÓN BMP280

Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature range	T_A	operational	-40	25	+85	°C
		full accuracy	0		+65	
Operating pressure range	P	full accuracy	300		1100	hPa
Sensor supply voltage	V_{DD}	ripple max. 50mVpp	1.71	1.8	3.6	V
Interface supply voltage	V_{DDIO}		1.2	1.8	3.6	V
Supply current	$I_{DD,LP}$	1 Hz forced mode, pressure and temperature, lowest power		2.8	4.2	µA
Peak current	I_{peak}	during pressure measurement		720	1120	µA
Current at temperature measurement	I_{DDT}			325		µA
Sleep current[1]	I_{DDSL}	25 °C		0.1	0.3	µA
Standby current (inactive period of normal mode) [2]	I_{DDSB}	25 °C		0.2	0.5	µA
Relative accuracy pressure V_{DD} = 3.3V	A_{rel}	700 ... 900hPa		±0.12		hPa
		25 . . . 40 °C		±1.0		m
Offset temperature coefficient	TCO	900hPa		±1.5		Pa/K
		25 . . . 40 °C		12.6		cm/K
Absolute accuracy pressure	A^P_{ext}	300. . 1100 hPa -20 . . . 0 °C		±1.7		hPa
	A^P_{full}	300 . . . 1100 hPa 0 . . . 65 °C		±1.0		hPa
Resolution of output data in ultra high resolution mode	R^P	Pressure		0.0016		hPa
	R^T	Temperature		0.01		°C
Noise in pressure	$V_{p,full}$	Full bandwidth, ultra high resolution See chapter 3.5		1.3		Pa
				11		cm
	$V_{p,filtered}$	Lowest bandwidth, ultra high resolution See chapter 3.5		0.2		Pa
				1.7		cm
Absolute accuracy temperature[3]	A^T	@ 25 °C		±0.5		°C
		0 . . . +65 °C		±1.0		°C
PSRR (DC)	PSRR	full V_{DD} range			±0.005	Pa/mV
Long term stability[4]	ΔP_{stab}	12 months		±1.0		hPa
Solder drifts		Minimum solder height 50 µm	-0.5		+2	hPa
Start-up time	$t_{startup}$	Time to first communication after both V_{DD} > 1.58V and V_{DDIO} > 0.65V			2	ms
Possible sampling rate	f_{sample}	*osrs_t* = *osrs_p* = 1; See chapter 3.8	157	182	tbd[5]	Hz
Standby time accuracy	$\Delta t_{standby}$			±5	±25	%

ANEXO B: ESPECIFICACIONES SENSOR DE TEMPERATURA LM35A

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60˚C to +180˚C
TO-92 Package,	−60˚C to +150˚C
SO-8 Package,	−65˚C to +150˚C
TO-220 Package,	−65˚C to +150˚C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300˚C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260˚C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215˚C
Infrared (15 seconds)	220˚C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55˚C to +150˚C
LM35C, LM35CA	−40˚C to +110˚C
LM35D	0˚C to +100˚C

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T_A=+25˚C	±0.2	±0.5		±0.2	±0.5		˚C
	T_A=−10˚C	±0.3			±0.3		±1.0	˚C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		˚C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	˚C
Nonlinearity (Note 8)	T_{MIN}≤T_A≤T_{MAX}	**±0.18**		**±0.35**	**±0.15**		**±0.3**	˚C
Sensor Gain (Average Slope)	T_{MIN}≤T_A≤T_{MAX}	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/˚C
Load Regulation (Note 3) 0≤I_L≤1 mA	T_A=+25˚C	±0.4	±1.0		±0.4	±1.0		mV/mA
	T_{MIN}≤T_A≤T_{MAX}	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25˚C	±0.01	±0.05		±0.01	±0.05		mV/V
	4V≤V_S≤30V	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25˚C	56	67		56	67		µA
	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25˚C	56.2	68		56.2	68		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	4V≤V_S≤30V, +25˚C	0.2	1.0		0.2	1.0		µA
	4V≤V_S≤30V	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/˚C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	˚C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			˚C

ANEXO C: ESPECIFICACIONES DEL ANEMOMETRO VANTAGE PRO2 6410

Sensor Output

Wind Direction

Display Resolution	16 points (22.5°) on compass rose, 1° in numeric display
Accuracy	±3°

Wind Speed

Resolution and Units	Measured in 1 mph. Other units are converted from mph and rounded to nearest 1 km/h, 0.1 m/s, or 1 knot
Range	1 to 200 mph, 1 to 173 knots, 0.5 to 89 m/s, 1 to 322 km/h
Accuracy	±2 mph (2 kts, 3 km/h, 1 m/s) or ±5%, whichever is greater
Maximum Cable Length	240' (73 m). Maximum wind speed reading decreases as length of cable from Anemometer to ISS increases. At 140' (42 m), maximum speed is 135 mph (60 m/s). At 240', the maximum is 100 mph.

Input/Output Connections

Black	Wind speed open drain to ground
Red	Ground
Green	Wind direction pot wiper (20KΩ potentiometer)
Yellow	Pot supply voltage
Wind Speed Translation Formula	1600 rev/hr = 1 mph V = P(2.25/T) *(V = speed in mph, P = no. of pulses per sample period T = sample period in seconds)*
Wind Direction Translation	Variable resistance 0 - 20KΩ; 10KΩ = south, 180°

ANEXO D: ESPECIFICACIONES DEL SENSOR DE CORRIENTE DHAB S/137

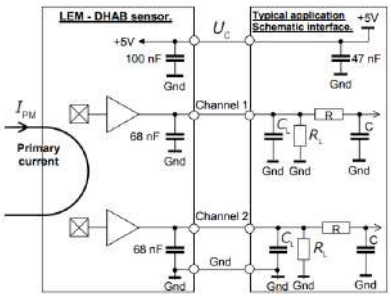
Parameter	Symbol	Unit	Specification			Conditions
			Min	Typical	Max	
Electrical Data						
Supply voltage [1]	U_C	V	4.75	5	5.25	
Current consumption	I_C	mA		15	20	
Maximum output current	I_{out}	mA	−1		1	
Load resistance	R_L	KΩ	10			
Capacitive loading	C_L	nF	1		100	
Ambient operating temperature	T_A	°C	−10		65	High accuracy
			−40		125	Reduced accuracy
Performance Data channel 1						
Primary current, measuring range	$I_{PM\ channel\ 1}$	A	−75		75	
Primary nominal DC or rms current	$I_{PN\ channel\ 1}$	A	−75		75	@ T_A = 25 °C
Offset voltage	V_O	V		2.5		@ U_C = 5 V
Sensitivity	G	mV/A		26.67		@ U_C = 5 V
Resolution		mV		2.5		@ U_C = 5 V
Output clamping voltage min [1]	V_{SZ}	V	0.2	0.25	0.3	@ U_C = 5 V
Output clamping voltage max [1]		V	4.7	4.75	4.8	@ U_C = 5 V
Output internal resistance	R_{out}	Ω		1	10	
Frequency bandwidth [2]	BW	Hz		70		@ −3 dB
Power up time		ms			1	
Setting time after overload		ms			10	
Ratiometricity error	ε_r	%	−0.6		0.6	
Output voltage noise peak-peak	$V_{no\ pp}$	mV	−10		10	
Performance Data channel 2						
Primary current, measuring range	$I_{PM\ channel\ 2}$	A	−1000		1000	
Primary nominal DC or rms current	$I_{PN\ channel\ 2}$	A	−1000		1000	@ T_A = 25 °C
Offset voltage	V_O	V		2.5		@ U_C = 5 V
Sensitivity	G	mV/A		2		@ U_C = 5 V
Resolution		mV		2.5		@ U_C = 5 V
Output clamping voltage min [1]	V_{SZ}	V	0.2	0.25	0.3	@ U_C = 5 V
Output clamping voltage max [1]		V	4.7	4.75	4.8	@ U_C = 5 V
Output internal resistance	R_{out}	Ω		1	10	
Frequency bandwidth [2]	BW	Hz		70		@ −3 dB
Power up time		ms			1	
Setting time after overload		ms			10	
Ratiometricity error	ε_r	%	−0.6		0.6	
Output voltage noise peak-peak	$V_{no\ pp}$	mV	−10		10	

Notes: [1] The output voltage V_{out} is fully ratiometric. The offset and sensitivity are dependent on the supply voltage U_C relative to the following formula:

$$I_P = \left(\frac{5}{U_C} \times V_{out} - V_O\right) \times \frac{1}{G} \text{ with G in (V/A)}$$

[2] Primary current frequencies must be limited in order to avoid excessive heating of the busbar, magnetic core and the ASIC (see feature paragraph in page 1.)

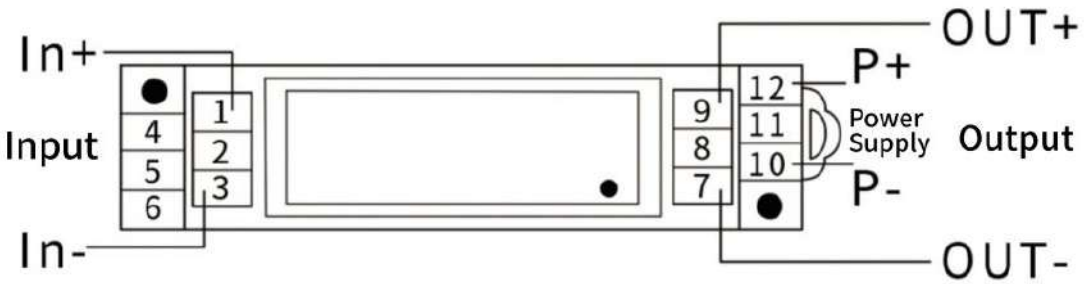
System architecture (example)



R_L > 10 kΩ optional resistor for signal line diagnostic
C_L < 100 nF EMC protection
RC Low pass filter EMC protection (optional)

ANEXO E: ESPECIFICACIONES DEL SENSOR DE VOLATJE JXDA4U

Measuring Range:	AC/DC 0-2000V (Customized)
Output Signal :	4-20mA / 0-10V / 0-5V / 1-5V/ 0-10mA / 0-20mA
Power Supply:	DC12V / DC15V / DC12-28V / AC220V (Customized)
Installation:	35mm DIN Rail
Load capacity :	Voltage output:5mA; current output:6V
Linearity :	0.1%
Accuracy:	0.2%; 0.5%
Overload capacity :	Double nominal input,sustainable
Temperature drift:	⩽500PPM/°C
Band width :	DC
Consumption of current :	<25mA+output current
Working temperature:	-10°C- +70°C
Storage temperature:	-25°C~+85°C
Response time :	<200mS
Isolation pressure resistance :	1kV;3kV/50Hz,1Min
Offset Voltage:	⩽10mV



ANEXO F: ESPECIFICACIONES DEL RECEPTOR GNSS PX1122R

TECHNICAL SPECIFICATIONS

Receiver Type	230 channel Phoenix GNSS engine GPS/QZSS L1/L2C, BeiDou B1I/B2I, Galileo E1/E5b, GLONASS L1OF/L2OF		
Accuracy	Position	1.5m CEP	autonomous mode
		1cm + 1ppm	RTK mode
	Velocity	0.05m/sec*1	
	Time	12ns	
	Moving Base Heading	0.13 degree*2	
Time to First Fix	1 second hot-start under open sky (average) 28 second warm-start under open sky (average) 29 second cold-start under open sky (average)		
RTK Convergence	< 10sec		
Reacquisition	1s		
Update Rate	RTK 1 / 2 / 4 / 5 / 8 / 10 Hz Raw Measurement 1 / 2 / 4 / 5 / 8 / 10 / 20 Hz Moving Base RTK and Advance Moving Base RTK1 / 2 / 4 / 5 / 8 Hz RTK: for precise positioning. Moving-Base (MB) RTK for precise heading. Advanced Moving Base (AMB) RTK for precise positioning & heading.		
Operational Limits	Altitude < 80,000m and velocity < 515m/s		
Serial Interface	3.3V LVTTL level		
Protocol	NMEA-0183 V4.1 GGA, GLL, GSA, GSV, RMC, VTG 115200 baud, 8, N, 1 RTCM 3.x or SkyTraq raw data binary 115200 baud, 8, N, 1		
Datum	Default WGS-84 and user definable in stand-alone mode Depends on base reference frame when in RTK mode		
Input Voltage	3.3V DC +/-10%		
Current Consumption	100mA		
Dimension	16.0mm L x 12.2mm W x 2.9mm H		
Weight:	1.7g		
Operating Temperature	-40°C ~ +85°C		
Storage Temperature	-55°C ~ +100°C		
Humidity	5% ~ 95% non-condensing		

Table 2: Overview of SkyTraq receiver's NMEA messages

$GPGGA	Time, position, and fix related data of the receiver.
$GNGLL	Position, time and fix status.
$GNGSA	Used to represent the ID's of satellites which are used for position fix. When GPS satellites are used for position fix, $GNGSA sentence is output with system ID 1. When GLONASS satellites are used for position fix, $GNGSA sentence is output with system ID 2. When Galileo satellites are used for position fix, $GNGSA sentence is output with system ID 3. When BDS satellites are used for position fix, $GNGSA sentence is output with system ID 4.
$GPGSV **$GLGSV** **$GAGSV** **$GBGSV**	Satellite information about elevation, azimuth and CNR, $GPGSV is used for GPS satellites, $GLGSV is used for GLONASS satellites, $GAGSV is used for GALILEO satellites, while $GBGSV is used for BDS satellites
$GNRMC	Time, date, position, course and speed data.
$GNVTG	Course and speed relative to the ground.
$GNZDA	UTC, day, month and year and time zone.
$GNTHS	True Heading and Status.

ANEXO G: ESPECIFICACIONES DE LA ANTENA PARA EL MODULO RECEPTOR GNSS

DESCRIPCIÓN

High precision antenna covering L1 GPS/GLONASS, E1 Galileo, B1 Beidou frequency bands.

PATCH ANTENNA SPEC

Frequency Range :	1558MHz ~ 1615MHz
Polarization :	RHCP
Gain :	5dBic @ PCB (Zenith)
VSWR :	< 1.5
Dimension :	33.5mm x 33.5mm
Axis Ratio :	3dB (max)

LNA SPEC

Noise Figure :	< 1.5dB
Gain :	28dB +/- 2dB
Supply Voltage :	3V ~ 5V DC
Current Consumption :	< 20mA
VSWR :	< 2

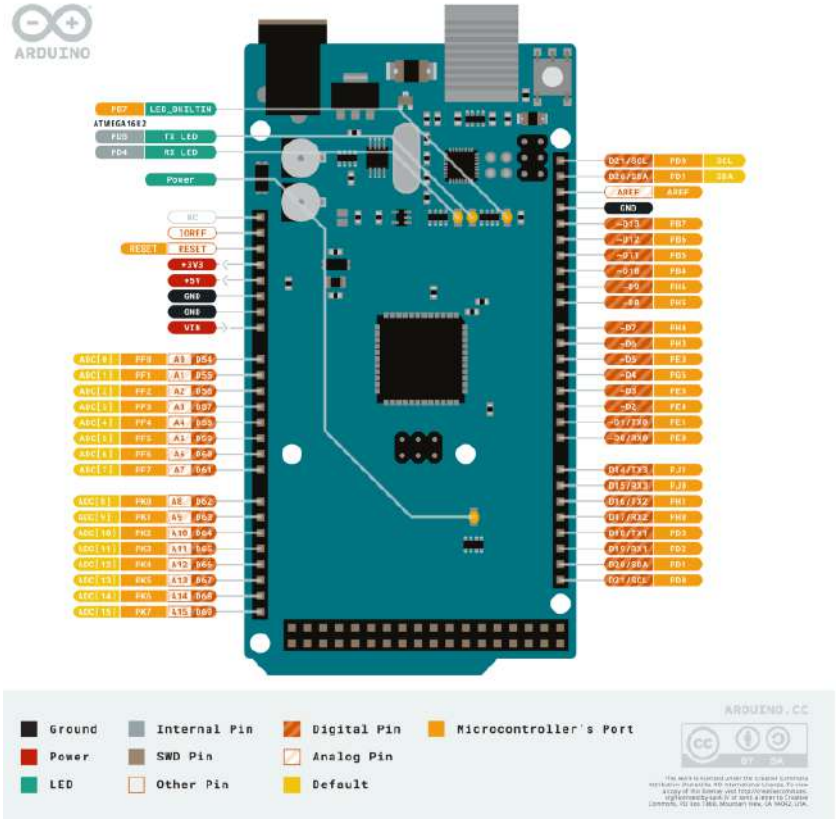
MECHANICAL SPEC

Connector :	SMA
RF Cable :	RG174, 3m

ANEXO H: ESPECIFICACIONES DE LA PLACA ARDUINO MEGA 2560

Features

- **ATmega2560 Processor**
  - Up to 16 MIPS Throughput at 16MHz
  - 256k bytes (of which 8k is used for the bootloader)
  - 4k bytes EEPROM
  - 8k bytes Internal SRAM
  - 32 × 8 General Purpose Working Registers
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Four Programmable Serial USART
  - Controller/Peripheral SPI Serial Interface
- **ATmega16U2**
  - Up to 16 MIPS Throughput at 16 MHz
  - 16k bytes ISP Flash Memory
  - 512 bytes EEPROM
  - 512 bytes SRAM
  - USART with SPI master only mode and hardware flow control (RTS/CTS)
  - Master/Slave SPI Serial Interface
- **Sleep Modes**
  - Idle
  - ADC Noise Reduction
  - Power-save
  - Power-down
  - Standby
  - Extended Standby
- **Power**
  - USB Connection
  - External AC/DC Adapter
- **I/O**
  - 54 Digital
  - 16 Analog
  - 15 PWM Output



5.1 Analog

Pin	Function	Type	Description
1	NC	NC	Not Connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog	Analog input 0 /GPIO
10	A1	Analog	Analog input 1 /GPIO
11	A2	Analog	Analog input 2 /GPIO
12	A3	Analog	Analog input 3 /GPIO
13	A4	Analog	Analog input 4 /GPIO
14	A5	Analog	Analog input 5 /GPIO
15	A6	Analog	Analog input 6 /GPIO
16	A7	Analog	Analog input 7 /GPIO
17	A8	Analog	Analog input 8 /GPIO
18	A9	Analog	Analog input 9 /GPIO
19	A10	Analog	Analog input 10 /GPIO
20	A11	Analog	Analog input 11 /GPIO
21	A12	Analog	Analog input 12 /GPIO
22	A13	Analog	Analog input 13 /GPIO
23	A14	Analog	Analog input 14 /GPIO
24	A15	Analog	Analog input 15 /GPIO

5.2 Digital

Pin	Function	Type	Description
1	D21/SCL	Digital Input/I2C	Digital input 21/I2C Dataline
2	D20/SDA	Digital Input/I2C	Digital input 20/I2C Dataline
3	AREF	Digital	Analog Reference Voltage
4	GND	Power	Ground
5	D13	Digital/GPIO	Digital input 13/GPIO
6	D12	Digital/GPIO	Digital input 12/GPIO
7	D11	Digital/GPIO	Digital input 11/GPIO
8	D10	Digital/GPIO	Digital input 10/GPIO
9	D9	Digital/GPIO	Digital input 9/GPIO
10	D8	Digital/GPIO	Digital input 8/GPIO
11	D7	Digital/GPIO	Digital input 7/GPIO
12	D6	Digital/GPIO	Digital input 6/GPIO
13	D5	Digital/GPIO	Digital input 5/GPIO
14	D4	Digital/GPIO	Digital input 4/GPIO

Pin	Function	Type	Description
15	D3	Digital/GPIO	Digital input 3/GPIO
16	D2	Digital/GPIO	Digital input 2/GPIO
17	D1/TX0	Digital/GPIO	Digital input 1 /GPIO
18	D0/Tx1	Digital/GPIO	Digital input 0 /GPIO
19	D14	Digital/GPIO	Digital input 14 /GPIO
20	D15	Digital/GPIO	Digital input 15 /GPIO
21	D16	Digital/GPIO	Digital input 16 /GPIO
22	D17	Digital/GPIO	Digital input 17 /GPIO
23	D18	Digital/GPIO	Digital input 18 /GPIO
24	D19	Digital/GPIO	Digital input 19 /GPIO
25	D20	Digital/GPIO	Digital input 20 /GPIO
26	D21	Digital/GPIO	Digital input 21 /GPIO

5.5 Digital Pins D22 - D53 LHS

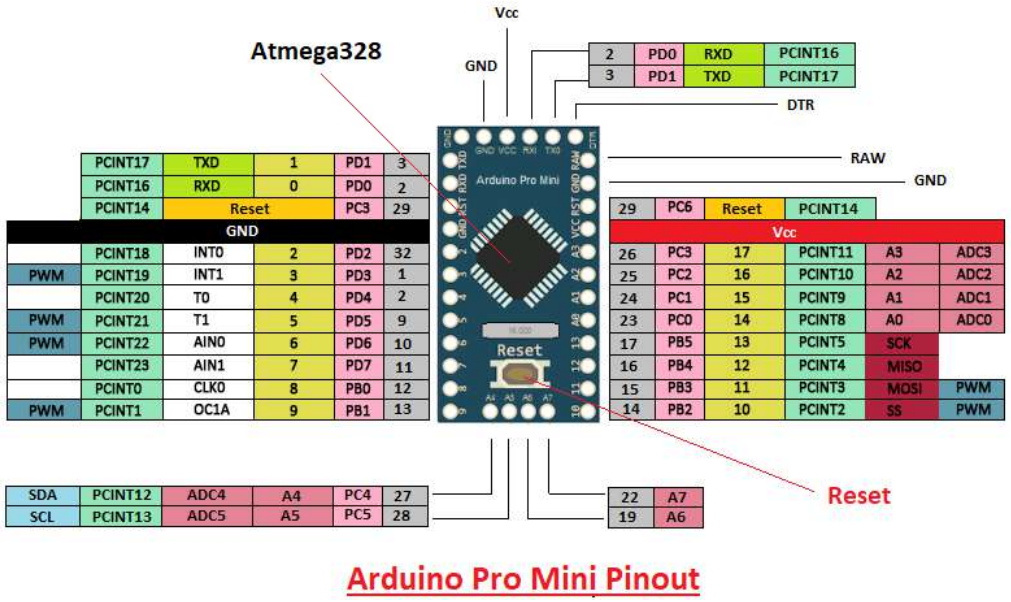
Pin	Function	Type	Description
1	+5V	Power	Power Supply of 5V
2	D22	Digital	Digital input 22/GPIO
3	D24	Digital	Digital input 24/GPIO
4	D26	Digital	Digital input 26/GPIO
5	D28	Digital	Digital input 28/GPIO
6	D30	Digital	Digital input 30/GPIO
7	D32	Digital	Digital input 32/GPIO
8	D34	Digital	Digital input 34/GPIO
9	D36	Digital	Digital input 36/GPIO
10	D38	Digital	Digital input 38/GPIO
11	D40	Digital	Digital input 40/GPIO
12	D42	Digital	Digital input 42/GPIO
13	D44	Digital	Digital input 44/GPIO
14	D46	Digital	Digital input 46/GPIO
15	D48	Digital	Digital input 48/GPIO
16	D50	Digital	Digital input 50/GPIO
17	D52	Digital	Digital input 52/GPIO
18	GND	Power	Ground

5.6 Digital Pins D22 - D53 RHS

Pin	Function	Type	Description
1	+5V	Power	Power Supply of 5V
2	D23	Digital	Digital input 23/GPIO
3	D25	Digital	Digital input 25/GPIO
4	D27	Digital	Digital input 27/GPIO
5	D29	Digital	Digital input 29/GPIO
6	D31	Digital	Digital input 31/GPIO
7	D33	Digital	Digital input 33/GPIO
8	D35	Digital	Digital input 35/GPIO
9	D37	Digital	Digital input 37/GPIO
10	D39	Digital	Digital input 39/GPIO
11	D41	Digital	Digital input 41/GPIO
12	D43	Digital	Digital input 43/GPIO
13	D45	Digital	Digital input 45/GPIO
14	D47	Digital	Digital input 47/GPIO
15	D49	Digital	Digital input 49/GPIO
16	D51	Digital	Digital input 51/GPIO
17	D53	Digital	Digital input 53/GPIO
18	GND	Power	Ground

ANEXO I: ESPECIFICACIONES DE LA PLACA ARDUINO PRO MINI

Microcontrolador	ATmega328P *
Fuente de alimentación de placa	3,35 -12 V (modelo de 3,3 V) o 5 - 12 V (modelo de 5 V)
Voltaje de funcionamiento del circuito	3.3V o 5V (según el modelo)
Pines de E/S digitales	14
Pines PWM	6
UART	1
SPI	1
I2C	1
Pines de entrada analógica	6
Interrupciones externas	2
Corriente continua por pin de E/S	40 mA
Memoria flash	32KB de los cuales 2 KB utilizados por el gestor de arranque *
SRAM	2 KB *
EEPROM	1 KB *
Velocidad de reloj	8 MHz (versiones de 3,3 V) o 16 MHz (versiones de 5 V)



https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-pro-mini.html

ANEXO J: ESPECIFICACIONES DE LA PANTALLA NEXTION lNX4827K043_011R

Modelos Nextion

Tipo de Nextion	Serie mejorada
Modelos Nextion	NX4827K043_011N (N: Sin contacto)
	NX4827K043_011R (R: Pantalla táctil resistiva)

Características técnicas

	Datos	Descripción
Color	64K 65536 colores	16 bits 565, 5R-6G-5B
Tamaño del diseño	120(L)×74(W)×5(H)	NX4827K043_011N
	120(L)×74(W)×6.2(H)	NX4827K043_011R
Área Activa (A.A.)	105.50mm (L) ×67.20mm (W)	
Área Visual (V.A.)	95.04mm (L) ×53.86mm (W)	
Resolución	480×272 píxeles	También se puede configurar como 272×480
Tipo táctil	Resistivo	
Toca	> 1 millón	
Luz de fondo	LED	
Vida útil de la luz de fondo (promedio)	>30.000 horas	
Brillo	250nit (NX4827K043_011N)	0% a 100%, el intervalo de ajuste es 1%
	230 nit (NX4827K043_011R)	0% a 100%, el intervalo de ajuste es 1%
Peso	79.3g (NX4827K043_011N)	
	93.8g (NX4827K043_011R)	

Características electrónicas

	Condiciones de prueba	Min	Típico	Máximo	Unidad
Voltaje de funcionamiento		4.75	5	7	V
Corriente de funcionamiento	VCC = + 5V, el brillo es 100%	–	250	–	mamá
	Modo SLEEP	–	15	–	mamá
Fuente de alimentación recomendada: **5V, 1.0A, DC**					

Parámetro de entorno de trabajo y fiabilidad

	Condiciones de prueba	Min	Típico	Máximo	Unidad
Temperatura de trabajo	5V, Humedad 60%	-20	25	70	°C
Temperatura de almacenamiento		-30	25	85	°C
Humedad de trabajo	25° C	10%	60%	90%	RH

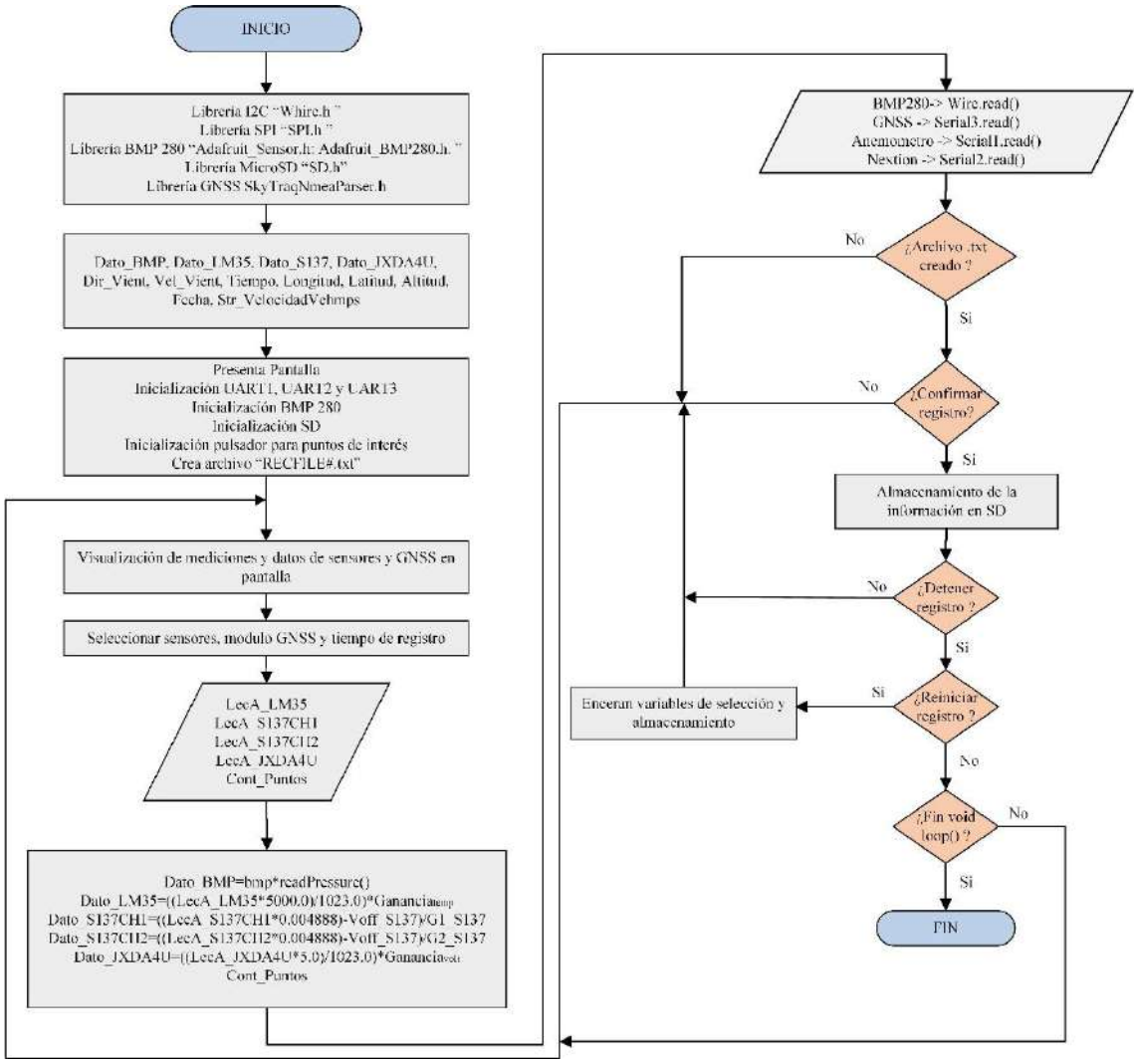
Rendimiento de las interfaces

	Condiciones de prueba	Min	Típico	Máximo	Unidad
Velocidad en baudios del puerto serie	Estándar	2400	9600	115200	bps
Salida de alto voltaje	IOH=-1mA	3.0	3.2		V
Salida de bajo voltaje	LIO = 1mA		0.1	0.2	V
Entrada de alto voltaje		2.0	3.3	5.0	V
Entrada de bajo voltaje		-0.7	0.0	1.3	V
Modo de puerto serie	TTL				
Puerto serie	4Pin_2,54 mm				
Interfaz USB	NO				
Zócalo para tarjeta SD	Sí (formato FAT32), admite un máximo de tarjetas Micro SD de 32G * El zócalo de la tarjeta microSD se utiliza exclusivamente para actualizar el firmware / diseño HMI de Nextion				
E/S extendida	8 GPIO digital extendido				
	IO0-IO7 admite entrada, salida y evento de enlace de componentes * Los pines / puertos de E/S no son exclusivos, se recomienda limitar el consumo de corriente a 1 mA				
	IO4-IO7 soporta PWM				
RTC	soporte RTC incorporado (Tipo de batería: CR1220)				

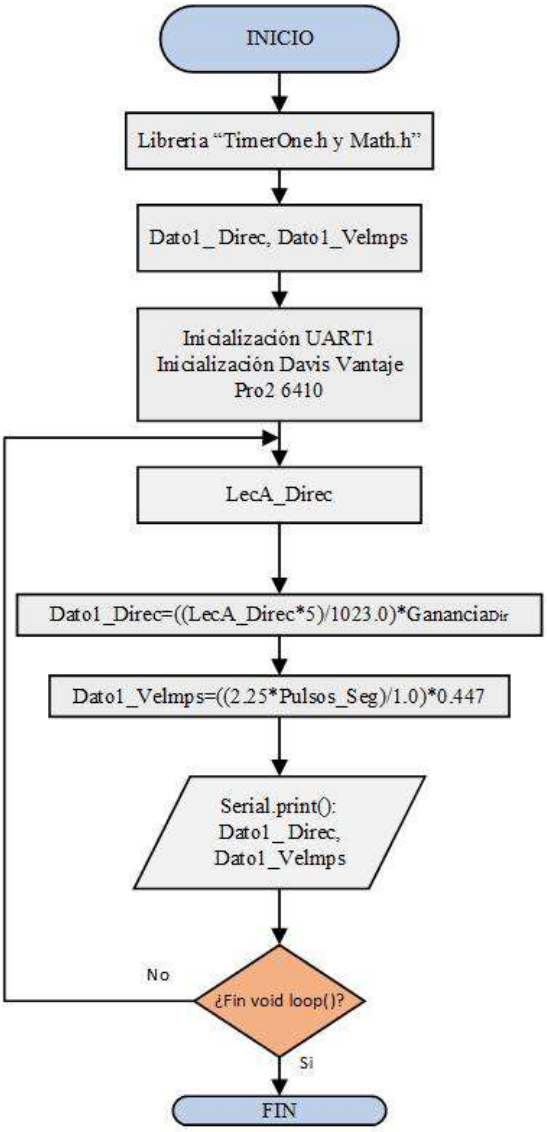
Características de la memoria

Tipo de memoria	Condiciones de prueba	Min	Típico	Máximo	Unidad
Memoria FLASH	Almacenar fuentes e imágenes			32	MB
Almacenamiento de usuario	EEPROM			1024	BYTE
Memoria RAM	Almacenar variables			8192	BYTE
Búfer de instrucciones	Búfer de instrucciones			1024	BYTE

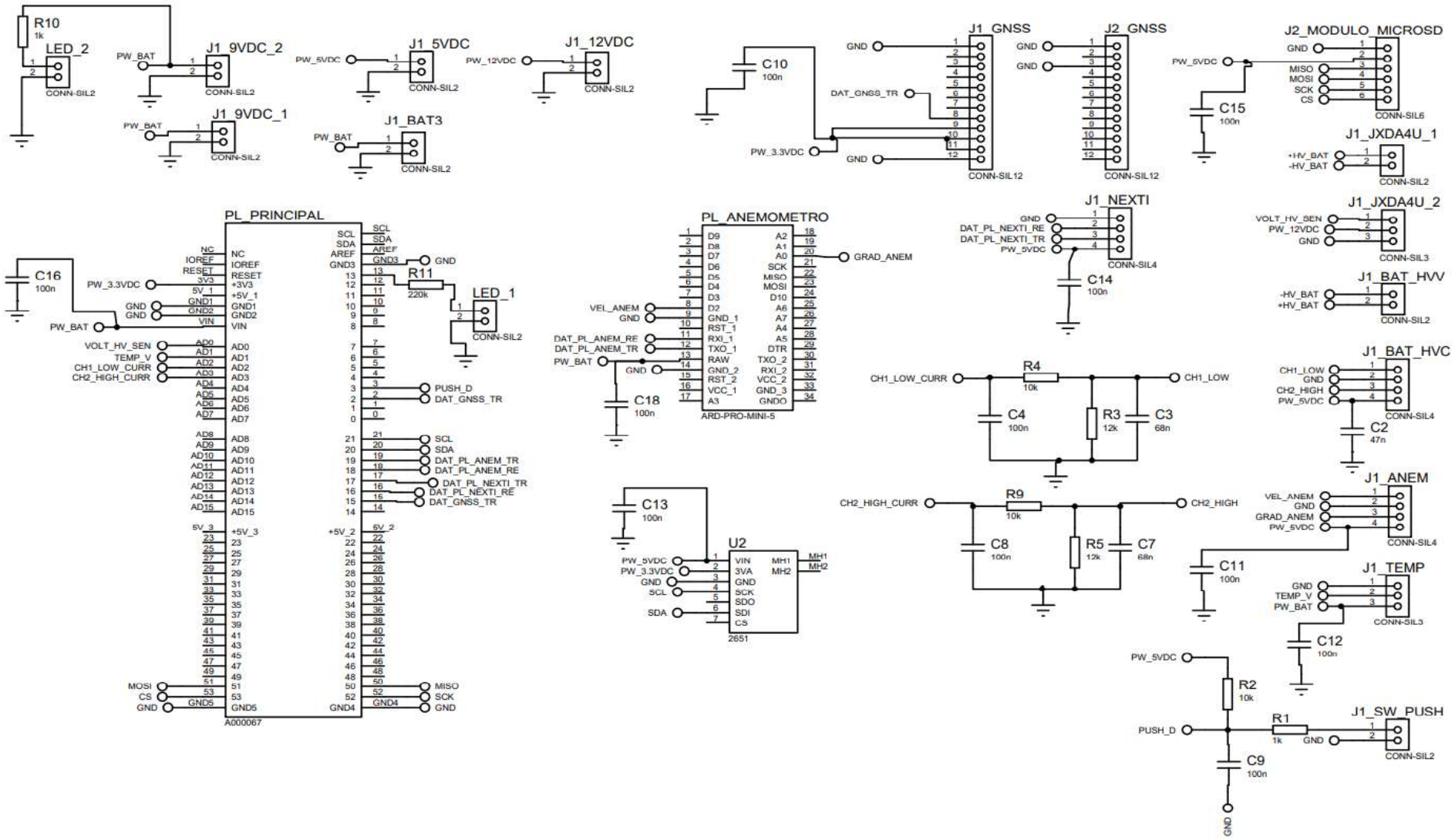
ANEXO K: DIAGRAMA DE FLUJO DE LA PROGRAMACIÓN DEL DATALOGGER CON ATMEGA 2560



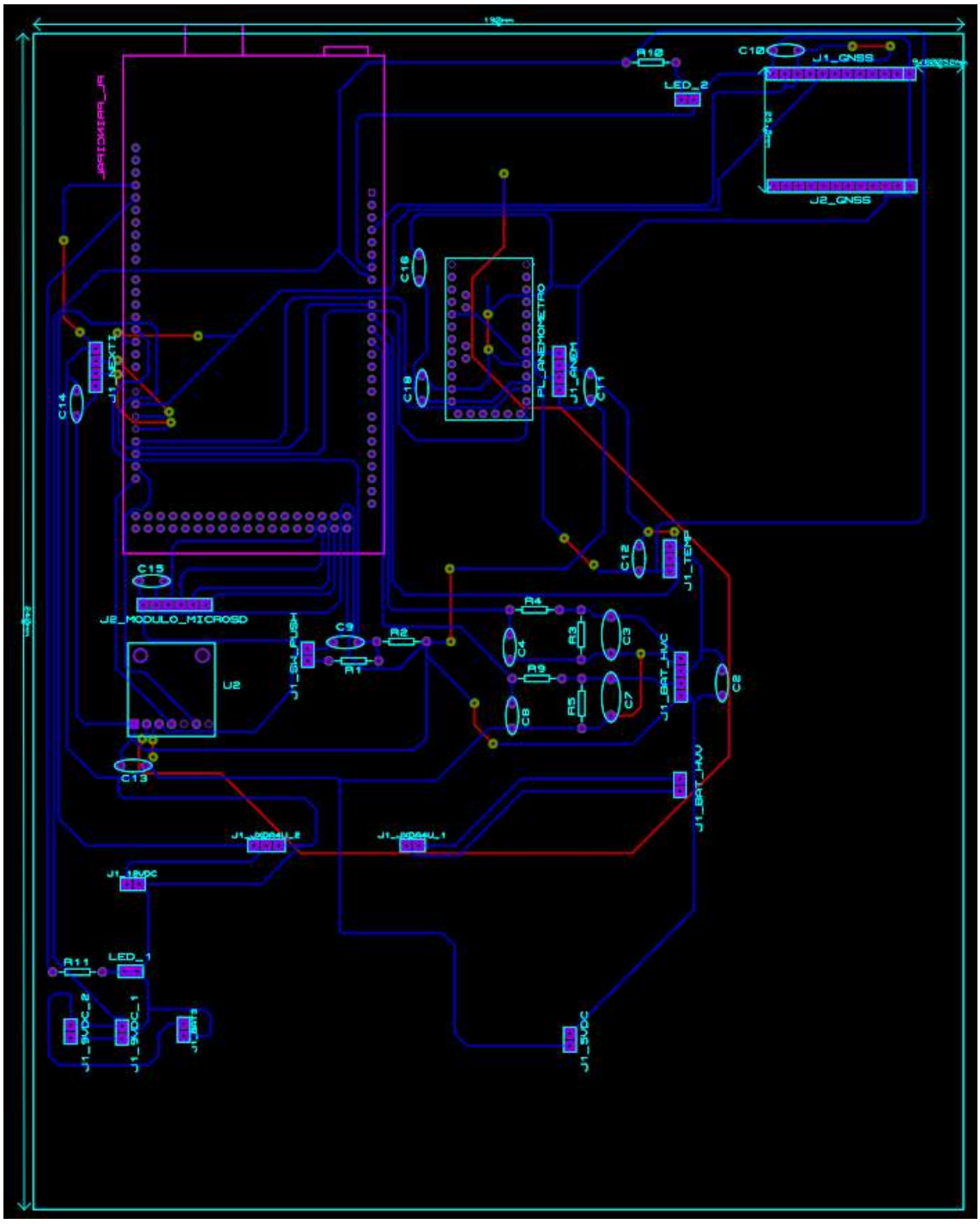
ANEXO L: DIAGRAMA DE FLUJO PARA ANEMÓMETRO CON ATMEGA 328P



ANEXO M: CIRCUITO COMPLETO DE DATALOGGER



ANEXO N: PCB PARA DATALOGGER



ANEXO O: FRAGMENTOS DE RESULTADO DE REGISTRO DE DATOS EN FORMATO .XLSX, PARA INTERVALOS DE REGISTRO DE 1 SEGUNDO

Cantidad de datos	TIEMPO_seg	PRESION_hPa	TEMPERATURA_°C	CORRIENTE_A	VOLTAJE_V	VELOCIDAD_VIENTO_m/s	DIRECCION_VIENTO_°	LATITUD_ddmm.mmmmmmm	LONGITUD_dddmm.mmmmmm	ALTITUD_m	VELOCIDAD_VEHICULO_m/s	PUNTOS_INTERES
1	1.00	732.77	13.69	19.32	0.00	2.01	225.60	139.21	7840.51	2651.41	0.05	0.00
2	2.00	732.75	12.71	14.74	0.00	1.01	225.60	139.21	7840.51	2651.65	0.05	0.00
3	3.00	732.76	13.20	18.22	0.00	1.01	224.90	139.21	7840.51	2651.47	0.05	0.00
4	4.00	732.75	13.69	15.84	0.00	1.01	225.60	139.21	7840.51	2651.64	0.05	0.00
5	5.00	732.73	12.71	14.56	0.00	0.00	225.60	139.21	7840.51	2651.87	0.05	0.00
6	6.00	732.75	13.20	13.82	0.00	1.01	232.30	139.21	7840.51	2651.61	0.36	0.00
7	7.00	732.72	12.71	17.30	0.00	0.00	350.90	139.21	7840.51	2651.91	0.36	0.00
8	8.00	732.68	12.71	12.72	0.00	0.00	349.40	139.21	7840.51	2652.42	2.88	0.00
9	9.00	732.67	13.20	12.72	0.00	2.01	346.60	139.20	7840.51	2652.42	3.80	0.00
10	10.00	732.69	13.69	15.29	0.00	2.01	356.80	139.20	7840.52	2652.29	4.37	0.00
11	11.00	732.68	12.71	12.91	0.00	3.02	355.80	139.20	7840.52	2652.31	4.78	0.00
12	12.00	732.64	13.20	15.11	0.00	4.02	358.20	139.20	7840.52	2652.79	4.63	0.00
13	13.00	732.65	12.71	15.47	0.00	4.02	359.30	139.20	7840.52	2652.72	4.78	0.00
14	14.00	732.63	12.71	16.59	0.00	4.02	356.50	139.20	7840.52	2652.89	5.04	0.00
15	15.00	732.62	13.20	12.17	0.00	4.02	360.00	139.19	7840.53	2652.98	5.29	0.00
16	16.00	732.60	12.71	11.62	0.00	4.02	360.00	139.19	7840.53	2653.28	5.60	0.00
17	17.00	732.56	12.71	15.29	0.00	4.02	360.00	139.19	7840.53	2653.69	5.50	0.00
18	18.00	732.54	12.22	12.17	0.00	4.02	360.00	139.19	7840.53	2653.91	6.01	0.00
19	19.00	732.49	13.20	15.66	0.00	5.03	360.00	139.19	7840.54	2654.48	6.84	0.00
20	20.00	732.46	12.71	15.66	0.00	7.04	360.00	139.18	7840.54	2654.78	7.86	0.00
21	21.00	732.48	12.71	18.77	0.00	7.04	360.00	139.18	7840.54	2654.56	8.07	0.00
22	22.00	732.51	13.20	17.30	0.00	6.03	360.00	139.18	7840.54	2654.18	7.45	0.00
23	23.00	732.49	12.71	18.22	0.00	6.03	360.00	139.17	7840.55	2654.44	6.37	0.00
24	24.00	732.51	12.22	16.76	0.00	4.02	360.00	139.17	7840.55	2654.19	5.14	0.00
25	25.00	732.50	12.71	16.76	0.00	4.02	360.00	139.17	7840.55	2654.33	4.16	0.00
26	26.00	732.50	12.71	13.82	0.00	3.02	360.00	139.17	7840.55	2654.32	3.44	0.00
27	27.00	732.49	12.71	14.37	0.00	3.02	0.00	139.17	7840.55	2654.42	3.70	0.00
28	28.00	732.47	12.71	15.66	0.00	3.02	360.00	139.17	7840.55	2654.63	4.42	0.00
29	29.00	732.44	12.22	11.07	0.00	3.02	356.80	139.16	7840.55	2655.00	5.65	0.00
30	30.00	732.40	12.22	12.91	0.00	5.03	360.00	139.16	7840.55	2655.42	6.17	0.00
31	31.00	732.39	12.71	15.47	0.00	6.03	360.00	139.16	7840.55	2655.34	6.94	0.00
32	32.00	732.38	12.22	12.72	0.00	5.03	360.00	139.15	7840.54	2655.62	7.35	0.00
33	33.00	732.39	11.73	14.92	0.00	6.03	360.00	139.15	7840.54	2655.46	7.09	0.00
34	34.00	732.36	12.22	18.22	0.00	6.03	358.90	139.15	7840.54	2655.78	6.48	0.00
35	35.00	732.42	12.71	18.04	0.00	5.03	359.70	139.15	7840.54	2655.20	6.07	0.00
36	36.00	732.43	11.73	15.84	0.00	5.03	360.00	139.14	7840.54	2655.10	5.45	0.00
37	37.00	732.43	12.22	16.94	0.00	4.02	359.30	139.14	7840.53	2655.05	4.78	0.00
562	562.00	724.17	12.71	14.56	0.00	1.01	357.50	139.61	7840.88	2744.88	0.00	2.00
563	563.00	724.18	12.22	13.09	0.00	1.01	360.00	139.61	7840.88	2744.79	0.00	2.00
564	564.00	724.06	12.22	12.72	0.00	9.05	349.10	139.61	7840.88	2746.13	0.00	2.00
565	565.00	724.04	12.22	11.44	0.00	1.01	356.10	139.61	7840.88	2746.37	0.00	2.00
566	566.00	724.01	12.71	12.54	0.00	1.01	360.00	139.61	7840.88	2746.68	0.00	2.00
567	567.00	723.98	13.20	13.09	0.00	1.01	360.00	139.61	7840.88	2747.00	0.00	2.00
568	568.00	723.94	12.22	12.36	0.00	9.05	40.00	139.61	7840.88	2747.47	0.00	2.00
569	569.00	723.94	12.22	12.36	0.00	9.05	40.00	139.61	7840.88	2747.47	0.00	2.00
570	570.00	723.84	12.71	14.56	0.00	0.00	0.00	139.61	7840.88	2748.56	0.00	2.00
571	571.00	723.82	12.71	11.62	0.00	0.00	0.00	139.61	7840.88	2748.79	0.00	2.00
572	572.00	691.30	12.71	18.04	0.00	0.00	0.00	139.61	7840.88	3110.91	0.00	2.00
573	573.00	691.30	12.22	14.56	0.00	9.05	60.00	139.61	7840.88	3110.91	0.00	2.00
574	574.00	691.30	12.71	14.19	0.00	8.05	350.90	139.61	7840.88	3110.91	0.00	2.00
575	575.00	691.30	12.22	12.36	0.00	9.05	360.00	139.61	7840.88	3110.91	0.00	2.00
576	576.00	691.30	12.22	14.92	0.00	8.05	356.50	139.61	7840.88	3110.91	0.00	2.00
577	577.00	691.30	12.22	14.92	0.00	9.05	35.00	139.61	7840.88	3110.91	0.00	2.00
578	578.00	691.30	12.71	14.74	0.00	8.05	360.00	139.61	7840.88	3110.91	0.00	2.00
579	579.00	691.30	12.22	15.11	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
580	580.00	691.30	12.71	15.47	0.00	9.05	360.00	139.61	7840.88	3110.91	0.00	2.00
581	581.00	691.30	12.71	13.46	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
582	582.00	691.30	12.71	13.46	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
583	583.00	691.30	12.22	12.17	0.00	1.01	357.20	139.61	7840.88	3110.91	0.00	2.00
584	584.00	691.30	11.73	11.62	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
585	585.00	691.30	12.71	13.64	0.00	1.11	0.00	139.61	7840.88	3110.91	0.00	2.00
586	586.00	691.30	11.73	12.54	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
587	587.00	691.30	12.71	15.47	0.00	1.11	360.00	139.61	7840.88	3110.91	0.00	2.00
588	588.00	691.30	12.22	12.91	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
589	589.00	691.30	12.22	15.11	0.00	1.11	358.20	139.61	7840.88	3110.91	0.00	2.00
590	590.00	691.30	12.22	12.91	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
591	591.00	691.30	12.22	13.46	0.00	1.01	0.00	139.61	7840.88	3110.91	0.00	2.00
592	592.00	691.30	12.22	14.92	0.00	1.01	352.60	139.61	7840.88	3110.91	0.00	2.00
593	593.00	691.30	11.73	15.11	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
594	594.00	691.30	12.22	15.11	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
595	595.00	691.30	11.73	14.74	0.00	1.11	360.00	139.61	7840.88	3110.91	0.00	2.00
596	596.00	691.30	12.22	15.47	0.00	1.01	358.60	139.61	7840.88	3110.91	0.00	2.00
597	597.00	691.30	12.22	14.74	0.00	1.11	11.30	139.61	7840.88	3110.91	0.00	2.00
598	598.00	691.30	12.22	12.72	0.00	9.05	360.00	139.61	7840.88	3110.91	0.00	2.00
599	599.00	691.30	12.22	12.54	0.00	1.01	360.00	139.61	7840.88	3110.91	0.00	2.00
600	600.00	691.30	11.73	14.92	0.00	1.01	0.00	139.61	7840.88	3110.91	0.00	2.00

Cantidad de datos	TIEMPO_seg	PRESION_hPa	TEMPERATURA_°C	CORRIENTE_A	VOLTAJE_V	VELOCIDAD_VIENTO_m/s	DIRECCION_VIENTO_°	LATITUD_ddmm.mmmmmmm	LONGITUD_dddmm.mmmmmm	ALTITUD_m	VELOCIDAD_VEHICULO_m/s	PUNTOS_INTERES
1	1	0	0	0	93.84	0	0	0	0	0	0	0
2	2	0	0	0	93.84	0	0	0	0	0	0	0
3	3	0	0	0	93.84	0	0	0	0	0	0	0
4	4	0	0	0	93.84	0	0	0	0	0	0	0
5	5	0	0	0	95.8	0	0	0	0	0	0	0
6	6	0	0	0	93.84	0	0	0	0	0	0	0
7	7	0	0	0	93.84	0	0	0	0	0	0	0
8	8	0	0	0	93.84	0	0	0	0	0	0	0
9	9	0	0	0	93.84	0	0	0	0	0	0	0
10	10	0	0	0	93.84	0	0	0	0	0	0	0
11	11	0	0	0	93.84	0	0	0	0	0	0	0
12	12	0	0	0	93.84	0	0	0	0	0	0	0
13	13	0	0	0	93.84	0	0	0	0	0	0	0
14	14	0	0	0	93.84	0	0	0	0	0	0	0
15	15	0	0	0	94.82	0	0	0	0	0	0	0
16	16	0	0	0	93.84	0	0	0	0	0	0	0
17	17	0	0	0	93.84	0	0	0	0	0	0	0
18	18	0	0	0	93.84	0	0	0	0	0	0	0
19	19	0	0	0	93.84	0	0	0	0	0	0	0
20	20	0	0	0	93.84	0	0	0	0	0	0	0
21	21	0	0	0	94.82	0	0	0	0	0	0	0
22	22	0	0	0	92.86	0	0	0	0	0	0	0
23	23	0	0	0	97.75	0	0	0	0	0	0	0
24	24	0	0	0	93.84	0	0	0	0	0	0	0
25	25	0	0	0	93.84	0	0	0	0	0	0	0
26	26	0	0	0	93.84	0	0	0	0	0	0	0
27	27	0	0	0	93.84	0	0	0	0	0	0	0
28	28	0	0	0	93.84	0	0	0	0	0	0	0
29	29	0	0	0	93.84	0	0	0	0	0	0	0
30	30	0	0	0	94.82	0	0	0	0	0	0	0
31	31	0	0	0	93.84	0	0	0	0	0	0	0
32	32	0	0	0	93.84	0	0	0	0	0	0	0
33	33	0	0	0	93.84	0	0	0	0	0	0	0
34	34	0	0	0	93.84	0	0	0	0	0	0	0
35	35	0	0	0	93.84	0	0	0	0	0	0	0
36	36	0	0	0	93.84	0	0	0	0	0	0	0
37	37	0	0	0	93.84	0	0	0	0	0	0	0
38	38	0	0	0	93.84	0	0	0	0	0	0	0
560	560	0	0	0	93.84	0	0	0	0	0	0	0
561	561	0	0	0	93.84	0	0	0	0	0	0	0
562	562	0	0	0	93.84	0	0	0	0	0	0	0
563	563	0	0	0	94.82	0	0	0	0	0	0	0
564	564	0	0	0	94.82	0	0	0	0	0	0	0
565	565	0	0	0	93.84	0	0	0	0	0	0	0
566	566	0	0	0	93.84	0	0	0	0	0	0	0
567	567	0	0	0	94.82	0	0	0	0	0	0	0
568	568	0	0	0	93.84	0	0	0	0	0	0	0
569	569	0	0	0	93.84	0	0	0	0	0	0	0
570	570	0	0	0	93.84	0	0	0	0	0	0	0
571	571	0	0	0	93.84	0	0	0	0	0	0	0
572	572	0	0	0	93.84	0	0	0	0	0	0	0
573	573	0	0	0	94.82	0	0	0	0	0	0	0
574	574	0	0	0	93.84	0	0	0	0	0	0	0
575	575	0	0	0	93.84	0	0	0	0	0	0	0
576	576	0	0	0	93.84	0	0	0	0	0	0	0
577	577	0	0	0	93.84	0	0	0	0	0	0	0
578	578	0	0	0	93.84	0	0	0	0	0	0	0
579	579	0	0	0	93.84	0	0	0	0	0	0	0
580	580	0	0	0	93.84	0	0	0	0	0	0	0
581	581	0	0	0	93.84	0	0	0	0	0	0	0
582	582	0	0	0	95.8	0	0	0	0	0	0	0
583	583	0	0	0	93.84	0	0	0	0	0	0	0
584	584	0	0	0	93.84	0	0	0	0	0	0	0
585	585	0	0	0	93.84	0	0	0	0	0	0	0
586	586	0	0	0	93.84	0	0	0	0	0	0	0
587	587	0	0	0	93.84	0	0	0	0	0	0	0
588	588	0	0	0	93.84	0	0	0	0	0	0	0
589	589	0	0	0	93.84	0	0	0	0	0	0	0
590	590	0	0	0	93.84	0	0	0	0	0	0	0
591	591	0	0	0	94.82	0	0	0	0	0	0	0
592	592	0	0	0	94.82	0	0	0	0	0	0	0
593	593	0	0	0	93.84	0	0	0	0	0	0	0
594	594	0	0	0	93.84	0	0	0	0	0	0	0
595	595	0	0	0	93.84	0	0	0	0	0	0	0
596	596	0	0	0	93.84	0	0	0	0	0	0	0
597	597	0	0	0	93.84	0	0	0	0	0	0	0
598	598	0	0	0	93.84	0	0	0	0	0	0	0
599	599	0	0	0	93.84	0	0	0	0	0	0	0
600	600	0	0	0	93.84	0	0	0	0	0	0	0

ANEXO P: FRAGMENTOS DE RESULTADO DE REGISTRO DE DATOS EN FORMATO .XLSX, PARA INTERVALOS DE REGISTRO DE 2 SEGUNDOS

Cantidad de datos	TIEMPO_seg	PRESION_hPa	TEMPERATURA_°C	CORRIENTE_A	VOLTAJE_V	VELOCIDAD_VIENTO_m/s	DIRECCION_VIENTO_°	LATITUD_ddmm.mmmmmmm	LONGITUD_dddmm.mmmmmm	ALTITUD_m	VELOCIDAD_VEHICULO_m/s	PUNTOS_INTERES
1	2	727.09	12.71	20.24	0	1.01	203.8	139.33705	7841.7632	2713.04	0	0
2	4	727.08	13.2	20.79	0	1.01	190.7	139.33705	7841.7632	2713.18	0	0
3	6	727.11	13.2	17.12	0	1.01	184.4	139.33705	7841.7632	2712.84	0	0
4	8	727.1	13.69	17.85	0	1.01	182.3	139.33704	7841.7631	2712.94	0.1	0
5	10	727.08	13.2	16.02	0	0	187.9	139.33704	7841.7631	2713.12	0	0
6	12	727.1	13.2	19.32	0	1.01	190.4	139.33705	7841.7631	2712.98	0	0
7	14	727.08	13.69	16.76	0	1.01	198.8	139.33705	7841.7631	2713.13	0	0
8	16	727.09	13.69	16.76	0	0	215.4	139.33705	7841.7631	2713.05	0	0
9	18	727.12	13.2	19.32	0	1.01	216.8	139.33706	7841.7631	2712.75	0	0
10	20	727.07	13.69	17.3	0	0	218.9	139.33706	7841.7631	2713.31	0.05	0
11	22	727.1	13.2	17.3	0	1.01	294.2	139.33734	7841.7621	2712.93	2.36	0
12	24	727.05	13.69	15.11	0	2.01	351.6	139.33869	7841.7576	2713.45	5.4	0
13	26	727.09	13.69	14.01	0	4.02	358.6	139.34108	7841.7514	2713.06	6.53	0
14	28	727.14	13.69	16.39	0	5.03	360	139.34395	7841.7439	2712.51	7.86	0
15	30	727.16	13.2	14.56	0	7.04	354.4	139.34729	7841.735	2712.26	9.25	0
16	32	727.18	13.2	20.24	0	7.04	360	139.35074	7841.7256	2712.05	9.2	0
17	34	727.19	13.2	14.56	0	8.05	360	139.35403	7841.7162	2711.99	9.25	0
18	36	727.18	13.69	17.67	0	9.05	360	139.35719	7841.7066	2712.11	9.51	0
19	38	727.23	13.69	16.94	0	10.1	360	139.36046	7841.6963	2711.51	10.33	0
20	40	727.23	13.69	14.37	0	10.1	360	139.36415	7841.685	2711.59	11.41	0
21	42	727.31	13.69	15.47	0	12.1	360	139.36836	7841.6726	2710.66	12.54	0
22	44	727.29	13.69	16.94	0	12.1	360	139.37294	7841.659	2710.92	13.52	0
23	46	727.32	12.71	14.92	0	13.1	360	139.37785	7841.6445	2710.56	14.49	0
24	48	727.39	13.2	17.85	0	14.1	360	139.38319	7841.6287	2709.77	15.99	0
25	50	727.58	13.2	18.95	0	15.1	360	139.38921	7841.6114	2707.68	17.32	0
26	52	727.73	13.2	20.05	0	15.1	360	139.39507	7841.5949	2706.05	15.52	0
27	54	727.83	13.2	17.85	0	12.1	0	139.40017	7841.5801	2704.96	13.93	0
28	56	727.89	13.2	14.01	0	13.1	360	139.4053	7841.5651	2704.4	15.32	0
29	58	727.91	13.2	21.15	0	14.1	360	139.41078	7841.5491	2704.12	15.78	0
30	60	728.07	12.71	17.49	0	15.1	360	139.41639	7841.5326	2702.45	16.04	0
31	62	728.18	13.2	17.3	0	14.1	359.7	139.42218	7841.5166	2701.15	15.88	0
32	64	728.32	12.71	20.97	0	15.1	360	139.42774	7841.5007	2699.69	15.42	0
33	66	728.51	13.2	20.6	0	14.1	360	139.43302	7841.4855	2697.57	14.65	0
34	68	728.62	13.2	18.59	0	11.1	360	139.43786	7841.4714	2696.41	13.21	0
35	70	728.74	12.71	17.12	0	10.1	360	139.44202	7841.4593	2695.06	11.21	0
36	72	728.73	13.2	15.66	0	10.1	360	139.44601	7841.4482	2695.17	11.21	0
37	74	728.74	12.71	14.56	0	11.1	360	139.45019	7841.4359	2695.08	12.49	0
260	520	732.89	14.17	18.59	0	14.1	360	139.28145	7840.33	2650.15	14.91	1
261	522	732.89	14.66	16.57	0	14.1	356.5	139.27161	7840.3429	2650.11	14.8	1
262	524	732.69	14.66	15.29	0	13.1	360	139.26174	7840.3558	2652.25	15.11	1
263	526	732.69	14.17	17.85	0	15.1	355.1	139.2517	7840.3689	2652.29	15.27	1
264	528	732.66	14.17	14.56	0	13.1	360	139.2416	7840.3821	2652.54	15.32	1
265	530	732.5	14.17	15.11	0	14.1	352.3	139.23148	7840.3952	2654.35	15.27	1
266	532	732.42	13.69	17.3	0	13.1	358.9	139.22147	7840.4081	2655.17	14.96	1
267	534	732.35	14.66	17.49	0	14.1	358.9	139.21189	7840.4205	2656.25	14.14	1
268	536	732.3	14.17	17.3	0	13.1	359.7	139.20264	7840.4324	2656.52	13.67	1
269	538	732.32	14.17	17.85	0	13.1	359.3	139.19355	7840.4439	2656.26	13.42	1
270	540	732.3	13.69	17.49	0	11.1	360	139.18458	7840.4549	2656.47	13.11	1
271	542	732.23	13.2	18.04	0	12.1	351.9	139.17597	7840.466	2657.23	13.06	1
272	544	732.25	13.69	17.67	0	12.1	356.5	139.16711	7840.4772	2657.06	13.26	1
273	546	732.17	13.2	19.32	0	10.1	360	139.15796	7840.4882	2657.88	13.16	1
274	548	732.21	13.69	27.75	0	8.05	360	139.14926	7840.4986	2657.48	12.03	1
275	550	732.17	13.2	19.32	0	8.05	353.7	139.1418	7840.5076	2657.84	10.07	1
276	552	732.26	13.69	26.47	0	7.04	360	139.13616	7840.5144	2656.89	7.25	1
277	554	732.31	13.2	24.63	0	5.03	351.9	139.13348	7840.519	2656.35	3.44	1
278	556	732.29	13.69	21.7	0	3.02	326.9	139.13332	7840.5205	2656.63	0.1	1
279	558	732.3	13.2	18.77	0	2.01	266	139.13332	7840.5205	2656.48	0	1
280	560	732.29	13.69	18.4	0	2.01	263.2	139.13332	7840.5205	2656.61	0.1	1
281	562	732.33	13.69	16.57	0	1.01	260.4	139.13332	7840.5205	2656.2	0	1
282	564	732.3	13.69	15.29	0	1.01	277	139.13332	7840.5205	2656.47	0.62	1
283	566	732.24	13.69	15.84	0	2.01	0	139.13343	7840.5227	2657.16	2.72	1
284	568	732.29	13.69	17.85	0	4.02	360	139.13401	7840.5259	2656.56	3.08	1
285	570	732.32	13.69	16.39	0	3.02	360	139.13485	7840.5283	2656.27	1.9	1
286	572	732.25	13.69	15.66	0	2.01	360	139.13578	7840.53	2656.97	1.7	1
287	574	732.2	13.2	17.85	0	3.02	360	139.13819	7840.5326	2657.52	4.16	1
288	576	732.15	13.2	15.29	0	6.03	360	139.14258	7840.536	2658.13	5.91	1
289	578	732.13	13.69	15.66	0	9.05	360	139.14923	7840.5407	2658.32	8.28	1
290	580	732.25	13.2	18.22	0	9.05	360	139.1568	7840.5462	2657.07	8.38	1
291	582	732.3	13.2	13.46	0	7.04	360	139.16292	7840.5504	2656.53	5.86	1
292	584	732.31	13.2	17.12	0	5.03	360	139.16808	7840.5515	2656.34	4.73	1
293	586	732.29	13.2	18.22	0	7.04	360	139.17361	7840.5481	2656.63	6.99	1
294	588	732.26	13.2	15.11	0	8.05	359.3	139.17986	7840.5416	2656.93	8.94	1
295	590	732.33	12.71	20.42	0	9.05	360	139.18693	7840.5341	2656.14	9.41	1
296	592	732.43	12.71	20.05	0	9.05	360	139.1934	7840.5273	2655.06	8.28	1
297	594	732.47	12.71	14.37	0	7.04	358.6	139.19891	7840.5212	2654.6	7.25	1
298	596	732.51	13.2	20.97	0	7.04	360	139.20434	7840.516	2654.22	7.04	1
299	598	732.6	12.71	18.59	0	6.03	353.7	139.21146	7840.5094	2653.22	4.78	1
300	600	732.62	12.71	15.29	0	4.02	339.6	139.21253	7840.5077	2652.96	3.65	1

Cantidad de datos	TIEMPO_seg	PRESION_hPa	TEMPERATURA_°C	CORRIENTE_A	VOLTAJE_V	VELOCIDAD_VIENTO_m/s	DIRECCION_VIENTO_°	LATITUD_ddmm.mmmmmmm	LONGITUD_dddmm.mmmmmm	ALTITUD_m	VELOCIDAD_VEHICULO_m/s	PUNTOS_INTERES
1	2	0	0	0	93.84	0	0	0	0	0	0	0
2	4	0	0	0	93.84	0	0	0	0	0	0	0
3	6	0	0	0	93.84	0	0	0	0	0	0	0
4	8	0	0	0	93.84	0	0	0	0	0	0	0
5	10	0	0	0	93.84	0	0	0	0	0	0	0
6	12	0	0	0	93.84	0	0	0	0	0	0	0
7	14	0	0	0	96.77	0	0	0	0	0	0	0
8	16	0	0	0	93.84	0	0	0	0	0	0	0
9	18	0	0	0	93.84	0	0	0	0	0	0	0
10	20	0	0	0	93.84	0	0	0	0	0	0	0
11	22	0	0	0	95.8	0	0	0	0	0	0	0
12	24	0	0	0	93.84	0	0	0	0	0	0	0
13	26	0	0	0	93.84	0	0	0	0	0	0	0
14	28	0	0	0	93.84	0	0	0	0	0	0	0
15	30	0	0	0	93.84	0	0	0	0	0	0	0
16	32	0	0	0	94.82	0	0	0	0	0	0	0
17	34	0	0	0	93.84	0	0	0	0	0	0	0
18	36	0	0	0	93.84	0	0	0	0	0	0	0
19	38	0	0	0	93.84	0	0	0	0	0	0	0
20	40	0	0	0	93.84	0	0	0	0	0	0	0
21	42	0	0	0	93.84	0	0	0	0	0	0	0
22	44	0	0	0	93.84	0	0	0	0	0	0	0
23	46	0	0	0	93.84	0	0	0	0	0	0	0
24	48	0	0	0	93.84	0	0	0	0	0	0	0
25	50	0	0	0	93.84	0	0	0	0	0	0	0
26	52	0	0	0	93.84	0	0	0	0	0	0	0
27	54	0	0	0	93.84	0	0	0	0	0	0	0
28	56	0	0	0	93.84	0	0	0	0	0	0	0
29	58	0	0	0	94.82	0	0	0	0	0	0	0
30	60	0	0	0	94.82	0	0	0	0	0	0	0
31	62	0	0	0	94.82	0	0	0	0	0	0	0
32	64	0	0	0	93.84	0	0	0	0	0	0	0
33	66	0	0	0	93.84	0	0	0	0	0	0	0
34	68	0	0	0	93.84	0	0	0	0	0	0	0
35	70	0	0	0	93.84	0	0	0	0	0	0	0
36	72	0	0	0	93.84	0	0	0	0	0	0	0
37	74	0	0	0	93.84	0	0	0	0	0	0	0
260	520	0	0	0	93.84	0	0	0	0	0	0	0
261	522	0	0	0	93.84	0	0	0	0	0	0	0
262	524	0	0	0	93.84	0	0	0	0	0	0	0
263	526	0	0	0	93.84	0	0	0	0	0	0	0
264	528	0	0	0	93.84	0	0	0	0	0	0	0
265	530	0	0	0	95.8	0	0	0	0	0	0	0
266	532	0	0	0	94.82	0	0	0	0	0	0	0
267	534	0	0	0	93.84	0	0	0	0	0	0	0
268	536	0	0	0	93.84	0	0	0	0	0	0	0
269	538	0	0	0	93.84	0	0	0	0	0	0	0
270	540	0	0	0	93.84	0	0	0	0	0	0	0
271	542	0	0	0	93.84	0	0	0	0	0	0	0
272	544	0	0	0	93.84	0	0	0	0	0	0	0
273	546	0	0	0	93.84	0	0	0	0	0	0	0
274	548	0	0	0	93.84	0	0	0	0	0	0	0
275	550	0	0	0	93.84	0	0	0	0	0	0	0
276	552	0	0	0	93.84	0	0	0	0	0	0	0
277	554	0	0	0	94.82	0	0	0	0	0	0	0
278	556	0	0	0	93.84	0	0	0	0	0	0	0
279	558	0	0	0	93.84	0	0	0	0	0	0	0
280	560	0	0	0	93.84	0	0	0	0	0	0	0
281	562	0	0	0	93.84	0	0	0	0	0	0	0
282	564	0	0	0	93.84	0	0	0	0	0	0	0
283	566	0	0	0	95.8	0	0	0	0	0	0	0
284	568	0	0	0	93.84	0	0	0	0	0	0	0
285	570	0	0	0	94.82	0	0	0	0	0	0	0
286	572	0	0	0	93.84	0	0	0	0	0	0	0
287	574	0	0	0	93.84	0	0	0	0	0	0	0
288	576	0	0	0	93.84	0	0	0	0	0	0	0
289	578	0	0	0	93.84	0	0	0	0	0	0	0
290	580	0	0	0	93.84	0	0	0	0	0	0	0
291	582	0	0	0	95.8	0	0	0	0	0	0	0
292	584	0	0	0	93.84	0	0	0	0	0	0	0
293	586	0	0	0	93.84	0	0	0	0	0	0	0
294	588	0	0	0	94.82	0	0	0	0	0	0	0
295	590	0	0	0	93.84	0	0	0	0	0	0	0
296	592	0	0	0	93.84	0	0	0	0	0	0	0
297	594	0	0	0	94.82	0	0	0	0	0	0	0
298	596	0	0	0	93.84	0	0	0	0	0	0	0
299	598	0	0	0	95.8	0	0	0	0	0	0	0
300	600	0	0	0	95.8	0	0	0	0	0	0	0

ANEXO Q: MANUAL DE USUARIO DE DATALOGGER

Manual **Datalogger**

Proyecto Técnico (2023)

DATALOGGER PARA RECOLECCIÓN DE DATOS DE CONSUMO ENERGÉTICO DE UN VEHÍCULO ELÉCTRICO

Carrera Ingeniería Automotriz

RIOBAMBA-ECUADOR 2023

CONTENIDO

1 GENERALIDADES 3

2 CONEXIÓN DE COMPONENTES 4

3 ALIMENTACIÓN DEL SISTEMA 4

4 PANEL DE MANDOS DEL DATALOGGER 5

5 PANTALLA NEXTION 6
5.1 Operación 7

6 ESTRUCTURA DE REGISTRO DE DATOS 7

7 MONTAJE DE COMPONENTES EXTERNOS E INTERNOS DEL DATALOGGER 8

RECOMENDACIONES 8

1 GENERALIDADES

Los registradores electrónicos de datos son un nuevo desarrollo, el cual se caracteriza por su precisión y funcionamiento.

El diseño compacto con su modulo técnico permite una aplicación versátil. A través del mínimo consumo de energía y de la gran capacidad de memoria, el producto es apto para mediciones a largo plazo y aplicaciones tanto fijas como móviles.
El datalogger está constituido principalmente por los siguientes elementos:

- Sensor de presión BMP280
- Sensor de temperatura LM35A
- Sensor de velocidad de viento o anemómetro Vantage Pro2 6410
- Sensor de corriente DHAB S/137
- Sensor de voltaje DVC JXDA4U
- Módulo GNSS NS-HP-GN2
- Pulsador para identificación de puntos de interés
- Módulo MicroSD
- Arduino Mega 2560 – Atmega2560
- Arduino Pro Mini – ATmega328P
- Pantalla Nextion lNX4827K043_011R

El siguiente esquema muestra los puertos de conexión para cada uno de los elementos, tomando como elemento central el Atemega2560.

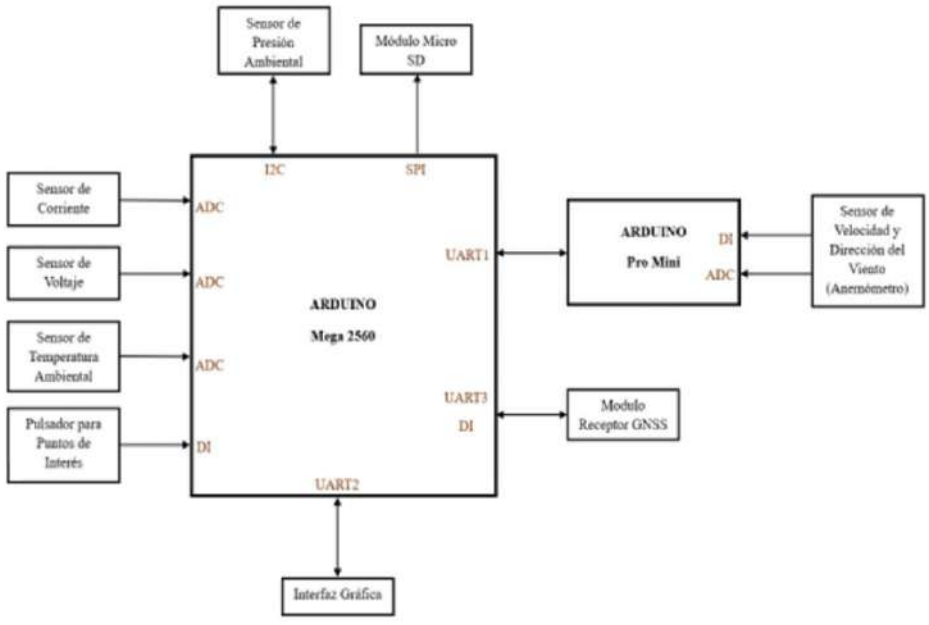


Figura 1 Elementos principales del datalogger

2 CONEXIÓN DE COMPONENTES

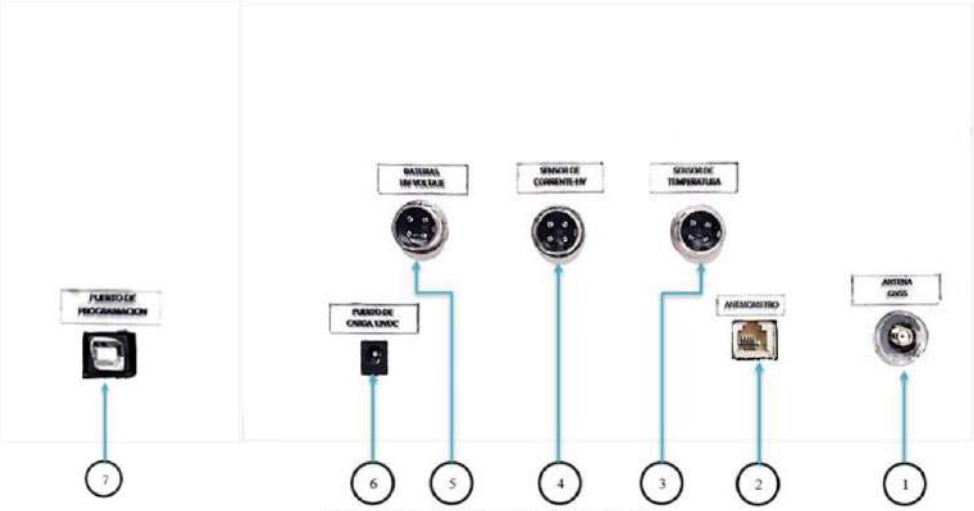


Figura 2 Conectores laterales del datalogger

1) Antena GNSS.
   Atornillar el cable de la antena al modulo
2) Anemómetro.
   Colocar el conector RJ11
3) Sensor de temperatura.
   Atornillar conector Gx16-4
4) Sensor de corriente – HV.
   Atornillar conector Gx16-4
5) Baterías HV – Voltaje.
   Atornillar conector Gx16-4 entrada de alto voltaje.
6) Puerto de carga 20 VDC.
   Conectar al Jack para recargar las baterías del datalogger
7) Puerto de programación USB JACK TYPE B

3 ALIMENTACIÓN DEL SISTEMA

1) Puerto de carga (Jack)
   Conectar al Jack para recargar las baterías a 16.8 V DC
2) BMS
   Supervisa la carga de las pilas individuales del paquete de baterías

3) Pilas

Las pilas 18650 es una batería recargable Li-ion, entrega a la salida un voltaje de 3,7 V y capacidad de 4800mAh

4 PANEL DE MANDOS DEL DATALOGGER

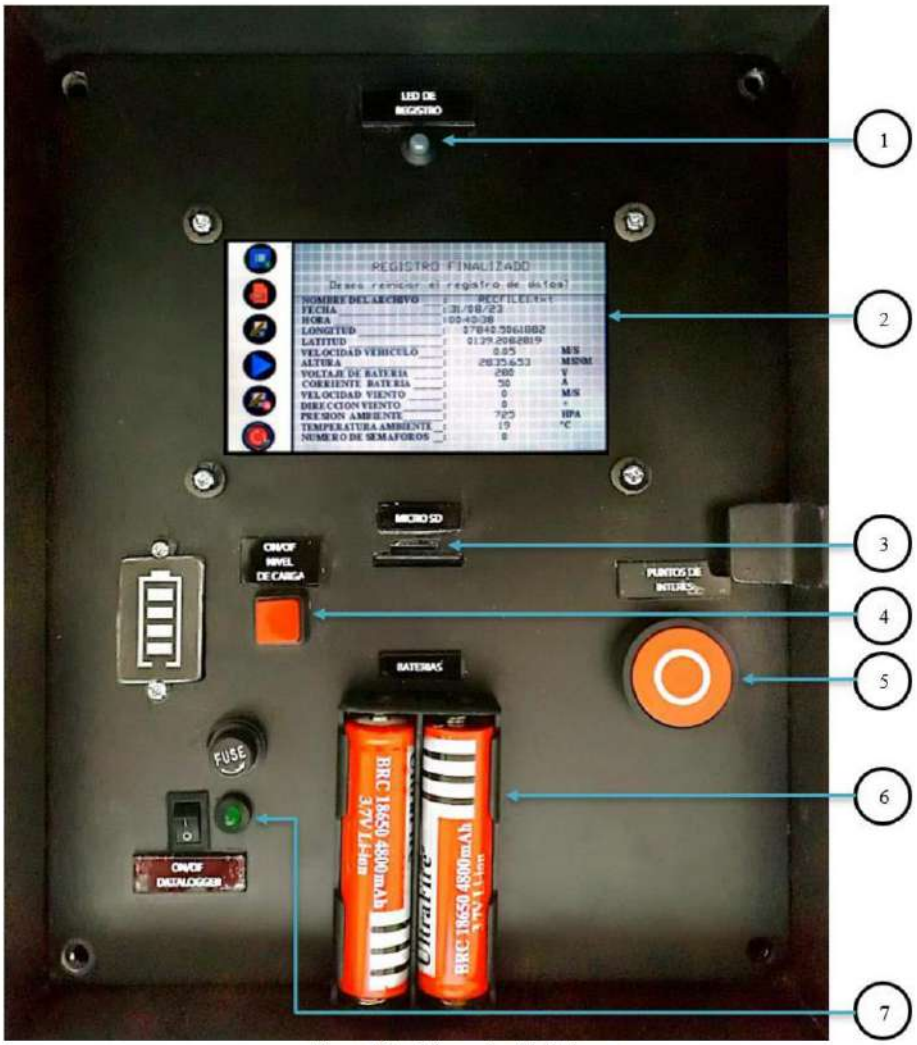


Figura 3 Panel de mandos del datalogger

1) Led

Indicador de registro de datos de los diferentes sensores seleccionados

2) Pantalla táctil

Para seleccionar y visualizar las operaciones

3) Micro SD

   Colocar el microSD para almacenar los datos y para retirar pulsar

4) Interruptor Switch Pulsador 2P, ON/OFF cuadrado

   Pulse para visualizar nivel de carga de las baterías del datalogger

5) Pulsador momentáneo conteo de puntos de interés

   Presionar el pulsador N/A para registrar los puntos de interés en ruta

6) Baterías

   Para la alimentación del sistema

7) Interruptor ON/OFF

   Pulse el interruptor ON (enciende led) para energizar el sistema

5 PANTALLA NEXTION

Partes de la interfaz gráfica:

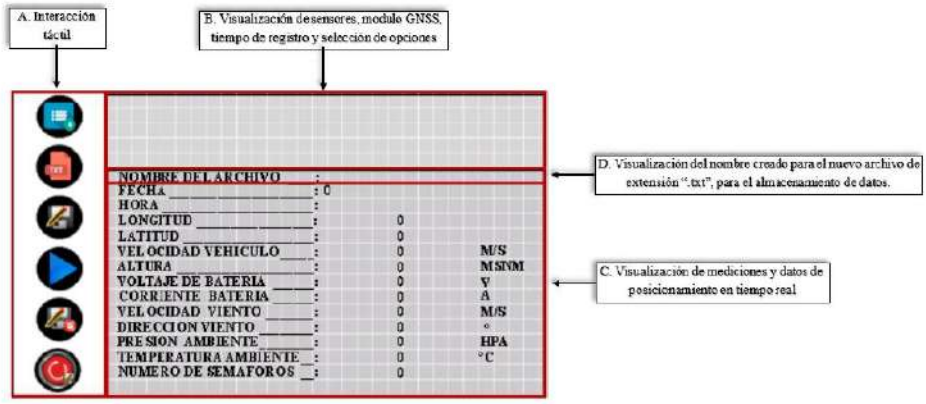


Figura 4 Partes de la interfaz grafica

En la siguiente imagen se presenta los botones táctiles para la operación del datalogger.

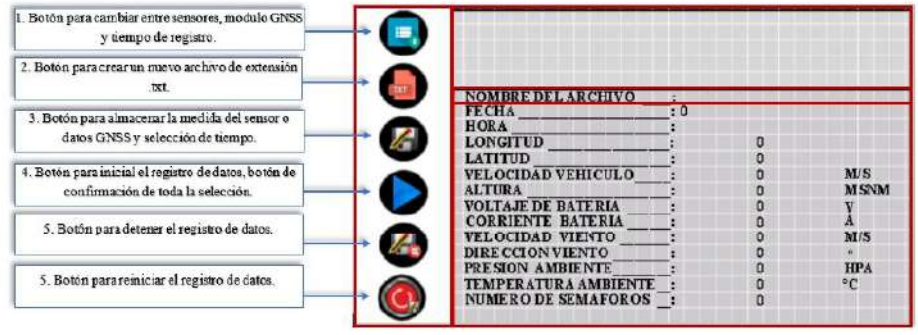


Figura 5 Botones de operación

Aclaración acerca de las funciones de la pantalla Nextion:

El equipo permite generar hasta nueve archivos de extensión .txt. que se muestra en la parte "D" Figura 4 donde al generar el número máximo de archivos indica mediante un mensaje.

En la figura 5 muestra 6 botones mismos que tienen la función de seleccionar los distintos sensores, modulo y tiempo de registro que dispone el datalogger, así como crear los archivos de extensión .txt., iniciar el registro de datos, detener el proceso y reiniciar.

5.1 Operación

1) Una vez que este todo conectado se procede a prender el datalogger
2) Prender la pantalla nextion
3) Seleccionar los sensores de interés con el botón 1
4) Presionar el botón 2 para crear el archivo de extensión .txt.
5) Presionar el botón 3 para almacenar los datos
6) Presionar botón 4 para inicialización de funcionamiento de datalogger para selección y registro de datos de los elementos seleccionados con el botón 1
7) Los botones 5 están configurados para detener y reiniciar el registro de datos

6 ESTRUCTURA DE REGISTRO DE DATOS

Los datos almacenados en el archivo RECFILE.txt. se visualizará como se muestra en la siguiente imagen de izquierda a derecha se tiene el tiempo en segundos, presión, temperatura, corriente, voltaje, velocidad viento, dirección del viento, latitud, longitud, altitud, velocidad del vehículo y puntos de interés.

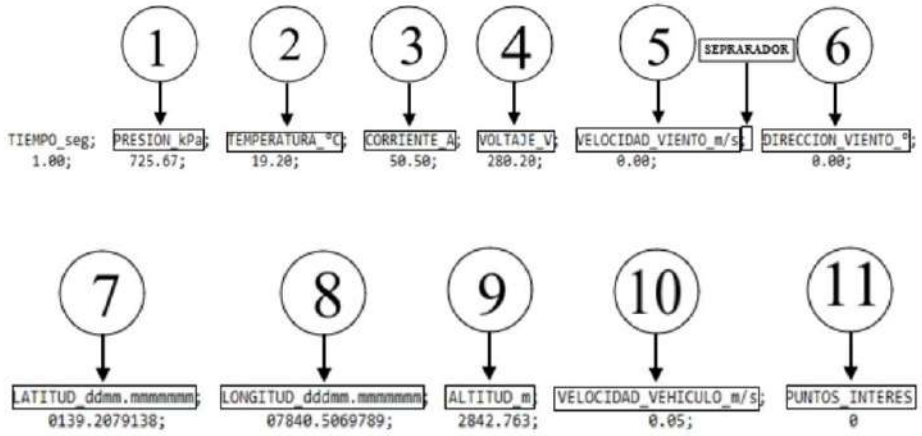


Figura 6 Estructura de registro de datos

7 MONTAJE DE COMPONENTES EXTERNOS E INTERNOS DEL DATALOGGER

1) Módulo receptor y antena GNSS

   La antena GNSS colocar parte superior del vehículo

2) Anemómetro

   Colocar en la parte frontal del vehículo en prueba

3) Sensor de temperatura

   Colocar fuera del vehículo

4) Sensor de corriente

   Colocar en el cable negativo o positivo de los cables de la batería de alta tensión del vehículo eléctrico.

5) Sensor de voltaje

   Mediante cables conectados a las baterías de alta tensión unir al conector Gx16-4 (conectado al interior del datalogger)

6) Sensor de presión

   Colocado en la PCB

7) Adaptador de mechero para coche

   Enchufe LED macho para cargar las baterías del datalogger

RECOMENDACIONES

- Cargar en un lapso de 3h la pila (Tipo: 18650-4800mAh-3.7V), en el caso de necesitar más tiempo de operación realizar una conexión en paralelo a cada una de las pilas de las mismas características con el fin de aumentar la capacidad del sistema de alimentación.
- Evitar rociar los sensores y el Datalogger con agua o con otro líquido.
- Constatar el correcto funcionamiento del sistema de carga.
- Prevenir la manipulación de los componentes del datalogger.
- A la hora de realizar las conexiones de los sensores de corriente y alto voltaje utilizar guantes dieléctricos.
- Evitar golpear los elementos externos del datalogger ya que en sus conexiones cuentan con elementos electrónicos.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICADO DE CUMPLIMIENTO DE LA GUÍA PARA NORMALIZACIÓN DE TRABAJOS DE FIN DE GRADO

Fecha de entrega: 26 / 01 / 2024

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: ALEXIS JAVIER GONZÁLEZ ROSILLO WILSON OSWALDO ORTIZ ESPINOZA
INFORMACIÓN INSTITUCIONAL
Facultad: MECÁNICA
Carrera: INGENIERÍA AUTOMOTRIZ
Título a optar: INGENIERO AUTOMOTRIZ
Ing. Luis Fernando Buenaño Moyano **Director del Trabajo de Integración Curricular** **Ing. Fabián Celso Gunsha Maji**