UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA:

"PLATAFORMA ABIERTA PARA DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SISTEMA EMBEBIDO"

AUTOR: DAVID PATRICIO GUZMÁN ABALCO

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR

FEBRERO 2020

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el *Repositorio Digital Institucional*, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR	
CEDULA DE IDENTIDAD:	1003671854
APELLIDOS Y NOMBRES:	GUZMÁN ABALCO DAVID PATRICIO
DIRECCION:	CAYAMBE-JUAN MONTALVO
EMAIL:	dpguzman@utn.edu.ec - deivyd-d-1993@hotmail.com
TELEFONO FUO:	022185137 TELEFONO MOVIL: 0988240782
DATOS DE LA OBRA	
TÍTULO:	"PLATAFORMA ABIERTA PARA EL DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SISTEMA EMBEBIDO"
AUTOR:	DAVID PATRICIO GUZMAN ABALCO
FECHA:	FEBRERO 2020
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN MECATRONICA
ASESOR/DIRECTOR:	CARLOS XAVIER ROSERO C.

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que se asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, 03 Febrero de 2020

David Patricio Guzmán Abalco

C.I.:1003671854

UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

En calidad de director del trabajo de grado "PLATAFORMA ABIERTA PARA EL DESARROLLO DE DISPOSITIVOS DE TARIFACIÓN VEHICULAR: SISTEMA EMBEBIDO", presentado por el egresado DAVID PATRICIO GUZMÁN ABALCO, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, Febrero de 2020

Carlos Xavier Rosero
DIRECTOR DE TESIS

UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

Yo, David Patricio Guzmán Abaleo con cédula de identidad Nro. 1003671854, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Ibarra, Febrero de 2020

David Patricio Guzmán Abaleo
C.I.: 1003671854

Agradecimiento

Agradezo a la Universidad Técnica del Norte, por ser la institución que me permitió cumplir una de las metas más importantes en mi vida académica y profesional.

A mi tutor Msc. Ing. Carlos Xavier Rosero por ser una persona paciente y brindarme su conocimiento y apoyo, quien además de ser un buen docente es una excelente persona ya que con su ejemplo y ayuda he realizado el presente trabajo.

A mis padres y hermanos por ser principal fuente de apoyo en cada momento de mi formación como profesional y mas importante aún como persona.

A mis amigos y compañeros de clase quienes han sido uno de los pilares fundamentales para salir adelante.

Finalmente a nuestra inolvidable facultad que sus aulas han sido fieles testigos de mis triunfos y fracasos en mi largo camino por alcanzar una de mis metas para así defenderme como una persona útil para la sociedad.

David Guzmán A.

Dedicatoria

A Dios, por darme la vida y estar a mi lado en cada paso que doy, por fortalecer mi corazón e iluminar nuestra mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el período de estudio.

A mis padres, por ser un ejemplo de perseverancia y humildad quienes con su sacrificio y esfuerzo me han ayudado a culminar mi carrera universitaria, y finalmente por siempre brindarme su amor y calor constante de padres y amigos.

A mi hermana y hermanos, que siempre confiaron en mí, por su apoyo y por estar conmigo, y para que vean en mí un ejemplo a seguir.

A toda mi familia y amigos porque con sus oraciones, consejos y palabras de aliento hicieron de mi una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

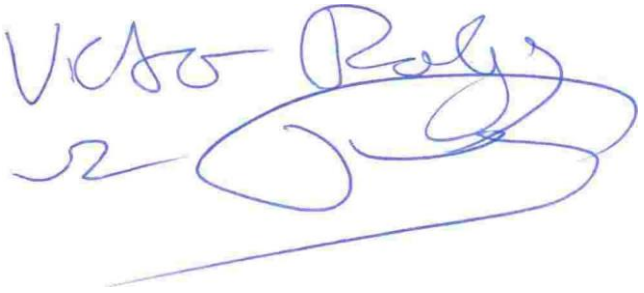
David Guzmán A.

Resumen

El presente proyecto está realizado con el objetivo de construir un dispositivo de tarifación que toma las señales del sensor de velocidad de un vehículo y las normaliza a través de un transductor, para luego ser interpretadas por el microcontrolador y efectuar el cálculo de la distancia recorrida, tiempo transcurrido y el costo de una carrera. La programación se la realiza en la plataforma de hardware libre Arduino que cuenta con un reloj en tiempo real, el cual permite que la tarifa sea automática. Además, posee una pantalla E-paper que permite una mejor visualización y un consumo bajo de energía, el dispositivo está basado acorde a los parámetros establecidos por la Agencia Nacional de Tránsito. Para el diseño y construcción del trabajo mencionado, se establecieron requerimientos funcionales y no funcionales basados en las necesidades del sistema y de los usuarios.

Abstract

This project is carried in order to build a taximeter that takes the signals of the vehicle speed sensor and normalizes these signals through a transducer, which is interpreted by the microcontroller, calculation of the joumey, elapsed time and the cost of a career. The programming is done on Arduino that has a real-time clock, which allows the fee to be automatic. In addition, it has an E-paper screen for a better visualization and low energy consumption, the device uses the parameters established by the National Transit Agency. For the design and construction of the aforementioned work, functional and non-functional requirements were established in the needs of the system and of the users.



Índice general

Índice de figuras

Índice de cuadros

Lista de Programas

Introducción

Problema

El taxímetro es un instrumento de medición y control instalado en los vehículos de servicio de transporte que indica gradualmente el valor que debe pagar el usuario [1]. Desde el ao 1982 en Ecuador se incluyó el uso obligatorio del taxímetro, sin embargo, se produjeron inconsistencias durante varios años que impedían su implementación [2]. Finalmente, en el ao 2013 la Agencia Nacional de Tránsito (ANT) resolvió el uso obligatorio del taxímetro para todas las ciudades del país [3].

Con la rápida construcción urbana, los taxis se han convertido en un medio de transporte indispensable en la vida de las personas [4]. Actualmente, en la ciudad de Ibarra existen aproximadamente 1000 taxis convencionales y 400 ejecutivos de las diferentes cooperativas y compañías [5]. El taxímetro aplicado a este medio de transporte presenta varios inconvenientes, entre ellos se encuentran el elevado costo de adquisición, constantes calibraciones debido a desconfiguración del firmware, fallas técnicas en el hardware e incumplimiento de la normativa en [3]. Es por esto que la mayoría de transportistas denuncian irregularidades en el uso del taxímetro, ocasionando que el cliente no perciba una tarifa justa [5].

Considerando la problemática citada es necesario el desarrollo de una plataforma abierta (hardware y software), de bajo costo, para experimentación sobre tarifado vehicular. Este dispositivo permitiría que en la academia se investigue acerca de algoritmos de tarifación. Sería el

primer paso para el desarrollo de taxímetros que cumplan con la normativa en [3] y puedan ser certificados.

Objetivos

El objetivo principal de este proyecto consiste en desarrollar una plataforma de hardware libre para experimentación sobre tarifación vehicular.

Los siguientes objetivos específicos son también realizados:

- Determinar las características de hardware de los taxímetros existentes en el medio.
- Diseñar el hardware de la plataforma en función del desempeño requerido.
- Desarrollar el programa que regirá al hardware en base a la funcionalidad estipulada.
- Validar la funcionalidad del hardware.

Alcance

Esta plataforma se desarrollará con software y hardware libre, y su funcionalidad será la mencionada en la norma INEN en [3]. Los ensayos para certificación con la norma se realizarán en un trabajo posterior ya que se encuentran fuera del enfoque del presente desarrollo. El taxímetro recibirá la señal de distancia y velocidad desde un transductor para el efecto, medirá el tiempo empleado en una carrera (incluso cuando el vehículo está detenido), y calculará el costo y los totales del recorrido en base a tarifas especificadas.

Para mejor entendimiento de las características, se recomienda referirse a [3].

Las pruebas de funcionamiento del hardware se las realizará en un automóvil.

Justificación

La importancia de esta investigación radica en solucionar la problemática que actualmente sufren los propietarios de taxis luego de la adquisición e instalación de un taxímetro en sus unidades. En el país un taxímetro e impresora se venden por separado y a un costo elevado en relación a las prestaciones que ofrecen. Por tal motivo en esta investigación se desarrollará un taxímetro de bajo costo, que incluya mecanismos para obtener medidas fiables y consistentes con un mínimo error, disminuyendo así las constantes calibraciones.

El desarrollo de esta plataforma abierta permitiría establecer los parámetros de configuración del taxímetro de manera inalámbrica mediante una interfaz gráfica que reemplazaría a los métodos tradicionales que se usan actualmente, los cuales implican el desmontaje del equipo resultando incómodo para el propietario del medio de transporte.

Además, esta plataforma servirá como base para realizar posteriores modificaciones de hardware y software que permitirían obtener taxímetros homologados cumpliendo los parámetros establecidos en la norma INEN en [3].

Capítulo 1

Revisión Literaria

1.1. Taxímetro

El término Taxímetro proviene de las palabras griegas “taxi”que significa tasa y “metro”que significa medir; es decir mide la tasa o costo que tiene el servicio. Es un dispositivo de medida parecido a un odómetro, pero en este caso mide la tarifa a cobrar por el servicio prestado en base a la distancia recorrida y al tiempo transcurrido [6].

1.2. Normativa Vigente

Desde el año 1982 en Ecuador se incluyó el uso obligatorio del taxímetro, sin embargo, se produjeron inconsistencias durante varios años que impedían su implementación [2]. Finalmente, en el año 2013 la Agencia Nacional de Tránsito (ANT), resolvió el uso obligatorio del taxímetro para todas las ciudades del país, ya sea bajo la modalidad convencional o ejecutiva [1].

El Instituto Ecuatoriano de Normalización (INEN) será el encargado de emitir el certificado de aprobación del modelo luego de realizar la verificación de ensayos y cumplimiento de la normativa. De conformidad con la norma INEN en [3], establece que el taxímetro debe cumplir con los siguientes parámetros metrológicos y técnicos:

- Incluyan un reloj.
- Un contador de eventos.
- Interfaz de usuario.
- Los parmetros deben ser programables.
- Nmero de identificacin del taxi.
- Sensor que proporcione datos para el clculo de la distancia recorrida.
- Programa legalmente relevante. [3]

1.3. Antecedentes

La ANT en [7] describe los diferentes modelos de taxímetros homologados en el país con las respectivas empresas proveedoras de estos dispositivos de medición, control y seguridad, como se detalla en la Tabla 1.1

Tabla 1.1: Taxímetros Homologados

NUM	MARCA	MODELO
1	VIRTUALTEC	VIRLOC 10
2	DIGITAX 550	M07
3	DIGITAX 550	M11
4	LAKEDRIVER	SLIM-800
5	TAXSYM	M09
6	DIGITAX	F3 PLUS
7	DIGITAX	M1 PLUS
8	SEMSA	PLATINUM
9	INTELLIGENT TAXIMETER	D10
10	DIVUTAXI	I700-3G
11	STALIN-TAX	ST-11
12	CENTRODYNE	S700
13	OPTRONIC	TX-10
14	FUL-MAR	TANGO XP
15	ARIEL TAX	MILENIO
16	ATA PRIMUS	RS-01
17	ATA PRIMUS	S-01
18	TAXITRONIC	TX-40
19	TAXITRONIC	TX-80 SKYGLASS

1.4. Taxímetros autorizados en el Ecuador

Los taxímetros presentados en la Tabla 1.1 varían dependiendo de las funcionalidades que poseen, sin embargo, la pantalla de visualización, impresora, navegación, interacción con el usuario y el tipo de comunicación que utilizan para la configuración de parámetros son similares. A continuación, se detallan las funcionalidades y parámetros de algunos de ellos.

1.4.1. Taxímetro Tango XP

Este dispositivo perteneciente a Full-Mar S.A., es uno de los taxímetros que se encuentran homologados en el Ecuador. Cuenta con un panel visual de 6 displays para el valor de la carrera y 5 displays para suplementos y extras. Cuenta con una botonera de 5 pulsadores que sirven para desplazarse entre las diferentes funciones del taxímetro. Incorpora una impresora térmica

de alta velocidad de 24 caracteres, y posee 6 tarifas independientes horarias y automáticas. [8].

Beneficios

- Bandera de led de alta luminosidad incorporada.
- Registro de los últimos 12 meses de recaudación.
- Localización y despacho de viajes en tempo real.
- Registro de los últimos 5 minutos de velocidad.
- Conexión con post para pago con tarjeta de crédito y débito.

Especificaciones

Tabla 1.2: Características Tango XP

Item	Parámetros
Fabricante	FUL-MAR S.A.
Voltaje de funcionamiento	9,5 15 Vcc
Dimensiones	160*90*70/85mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	-10˚ C /+70˚ C
Altura de dígitos de lectura	13mm
Peso	470 gr
Puertos de comunicación	Can Bus, K line
Precio	260.00$

Figura 1.1: Taxímetro Tango XP [8]

1.4.2. Taxímetro D10

El taxímetro D10 perteneciente a la empresa Sumitrag, contiene un panel visual de 6 displays el cual muestra la hora cuando no está en uso el dispositivo y el importe cuando lo está, contiene 2 botones para su navegación. La tarifa es de accionamiento automático y lleva un reloj en tiempo real incorporado internamente. El taxímetro y la impresora se adquieren por separado [9].

Especificaciones

Tabla 1.3: Características D10

Item	Parámetros
Fabricante	SUMITRAG
Voltaje de funcionamiento	9-16 Vcc
Dimensiones	95*30*45 mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	Si
Altura de dígitos de lectura	15mm
Peso	100 gr
Puertos de comunicación	RS232
Precio	280.00$

Figura 1.2: Taxímetro D10 [9]

1.4.3. Taxímetro F3 Plus

Este taxímetro tiene en la parte frontal 5 teclas denominadas: OP, K1, K2, K3, K4, utilizadas para la selección y desplazamiento de funciones. La pantalla de visualización tiene un total de 11 displays. Este taxímetro permite el registro de 2 conductores mediante el ingreso de un código de usuario. Además, presta un servicio de pausa para casos fortuitos como: la reposición del combustible, cambio de neumático y contestación de llamada. Incluye una impresora la cual si no esta conectada al taxímetro o sin su rollo de papel no funciona [10].

Beneficios

- 64 tarifas inteligentes independientes automáticamente gestionadas por hora y fecha.
- Reloj calendario autoalimentado.
- 45 contadores para conductores y propietarios.
- Dos puertos serie RS232.

Especificaciones

Tabla 1.4: Características F3 Plus

Item	Parámetros
Fabricante	TAXSYM
Voltaje de funcionamiento	8,5 - 16 Vcc
Dimensiones	18,5*6,2*2,5 mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	-15° C a 70° C
Altura de dígitos de lectura	15mm
Peso	350 gr
Puertos de comunicación	COM1 y COM2
Precio	438.70$

Figura 1.3: Taxímetro F3 Plus [10]

1.4.4. Taxímetro TX-10

Este dispositivo diseñado y construido en Ecuador por la empresa Electrónica Industrial perteneciente a Optronic CIA. LTDA. Es multitarifario, es decir, se puede operar con tarifas diferentes dependiendo de la hora y fecha del día. Posee indicadores y pantallas luminosas tanto para el conductor como para el pasajero, además contiene un reloj en tiempo real interno con hora y fecha. Facilidades de calibración en recorrido para establecer el cobro exacto del valor por kilómetro. Interfaz de comunicaciones para controlar dispositivos periféricos, tales como: Impresora, GPS, lectores de tarjetas, etc [11].

Beneficios

- Conexión con dispositivos externos como el indicador luminoso del taxi cuando está ocupado.
- Capacidad de imprimir facturas, de acuerdo a las normas fijadas por el SRI.
- Número de serie electrónico único de 5 dígitos para su identificación.
- Capacidad de utilizar sellos de seguridad para garantizar la inviolabilidad del sistema electrónico.

Especificaciones

Tabla 1.5: Características TX-10

Item	Parámetros
Fabricante	OPTRONIC CIA. LTDA
Voltaje de funcionamiento	8 - 24 Vcc
Dimensiones	92,2*57,6*25,6 mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	0˚ C a 70˚ C
Altura de dígitos de lectura	15mm
Peso	100 gr
Puertos de comunicación	Serial TTL
Precio	210.00$

Figura 1.4: Taxímetro TX-10 [11]

1.4.5. Taxímetro RS-01

El taxímetro retrovisor PRIMUS RS-01 es un producto fuerte, de diseño simple y discreto. Pasa inadvertido en la cabina del vehículo fijándose directamente sobre cualquier retrovisor original. Cuenta con 4 botones pulsadores para su navegación y su información está repartida en 4 zonas: Zona de visualización de pictogramas, Zona de visualización de información de estados, Zona de visualización del importe de los suplementos, Zona de visualización del precio. Además, tiene una selección automática de la tarifa, un pasaje automático a la hora verano/invierno y se apaga automáticamente en caso de inactividad prolongada [12].

Beneficios

- Compatible con la aplicación 1001TAXIS.
- Selección automática de la tarifa día/noche.
- Gestión y selección manual de los suplementos.
- Pasaje automático a la hora verano/invierno.
- Visualización automática del mínimo de percepción

Especificaciones

Tabla 1.6: Características RS-01

Item	Parámetros
Fabricante	ATA
Voltaje de funcionamiento	9 - 16 Vcc
Dimensiones	278*82*28 mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	-25° C a +70° C
Altura de dígitos de lectura	12mm
Peso	280 gr
Precio	250.00$

Figura 1.5: Taxímetro RS-01 [12]

1.4.6. Taxímetro TX40

El taxímetro TX40, tiene un diseño compacto con un innovador display de leds blancos y azules. Posee un teclado clásico de Taxitronic con 8 botones los cuales están iluminados para trabajar en todas las condiciones. Un equipo de dimensiones reducidas con un control horario opcional, cambios de tarifa automáticos por tiempo, distancia e importe, visualización e impresión de los totalizadores y con un código secreto antirrobo. Además, presenta un sistema básico de hasta 10 turnos y control de conductores [13].

Beneficios

- Control horario opcional.
- Vinculado por Bluetooth con la aplicación Android Smart TD en tu smartphone o tablet.
- Capacidad para 32 tarifas completas e independientes.
- Sistema básico de hasta 10 turnos y control de conductores.

Especificaciones

Tabla 1.7: Características TX40

Item	Parámetros
Fabricante	TAXITRONIC
Voltaje de funcionamiento	8 - 16 Vcc
Dimensiones	180*50*32 mm
Tarifa diurna y nocturna	Si
Resistencia mínima a los cambios de temperatura	-25˚ C a +70˚ C
Altura de dígitos de lectura	13mm
Peso	187 gr
Precio	280.00$

Figura 1.6: Taxímetro TX40 [13]

1.5. Funcionalidad de un taxímetro

1.5.1. Distancia

Para que el taxímetro convierta en una tarifa cuantificable la distancia recorrida y el tiempo empleado (incluido cuando está parado), el sistema de medición se sirve de unos impulsos eléctricos que da el transductor del vehículo, un sensor unido a la transmisión que ofrece datos al taxímetro sobre la velocidad y el número de kilómetros recorridos. Estos impulsos se dan cuando se ha rodado una distancia que se ha definido previamente en el taxímetro [15].

1.5.2. Tiempo

Se entiende que, si un taxi está atrapado en el tráfico sin moverse, el taxi sigue trabajando, por lo que también se cuantifica ese tiempo de espera cuando está detenido. Esto es mucho más sencillo, ya que lo único que hay que hacer es establecer una medida de tiempo en el taxímetro, que irá añadiendo, de nuevo mediante impulsos eléctricos, un valor a la cantidad final [15].

1.5.3. Principio de operación

Un transductor analógico envía impulsos eléctricos con una frecuencia proporcional a la velocidad del taxímetro, el cual es controlado por el programa que calcula la velocidad y transmite el valor de la cantidad en un datagrama a través de un bus de datos para el taxímetro [3]. El transductor que utilizan los taxímetros comerciales se visualiza en la figura 1.7.

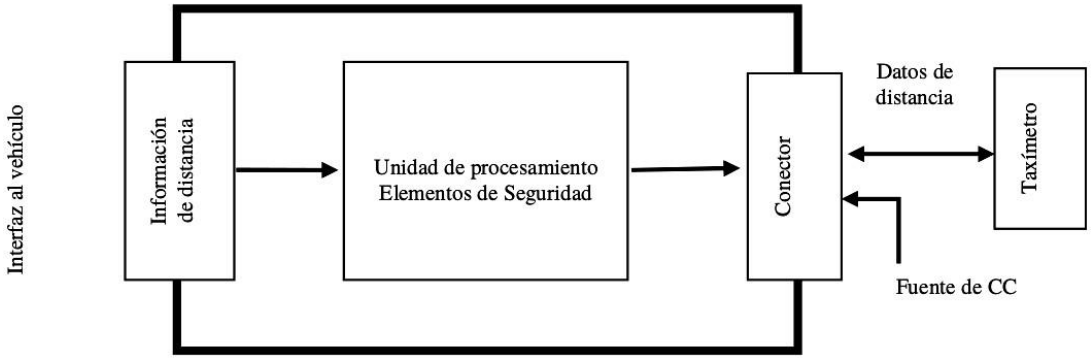


Figura 1.7: Transductor típico de medición de distancia. [3]

1.6. Estados de Funcionamiento

El taxímetro posee varias posiciones o estados de funcionamiento, cada uno presenta un comportamiento especial. La siguiente figura presenta un cuadro con los ciclos de los estados del taxímetro y su secuencia [5].

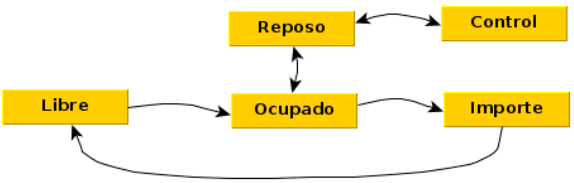


Figura 1.8: Estados de funcionamiento de un taxímetro. [3]

a). Libre: Posición de funcionamiento en la que el taxímetro no está calculando un costo ni hay un cliente que esté realizando un viaje al interior del taxi.

b). Ocupado: En este estado el taxímetro indica y calcula el costo basado en un posible costo inicial y una tarifa correspondiente al tiempo del viaje y/o la distancia desplazada.

c). Importe (A Pagar): Se indica el valor que tiene que cancelar el pasajero, también se imprime el recibo en el que se visualiza información como el valor por la distancia recorrida, la tarifa utilizada y el tiempo de espera transcurrido durante el viaje.

Para cambiar los estados se lo realiza a través de palancas, botones o interruptores, su operación es muy fácil. Un taxímetro en funcionamiento normal, repite cíclicamente las siguientes etapas: Libre, Ocupado e Importe [2].

1.7. Tarifas

Para tener un buen funcionamiento del taxímetro se debe fijar las tarifas vigentes en las diferentes modalidades, las cuales son establecidas por la Comisión Nacional de Tránsito, Transporte Terrestre y Seguridad Vial. En el Ecuador únicamente se usan dos tarifas, diurna y nocturna, cada una tiene diferente precio de arranque (precio inicial de la carrera, que el taxímetro muestra al empezar el servicio de taxi), costo por minuto de espera y por kilómetro recorrido [5]. La tabla 1.8 refleja los costos con los que los taxímetros son calibrados en el Ecuador.

Tabla 1.8: Cuadro tarifario de taxis. [16]

Parámetros	Diurno($)	Nocturno($)
Arranque	0.40	0.44
km. Recorrido	0.40	0.40
Minuto de espera	0.08	0.09
Carrera mínima	1.25	1.50

El horario nocturno comprende desde las 19h00 hasta las 06h00. Para los fines de semana y feriados se aplica los valores de la tarifa nocturna.

Capítulo 2

Diseño del Hardware

2.1. Requerimientos del Sistema

Representan las necesidades de los clientes y las personas involucradas en el desarrollo del sistema. Estos requisitos describen lo que un sistema debe hacer, informacion específica sobre los servicios que proporciona y las restricciones con las cuales debe operar. Se clasifican en funcionales y no funcionales [18].

2.1.1. Requerimientos Funcionales

Están rigurosamente relacionados con las entradas, las salidas de cada proceso y el almacenamiento de información en el sistema [18].

- El microcontrolador y la interfaz de usuario se comunican inalámbricamente mediante un módulo wifi para visualizar y configurar los diferentes parámetros del taxímetro establecidos en la normativa INEN en [3].
- El hardware utiliza una plataforma de uso libre que permite el diseño y programación del mismo a través de la consola de Linux.
- El microcontrolador posee un subsistema para definir la hora y fecha actual del módulo

RTC (Real-Time-Clock) que se comunica con la interfaz y los cambios se lo puede realiar a través de la interfaz gráfica o automáticamente tomando como referencia la hora del ordenador.

2.1.2. Requerimientos no Funcionales

Surgen de las necesidades del usuario y representan las restricciones que el sistema debe tener, no se relacionan directamente con los servicios específicos, sin embargo el incumplimiento de estos requisitos haría que el sistema no funcione de la manera esperada o fuera inútil [18].

- Usabilidad.-El hardware es de fácil operación, posee un panel de 6 botones los cuales permitirán el ingreso a los diferentes estados del taxímetro.
- Portabilidad.- Usar herramientas y lenguajes basados en software libre dan como resultado una instalación sencilla, sin costo y de fácil mantenimiento. Esto facilita para realizar mejoras, corregir errores y adaptar a plataformas sin costo alguno.
- Económico.- Su adquisición es mucho más accesible en comparación de otros taxímetros del mercado debido a su bajo costo.

2.2. Diagrama de bloques

El sistema que el taxímetro maneja está compuesto de partes físicas o hardware; cada una de estas partes cumple con una función específica. Para lograr los objetivos de este proyecto dichas partes están conectadas de tal forma que existe interacciones entre la mayoría de ellas y el microcontrolador ó interacciones únicamente entre ellas; cada interacción puede realizarse en un solo sentido y en doble sentido, esto debido a que existen dispositivos de entrada, salida, o bidireccionales [19].

El sistema está formado por los elementos que se visualizan en la Figura 2.1, en el cual se ha identificado los bloques de las diferentes etapas y procesos, las flechas muestran la direccin del flujo de datos.

Figura 2.1: Diagrama de bloques

2.3. Microcontrolador

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica [20].

A continuación en la Tabla 2.1, se detallan las características de tres tipos diferentes utilizadas para el desarrollo de aplicaciones electrónicas en hardware libre.

Tabla 2.1: Características Arduino Mega 2560 y Rasberry Pi3

Características	Arduino Mega 2560	Rasberry Pi3	Arduino Nano
Precio	15 dólares	60 dólares	10 dólares
Velocidad de reloj	16 MHz	700 MHz	16MHz
Multitarea	No	Si	No
Voltaje de entrada	7 a 12V	5V	7 a 12V
Memoria Flash	256KB	SD (2 a 16GB)	32KB
Memoria SRAM	8KB	1GB	2KB
Memoria EEPROM	4KB	No Tiene	1KB
Entorno de desarrollo integrado	Arduino	Scratch, IDLE	Arduino

Como se observa en la Tabla 2.1, el arduino mega y el arduino nano cumplen con las especificaciones básicas que permiten el funcionamiento de taxímetro. Sin embargo, el uso de librerías de la pantalla requiere de una cantidad mínima de memoria, lo que hace que el arduino nano no sea factible para esta aplicación, por otro lado la raspberry pi tiene características muy avanzadas que serán un desperdicio de recursos para el grado de complejidad que requiere el taxímetro.

2.4. Alimentació n y regulación de voltaje

El taxímetro propuesto está basado en sistemas microcontrolados, y al ser una aplicación para ser instalada en un vehículo, utiliza como fuente de voltaje continuo una batería de 12 voltios, por lo que se hace imprescindible la construcción de un sistema de regulación de voltaje a 5 voltios de corriente continua.

En esta aplicación existirán factores de interferencia y variación de voltaje a los que va a estar sometido el microcontrolador, por ello se implementará un circuito de regulación y de protección de voltaje, lo más usual es armar un circuito con un regulador de voltaje a 5 voltios, además en el circuito deberá estar presente un fusible, un filtro pasa bajos, diodos, entre otros, todo con la finalidad de que el circuito proteja la integridad elctrica del micro controlador, y as permitir al sistema que funcione correctamente.

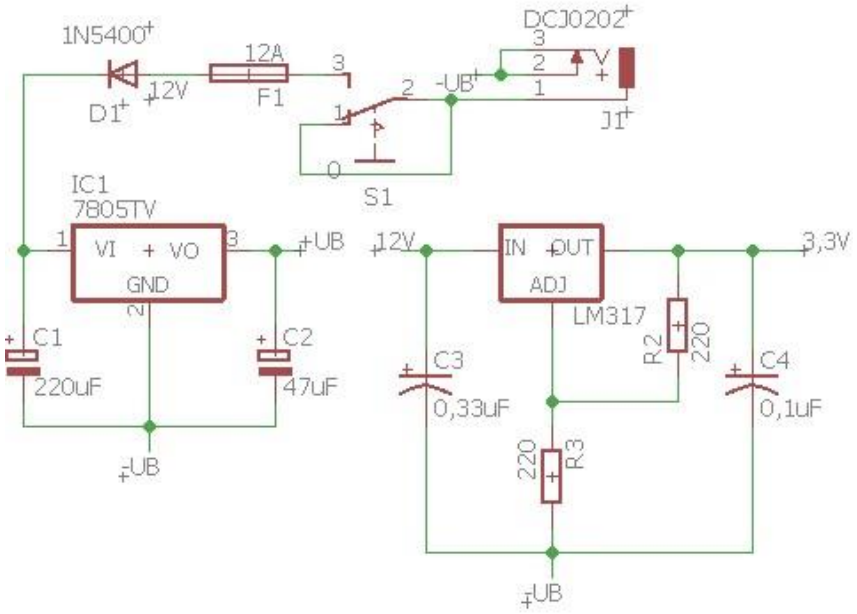


Figura 2.2: Circuito de regulación de voltaje

La Figura 2.2 muestra el circuito que se usará en el taxímetro para la regulación y protección de voltaje, cuya fuente de poder es la batería de vehículo.

2.5. Sensor de velocidad (VSS)

Es el dispositivo que se encarga de transmitir la información de la velocidad del vehículo, y envía esta señal a la unidad de control de motor (ECU) [21].

El taxímetro está diseñado para aceptar señales digitales de entre 0 voltios y 5 voltios, las cuales son interpretadas por el microcontrolador para conocer la distancia que recorre, dicha distancia será usada para calcular el precio del servicio.

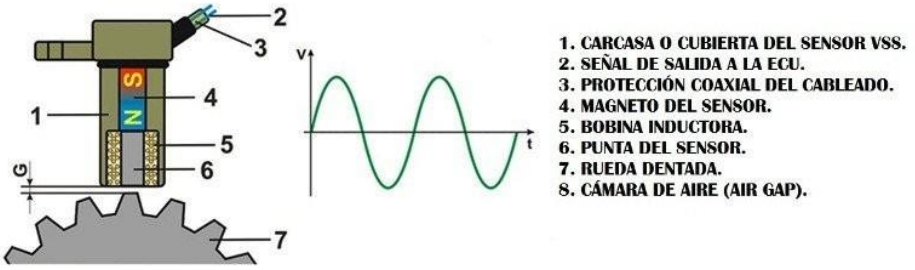


Figura 2.3: Diagrama del sensor VSS [21]

En la actualidad los sensores de velocidad en su mayoría son de efecto hall y envían una señal digital entre 0 y 10 voltios garantizado la calidad de la señal digital siempre (aún a bajas velocidades), que genera una forma de onda con una frecuencia que es proporcional a la velocidad del vehículo.

Las caractersticas de estos sensores crean la necesidad de implementar un circuito de acoplamiento entre el sensor y el taxmetro, para lo cual se us un transductor.

2.5.1. Transductor

El transductor es un circuito que permite transformar señales con unas características a otras señales pero normalizadas [22].

Este transductor convierte pulsos digitales de voltaje de 10 voltios y de baja intensidad de corriente, en pulsos digitales de voltaje de 5 voltios a una corriente estndar que ser la seal de entrada para que el microcontrolador calcule la distancia recorrida.

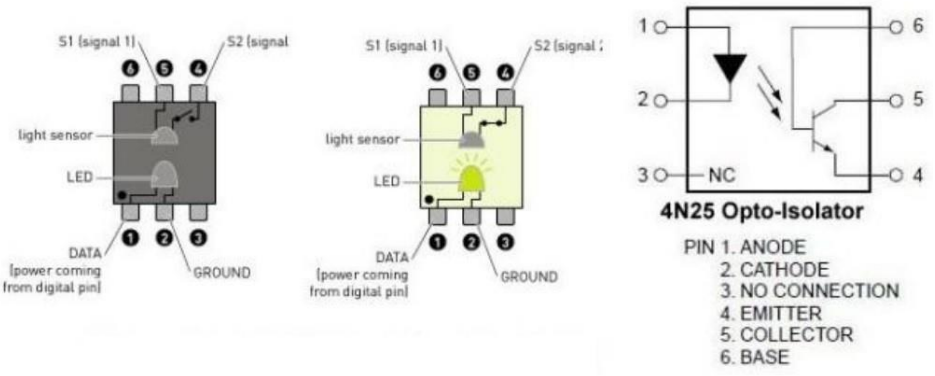


Figura 2.4: Optoacoplador 4N25 [23]

El optoacoplador cumple con la función de acoplar el sistema de 10 voltios del sensor VSS al sistema microcontrolado el cual funciona a 5 voltios, además lo aísla eléctricamente, pero no es suficiente para el funcionamiento del taxímetro, para lo que se hace necesaria la implementación de un sistema extra para que ninguno de los sistemas se vea atrofiado por la densidad de corriente. El sistema extra se conecta al optoacoplador antepuesto como se observa en la Figura 2.5.

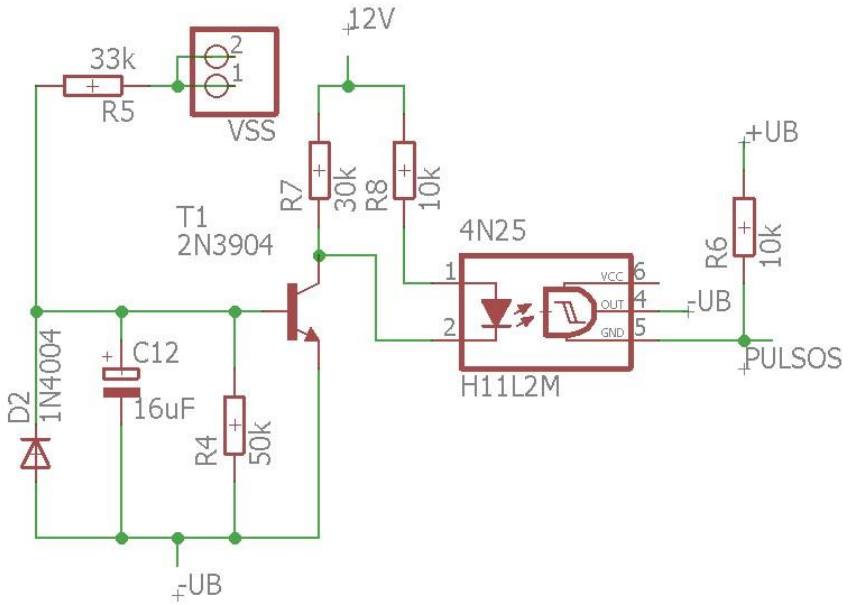


Figura 2.5: Sistema de acoplamiento de señales

Este sistema cumple con la función de tomar las señales digitales del sensor de efecto Hall de entre 0 y 10 voltios, la señal pasa por un transistor para que la corriente de la señal no se vea afectada al conectar el taxímetro.

2.6. Pantalla

La pantalla es un dispositivo de salida que representa visualmente la informaciíon. Este dispositivo permite la comunicación máquina humano, ya que puede mostrar mensajes de funcionamiento de las mismas. Existe una amplia gama de pantallas para el desarrollo de proyectos de hardware, entre ellas destacamos las más comunes: monocromática, led, cristal líquido y plasma.

2.6.1. Pantallas monocromáticas

Una pantalla es monocromática cuando sólo tiene dos colores. En realidad es un único color y el color del fondo (usualmente negro o gris oscuro). A continuación en la Tabla 2.2. se detallan las características de dos pantallas utilizadas para el desarrollo de aplicaciones electrónicas en hardware libre.

Tabla 2.2: Características GLCD 2.8’ y E-Paper SPI 2.9’

Características	GLCD 2.8’	E-Paper SPI 2.9’
Precio	15 dólares	18 dólares
Interfaz	Paralela de 8 bits, 4 bits y serial	SPI (3 o 4 cables)
Tamaño	70*45*1.40 mm	79*36.7*1.05 mm
Resolución	128x64 píxeles	296x128 píxeles
Voltaje de operación	5V	3.3V
Color	Azul y blanco	Negro y blanco
Peso	10 g	4 g

Para el desarrollo de este proyecto se elige la pantalla E-paper, ya que, de acuerdo a la Tabla 2.2, el consumo de energía es muy bajo, la resolución es mayor a los 150 dpi, tiene un peso ligero y además posee un brillo que no lastima los ojos con su visualización prolongada.

e-Paper	SPI	Info
3.3V	-	-
GND	-	-
DIN	MOSI	-
CLK	SCK	-
CS	-	LOW seleccionado
DC	-	Datos/Comandos HIGH para datos, LOW para comandos
RST	-	LOW activado
BUSY	-	LOW activado

Figura 2.6: Pantalla E-paper 2.9 inch [24]

2.7. Reloj en tiempo real (RTC)

Es un reloj que funciona con una batería y le permite ahorrar tiempo, incluso cuando se produce un corte de energía. Usando un RTC, se puede hacer un seguimiento de las largas líneas de tiempo, incluso si se reprograma su microcontrolador o se lo desconecta del USB o de una fuente de alimentación [25].

Tabla 2.3: Características RTC DS1307 y DS3231

Características	DS1307	DS3231
Precio	2 dólares	3.50 dólares
Comunicación	I2C	I2C
Exactitud	+3ppm	+2ppm
Voltaje de operación	4.5 a 5.5V	3 a 5.5V
Batería	Si	Si
Sensor de temperatura	No	Si

De acuerdo a la Tabla 2.3, el modelo DS1307 no posee un sensor de temperatura, por tanto tiene variaciones de temperatura que afectan a la medición y provocan un desfase temporal, por otro lado, el DS3231 si posee dicho sensor y además incorpora medición y compensación de la temperatura.

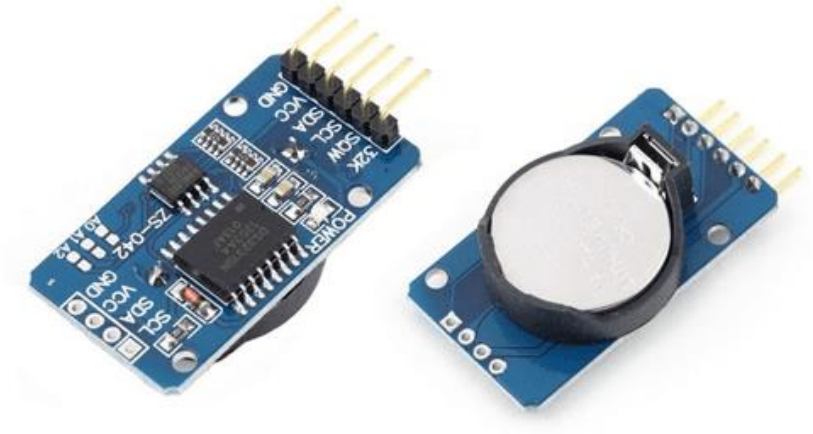


Figura 2.7: Módulo RTC DS3231 [25]

2.8. Wi-Fi

Wi-Fi (Wireless Fidelity) es un conjunto de estandares de la IEEE 802.11 creadas para redes locales inalámbricas, las cuales se utilizan para acceso a internet y redes privadas.

2.8.1. Módulo Wi-Fi

Se trata de un chip integrado con conexión Wi-Fi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red. A continuación se presenta dos tipos de modulos Wi-Fi [26].

Tabla 2.4: Características Módulos ESP8266 y ATSAMW25

Características	ESP8266	ATSAMW25
Precio	4.25 dólares	30.81 dólares
Protocolo	802.11 b/g/n	802.11 b/g/n
Rango de frecuencia	WiFi 2.4 GHz	WiFi 2.4 GHz
Seguridad	WPA/WPA2	WPA/WPA2, TLS, SSL
Voltaje de operación	2.5 a 3.6V	2.7 a 3.6V
Protocolo de red	IPv4, TCP/UDP/HTTP	DHCP, DNS, TCP/IP (IPv4), UDP, HTTP, HTTPS

Acorde a la Tabla 2.4, los dos módulos poseen similares características y para la selección de este módulo nos basaremos principalmente en el precio ya que este proyecto esta enfocado en su bajo costo.

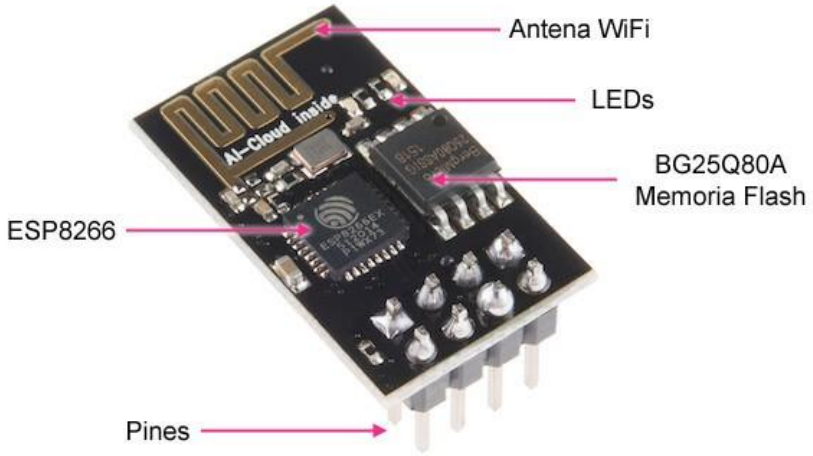


Figura 2.8: Módulo Wi-Fi ESP8266-01 [27]

2.9. Impresora Térmica

La impresora es térmica, pues no necesita de tinta para la impresión de los recibos por lo que disminuye costos de operación del dispositivo. Es ideal para la interfaz con un microcontrolador que tiene salida de 3.3-5V con comunicacin serie TTL para imprimir texto, códigos de barra, gráficos de mapa bits [28].

El panel trasero consta de dos conectores de tres pines: uno para la alimentación y el otro para la comunicación serial con velocidad de transmisión de 19200 bps. Esta no puede funcionar con la alimentación que proporciona el arduino por lo que requiere de una corriente externa de 5-9V con 2 amperios para generar el calor necesario y realizar la impresión. Las características técnicas de la mini impresora térmica son las que se detallan a continuación:

Tabla 2.5: Características Impresora Térmica TTL RS232

Item	Parámetros
Protocolo	TTL Serial, 19200bps
Modo de impresión	Impresora de línea térmica
Ancho de impresión efectivo	48mm
Velocidad de impresión	50-80 mm/seg
Resolucin	8 puntos/mm, 384 puntos/línea
Alimentacin	5 9 VDC @ 1.5A (Máximo durante la impresión)
Fonts	ANK: 5x7, Chino: 12x24, 24x24
Dimensiones exteriores	111x65x57 mm

Figura 2.9: Impresora Térmica TTL RS232 [28]

2.10. Almacenamiento de datos

El almacenamiento de datos es la retención de información mediante el uso de tecnología desarrollada especialmente para guardar esos datos y mantenerlos lo más accesibles posible [29]. En informática el almacenamiento de datos es usualmente digital y existen varios dispositivos o soportes empleados para almacenar datos como por ejemplo: Blu-Ray, CD-ROM, SD, Flash Drive, etc.

2.10.1. Módulo Micro SD

Permite conectar a un microcontrolador una tarjeta de memoria microSD para almacenar grandes cantidades de datos. La Tabla 2.6 muestra las características del módulo micro SD.

Tabla 2.6: Características Módulo Micro SD Arduino

Item	Parámetros
Voltaje de operación	3.3 a 5V
Interfaz	SPI
Peso	5 g
Dimensiones	42x24x12 mm
Consumo de corriente	200mA
Soporte	microSD (hasta 2G) y micro SDHC (hasta 32G)

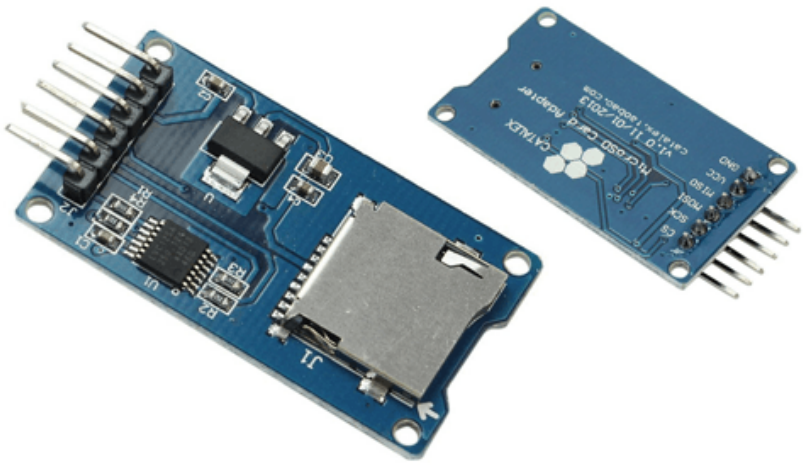


Figura 2.10: Módulo micro SD para arduino [29]

Capítulo 3

Diseño del Firmware

3.1. Flujogramas del software

El diseño del software que realiza el control, cálculo y tarifación del servicio de taxi, se lo concibe a partir de una máquina de estados finitos(autómata finito) [30]. El autómata finito del taxímetro se muestra en la Figura 3.2, dentro de la misma existen tres estados:

- Libre
- Tarifación
- Impresión

Para ingresar en estos estados existen seis botones que estan distribuidos como se lo muestra en la Figura 3.1 y cumplen con una función específica, como se detalla en la Tabla 3.1.

Tabla 3.1: Funcionalidad de botones

Botón	Funcionalidad
Ok	Ingresa o activa alguna función del dispositivo.
Atrás	Regresa a la pantalla anterior, normalmente la más reciente.
Arriba	Mueve el cursor o la selección hacia arriba.
Abajo	Mueve el cursor o la selección hacia abajo.
Izquierda	Mueve el cursor o la selección hacia la izquierda.
Derecha	Mueve el cursor o la selección hacia la derecha.

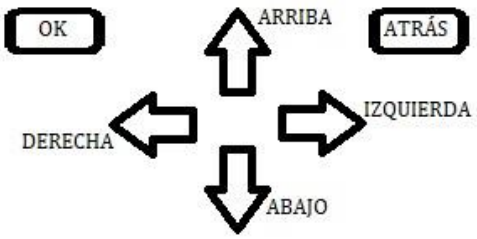


Figura 3.1: Diagrama de botones

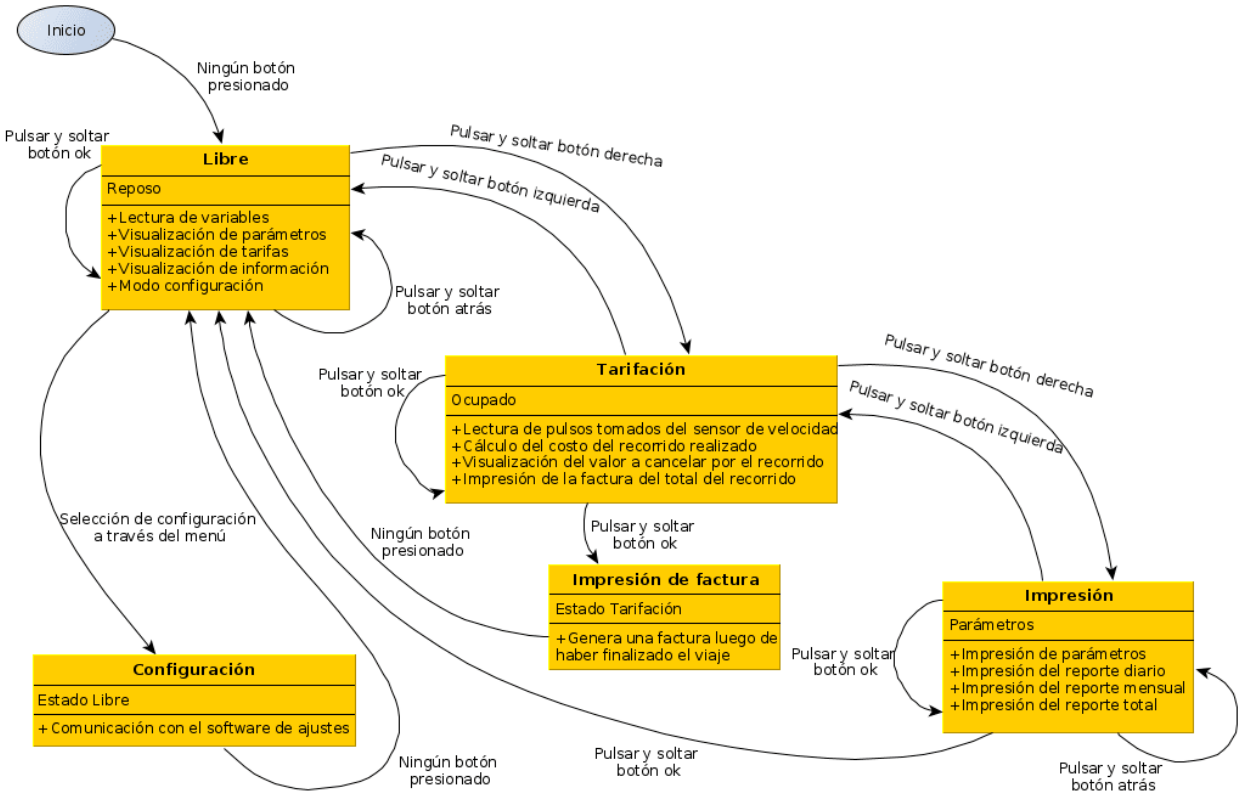


Figura 3.2: Diagrama de estados del taxímetro

Cada estado se entiende como un programa independiente que cumple con un objetivo y se ejecutan dependiendo el estado de funcionamiento en el que se encuentra el taxímetro. Éstas se explican detalladamente en los siguientes literales.

3.1.1. Estado Libre

Al iniciar un sistema microprocesado siempre se definirán variables y sus respectivos valores, se configurarán puertos, etc. Este estado estará selecionado por defecto al encender el taxímetro. Para ingresar en él se debe presionar el botón ok y se podrá observar las opciones de las mencionadas varibles del taxímetro como: parámetros, tarifas, información general y además el ingreso al modo configuración a través de la conexión Wi-Fi, definidos a continuación.

- Información general.- Datos principales del propietario y del vehículo.
- Parámetros.- Valores que dependen del tipo de taxímetro, como la identificación de programa y valores utilizados para el cálculo del costo y redondeo.
- Tarifas.- Monto fijado por el municipio de cada ciudad mediante una ordenanza para el cobro del servicio que presta el medio de transporte a la ciudadanía.
- Modo configuración.- El día de la próxima inspección, lectura del último mantenimiento y la configuración de fecha y hora actual de forma manual o automática.

El flujograma de la Figura 3.3. indica el funcionamiento de la subrutina Libre.

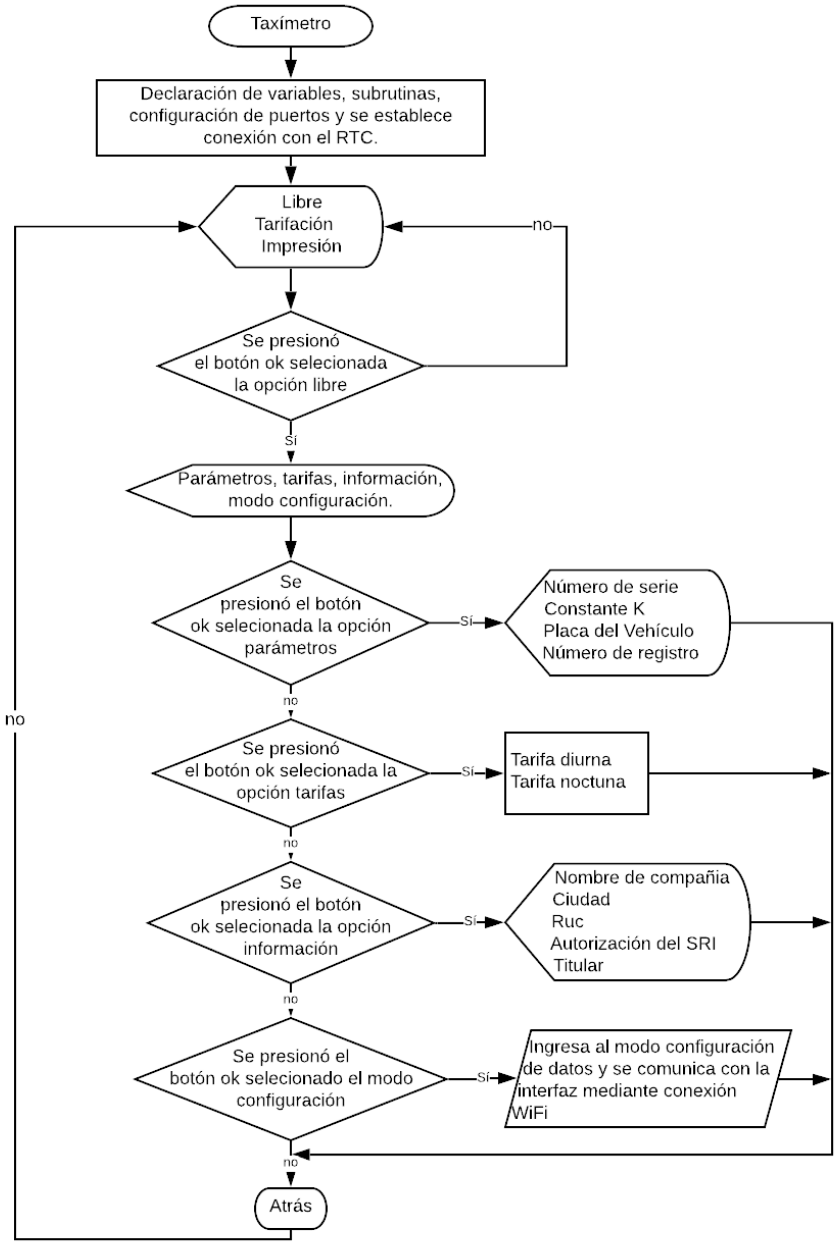


Figura 3.3: Flujograma de estado libre

La composición de cada una de las opciones contiene ítems de información con respecto a cada parámetro establecido, y solamente son configurables a través del software de ajustes.

3.1.1.1. Opción Tarifas

El taxímetro está compuesto por dos tarifas: Tarifa diurna y tarifa nocturna, que cambian automáticamente dependiendo de la hora del día. Esta opción permite visualizar los valores que incluyen las tarifas de acuerdo al reglamento que se establece en la norma Inen en [3] y solo puede ser modificada mediante el software de ajustes por las entidades autorizadas de regulación. La Figura 3.4 muestra los parámetros que se visualizan en esta opción.

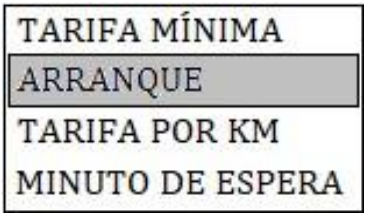


Figura 3.4: Pantalla de opción tarifa

3.1.2. Estado Tarifación

La subrutina de tarifación inicia encerando las variables de distancia, tiempo de espera y el valor del importe del servicio. A continuación, muestra en pantalla los valores iniales(tarifa) de la carrera y posteriormente calcula la velocidad del vehículo, si es ésta es mayor o igual a 12 km/h se realiza el cobro de por kilómetro recorrido, caso contrario, si es menor a 12 km/h se realiza el cobro por tiempo de espera según los parámetros en [3]. El proceso que realiza la subrutina tarifación se muestra en la Figura 3.5.

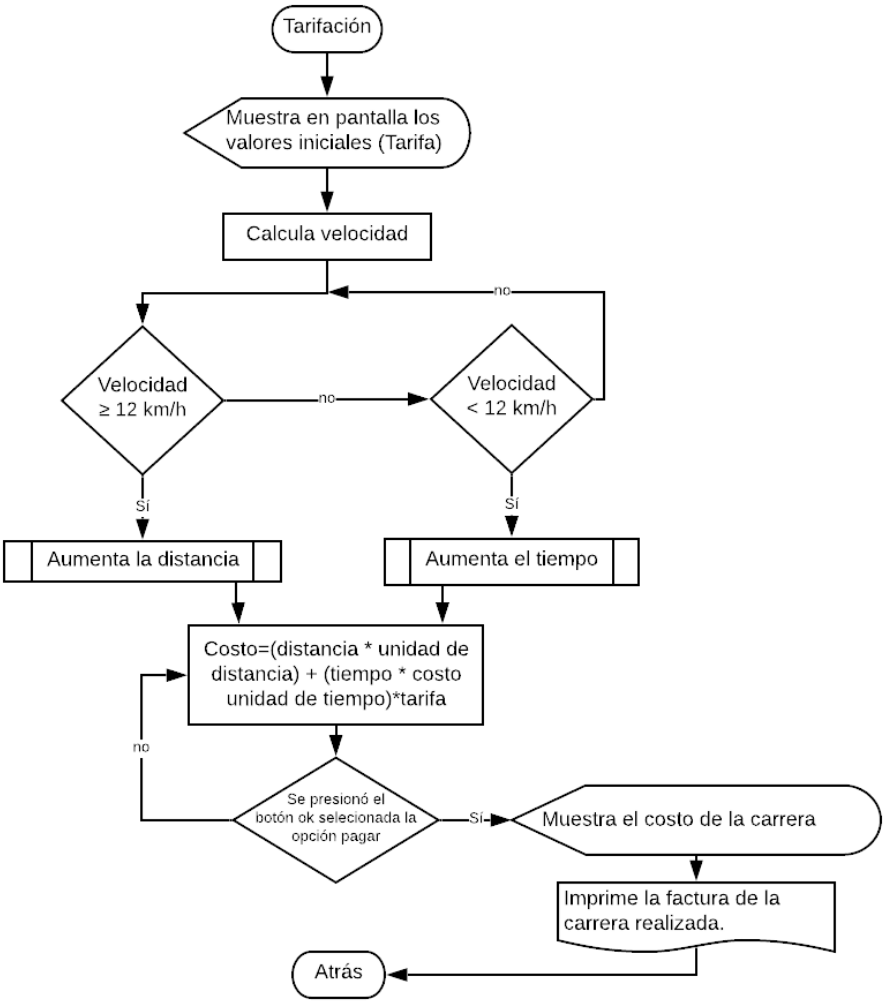


Figura 3.5: Flujograma de subrutina tarifación

Para finalizar la subrutina, en la pantalla se visualiza una opción denominada pagar, el cual termina el proceso de la subrutina mostrando el total de la carrera y dando la orden a la impresora de generar la factura para despúes dirigirse a la pantalla principal.

3.1.3. Estado Impresión

El dispositivo de tarifación también cuenta con una subrutina denominada impresión. Para ingresar en ella previamente se debió haber selecionado esta opción. Esta función se encarga

de leer los datos anteriormente almacenados en la memoria eeprom del microcontrolador y la tarjeta SD. El estado tiene cuatro items de impresión: Parámetros, reporte diario, mensual y total, que se pueden selecionar presionando los botones de navegación. Cada una de ellas accede a la memoria para tomar los datos y generar un ticket. El funcionamiento de la subrutina impresión se observa en la Figura 3.6.

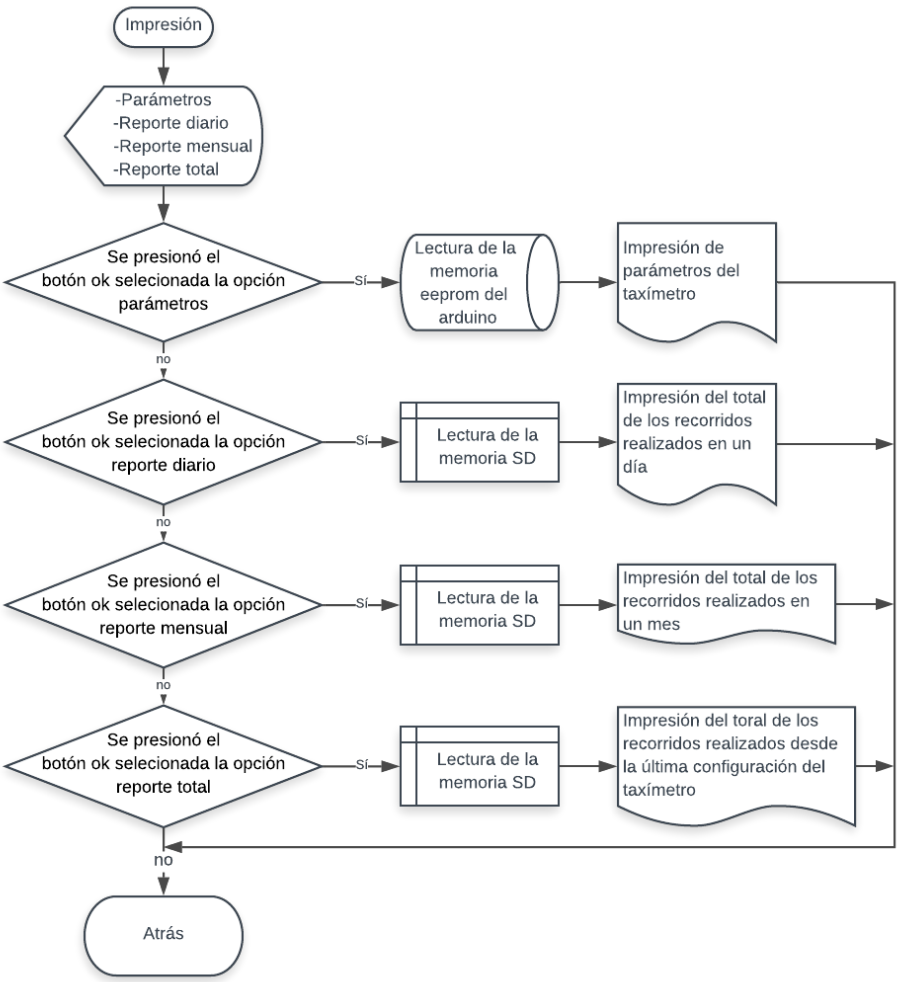


Figura 3.6: Flujograma de subrutina imprimir

En este proceso se cuenta con 2 botones de imprimir y atrás; el primero se visualiza en pantalla y presionando el botón ok da la orden a la impresora de generar un ticket de la op-

ción selecionada para luego regresar a la pantalla principal, mientrás que el segundo solamente regresa a la pantalla principal.

3.2. Características del firmware

El software del dispositivo de tarifación vehicular que se utiliza para el desarrollo, compilación y carga del programa en el arduino es el IDE (Integrated Development Environment), el cual está disponible en diferentes versiones y varios sistemas operativos. El lenguaje de programacin est basado en el lenguaje C.

El programa tiene dos partes fundamentales que son la función setup y la función loop. La setup se ejecuta una sola vez, mientras que la función loop se ejecuta un número infinito de veces. Además para la programación se utilizan comandos, variables para el tipo de datos, operadores aritméticos, operadores lógicos, comparadores, estructuras de control y librerías.

Para el desarrollo del programa que pone en funcionamiento al dispositivo de tarifación se utilizan las librerías que se indican en la Tabla 3.2.

Tabla 3.2: Librerías para los diferentes elementos

Librería	Función
SD.h	Librería del módulo SD
EEPROM.h	Permite el acceso a la EEPROM del Arduino
SPI.h	Librería para la comunicación con el módulo SD
RTClib.h [31]	Permite la comunicación con un módulo RTC
Wire.h	Comunica la placa arduino con dispositivos que trabajan mediante el protocolo I2C/TWI
Separador.h	Permite separar valores
epd2in9.h [32]	Definel el tamaño de la pantalla en pulgadas
epdpaint.h [32]	Librería que permite establecer la orientación, fondo y tamaño de letra de los caracteres de la pantalla
Adafruit Thermal.h [33]	Librería de la impresora térmica
SoftwareSerial.h	Proporciona una puerta software adicional

Capítulo 4

Implementación del Hardware y Análisis de Datos

En este capítulo se describe la implementación del dispositivo de tarifación, abordando la metodología mostrada previamente en el Capítulo 3. Además se presentan los resultados obtenidos a partir de las pruebas realizadas en condiciones reales de funcionamiento.

4.1. Prototipado mediante impresió n 3D

La impresión 3D es una técnica de prototipado rápido la cual es utilizada en distintos campos de investigación apoyado en el uso de datos CAD 3D, en el presente proyecto es utilizado para la impresión de la carcasa del dispositivo de tarifación.

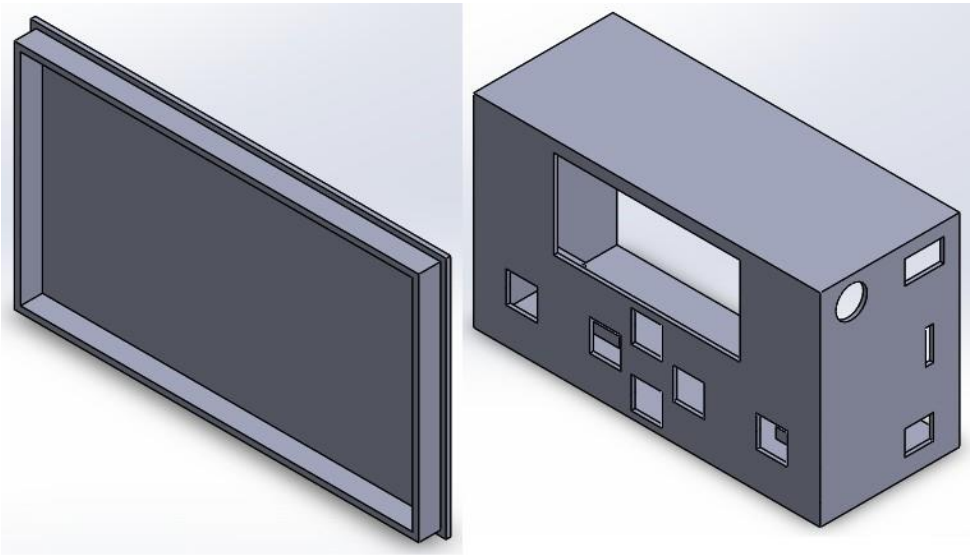


Figura 4.1: Diseño de la carcasa del dispositivo

4.2. Implementación de la tarjeta de control

El hardware está conectado mediante una placa electrónica doble cara particularmente diseñada para esta aplicación, la placa contiene los elementos necesarios para el correcto funcionamiento del dispositivo.

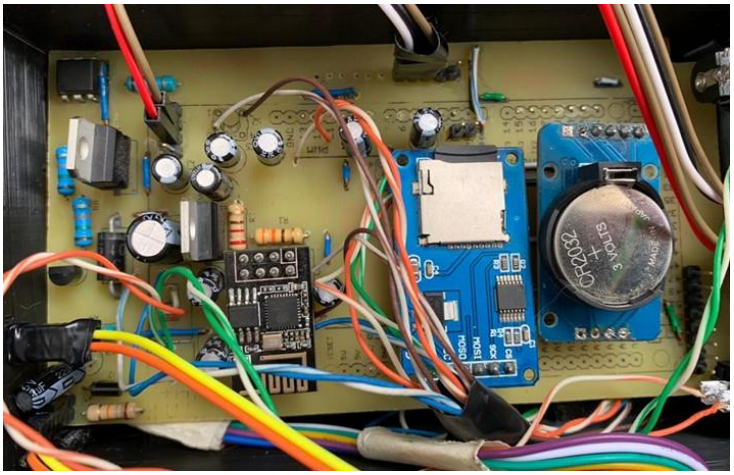


Figura 4.2: Placa electrónica de control

4.3. Pruebas de funcionamiento y validación

Incluye todas las pruebas de funcionamiento realizadas al taxímetro, como son: pruebas de navegación del menú, funcionamiento del reloj, pruebas de velocidad, distancia recorrida, y por último pruebas de costo de la carrera e impresión con detalles de la misma, estas pruebas fueron realizadas en el exterior con el dispositivo instalado en un vehículo. En las pruebas se consideró todas las posibles variables que afectan el funcionamiento y se ha realizado los ajustes necesarios del firmware.

4.3.1. Pruebas de navegación por el menú del sistema

Al iniciar el sistema se mostrará la pantalla principal como se aprecia en la Figura 4.3, donde se observará la hora que estará expresada en el formato HH:MM y los tres estados del taxímetro: Libre, Tarifación e Impresión siendo el primero el que esté selecionado por defecto.

Figura 4.3: Pantalla de inicio del dispositivo

4.3.1.1. Estado Libre

En este estado se observarán cuatro ítems como se detalló en capítulo 3, cada uno de los parámetros contiene información específica. La Figura 4.4 muestra la pantalla del estado libre.

Figura 4.4: Pantalla de estado libre

4.3.1.2. Estado Tarifació n

Para cambiar al estado tarifación desde el estado libre, se dede presionar el botón atrás y luego selecionar la opción tarifación con las teclas de navegación y posteriormente presionar el botón ok.

Figura 4.5: Pantalla de estado tarifación

Cuando ingresamos en este estado del taxímetro automáticamente empieza el cálculo del servicio y muestra en pantalla como valor incial la tarifa, como se observa en la Figura 4.5, para finalizar con este estado debe presionar el botón ok, el cual nos dirigirá a la pantalla principal y dará la orden a la impresora de generar la factura.

4.3.1.3. Estado Impresió n

En el estado de impresión tenemos 4 opciones de impresión de ticket como se observá en la Figura 4.6.

Figura 4.6: Pantalla del estado impresión

Las pruebas de funcionamiento de la impresora se muestran en el desarrollo de este capítulo.

4.3.2. Pruebas del hardware en un vehículo

Se realiz todas las pruebas con el equipo instalado en un automóvil, estas necesariamente fueron realizadas en condiciones de campo, por sus características de cambio de posición y/o cambio de velocidad, pruebas como: las de velocidad del vehículo, la distancia recorrida, la velocidad de cambio o de cruce. A continuación se procede a presentar más detalladamente.

4.3.2.1. Pruebas de velocidad

Para estas pruebas se necesitó la ayuda de un vehículo y su correspondiente velocímetro. Se realizó mediciones de velocidad del vehículo por varios lugares de la ciudad de Cayambe, y se comparó entre la velocidad del velocímetro y la velocidad indicada por el dispositivo obteniendo las diferencias que se muestran en la Tabla 4.1.

Tabla 4.1: Valores registrados en las pruebas de velocidad del dispositivo

t (seg)	Vel. Velocímetro en t (km/h)	Vel. Taxímetro en t (km/h)	Error Velocidad (km/h)	Error Prom (km/h)
1	0	0	0.00	0.07
2	6	5.8	-0.2	0.07
3	21	22	1	0.07
4	41	42.5	1.5	0.07
5	63	61.6	-1,4	0.07
6	59	60	1	0.07
7	61	60	-1	0.07
8	68	66.6	-1.4	0.07
9	59	59.5	0.5	0.07
10	46	47	1	0.07
11	13	12	-1	0.07
12	1	3	2	0.07
13	0	0	0	0.07
14	0	0	0	0.07
15	14	11.9	-2.1	0.07
16	46	46.9	0.9	0.07
17	61	62.4	1,4	0.07
18	76	75.9	-0.1	0.07
19	98	97.2	-0.8	0.07
20	101	101.1	0.1	0.07

En las pruebas de velocidad del vehículo, se puede observar que el error máximo es de 2.10 km/h en todo el transcurso de la prueba. No se puede descartar que existan valores mayores al error máximo presentado en la prueba, sin embargo este valor sería en primera instancia un error despreciable debido a que se daría en un segundo de todo el tiempo de la muestra, y sabiendo que este parámetro de velocidad solo se utiliza para tomar la decisión de facturar por tiempo o por distancia.

Error(v)= Vel. Velocímetro(t) - Vel. taímetro(t) (Ec. 4.1)

4.3.2.2. Pruebas de distancia del recorrido

Para estas pruebas se necesitó de un vehículo de prueba, una pista despejada y del tablero del vehículo donde se puede observar el kilometraje encerado. Para la comparación de la distancia recorrida. Se escogió la panamerica desde el Yaznán hasta la Unidad Educativa Cayambe por

las razones de facilidad de pruebas. En la Tabla 4.2, se muestra los datos tomados del dispositivo en recorrer la carretera mencionada con un rango de velocidad de 30 a 60 km/h, velocidades mayores a la velocidad de cambio, adicionalmente se comprobó su distancia con el odómetro del vehículo.

Tabla 4.2: Datos de distancia recorrida

No.	Datos del dispositivo	Error(km)	% Error	Error Prom(km)	%Error Prom
1	10.01	0.01	0.1	0.016	0.16
2	9.94	0.06	0.6	0.016	0.16
3	9.93	0.07	0.7	0.016	0.16
4	10.02	0.02	0.2	0.016	0.16

Para el cálculo de errores de distancia recorrida se utilizó la Ecuación 4.2, con los valores obtenidos al dar varias vueltas en la carretera de prueba, encerando tanto el odómetro del vehículo como el marcador de distancia recorrida del prototipo con cada inicio de prueba.

Error(d) = distancia odómetro - distancia del dispositivo (Ec. 4.2)

Se realizó las mismas pruebas de distancia recorrida dentro de la ciudad de Cayambe, obteniendo los datos presentados en la Tabla 4.3, la cual también detalla los errores del prototipo en relación a las lecturas obtenidas por el odómetro del vehículo considerando una distancia de 2.2 km.

Tabla 4.3: Comparación de datos del odómetro con la distancia recorrida del prototipo

No.	Datos del dispositivo	Error(km)	% Error	Error Prom(km)	%Error Prom
1	2,1	0,1	4,55	0,164	7,45
2	2,14	0,06	2,73	0,164	7,45
3	1,92	0,28	12,73	0,164	7,45
4	2,05	0,15	6,82	0,164	7,45
5	2,02	0,18	8,18	0,164	7,45
6	1,94	0,26	11,82	0,164	7,45
7	2,15	0,05	2,27	0,164	7,45
8	2,03	0,17	7,73	0,164	7,45
9	2,01	0,19	8,64	0,164	7,45
10	2	0,2	9,09	0,164	7,45

En la Tabla 4.3, se tiene un mayor grado de error, esto debido a que la distancia recorrida indicada por el dispositivo solo es la que recorre cuando supera la velocidad de cruce, valor de distancia necesario para el cálculo del costo del servicio. Es decir si el vehículo viaja a una velocidad menor a la velocidad de cruce 12km/h, el odómetro incrementará su distancia mientras que la distancia del dispositivo no incrementará su medida, estos valores de error se observarán con mayor porcentaje cuando el vehículo realice mayor número de paradas.

Como no fue el caso de las pruebas en la panamerica de la ciudad de Cayambe donde no se tuvo ninguna parada y el error fue menor al 1%.

4.3.2.3. Pruebas de tiempo de espera

Estas pruebas se las realizó dentro de un ambiente cerrado. Se escogió la opción tarifación en el dispositivo, y se la dejó por variar horas. Llegando al límite de la memoria, observando que se produjo una falla de saturación debido al espacio de memoria de la variable asociada a este parámetro, se amplió el espacio en memoria para esta variable a un formato entero de mayor capacidad, solventando el inconveniente tomando en cuenta que el tiempo de espera lógico por carrera no superará las 9 horas 6 minutos 7 segundos, que transcurre hasta saturar el espacio de memoria.

4.3.2.4. Pruebas de velocidad de cambio

Esta parte indica pruebas con el dispositivo montado en el vehículo, la primera prueba que se realizó es la prueba de velocidad de quiebre. Esta prueba implica que en primera instancia se comparó la velocidad que marca el taxímetro con la que marca en el velocímetro del vehículo. A continuación, en la segunda columna de la tabla 4.4 se presentan los resultados obtenidos en la Tabla 4.1.

Tabla 4.4: Valores de tiempo de espera y distancia recorrida en función de la velocidad

t (seg)	Vel Taxímetro en t (km/h)	Incr T. espera (seg)	T. espera (seg)	Incr Dis. recorr (km)	Dis recorr (km)
1	0	1	1	0.000	0.000
2	5,8	1	2	0.000	0.000
3	22	0	2	0.006	0.006
4	42.5	0	2	0.012	0.017
5	61.6	0	2	0.017	0.034
6	60	0	2	0.016	0.051
7	60	0	2	0.016	0.067
8	66.6	0	2	0.018	0.085
9	59.5	0	2	0.016	0.101
10	47	0	2	0.013	0.114
11	12	1	3	0.000	0.114
12	3	1	4	0.000	0.114
13	0	1	5	0.000	0.114
14	0	1	6	0.000	0.114
15	11.9	1	7	0.000	0.114
16	46.9	0	7	0.013	0.127
17	62.4	0	7	0.017	0.144
18	75.9	0	7	0.021	0.165
19	97.2	0	7	0.027	0.191
20	101.1	0	7	0.032	0.223

Teniendo valores aceptables de las mediciones de velocidad en el prototipo, se puede verificar la velocidad de cruce, esta es de 12 km/h que es la velocidad límite para que el prototipo facture por tiempo de espera y facture por distancia recorrida. Como se puede observar en la Tabla 4.3.

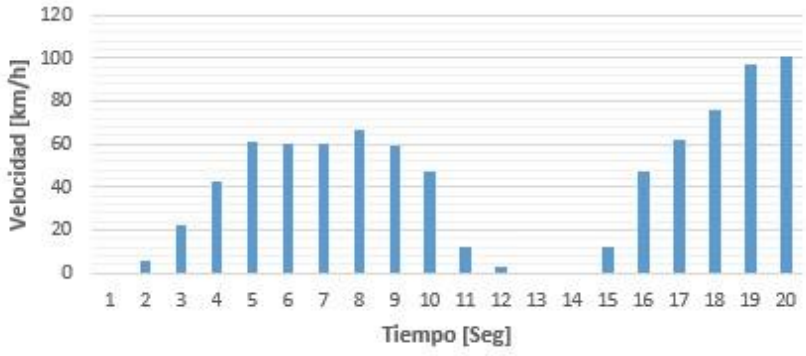


Figura 4.7: Gráfica de las variaciones de la velocidad del automóvil en función del tiempo

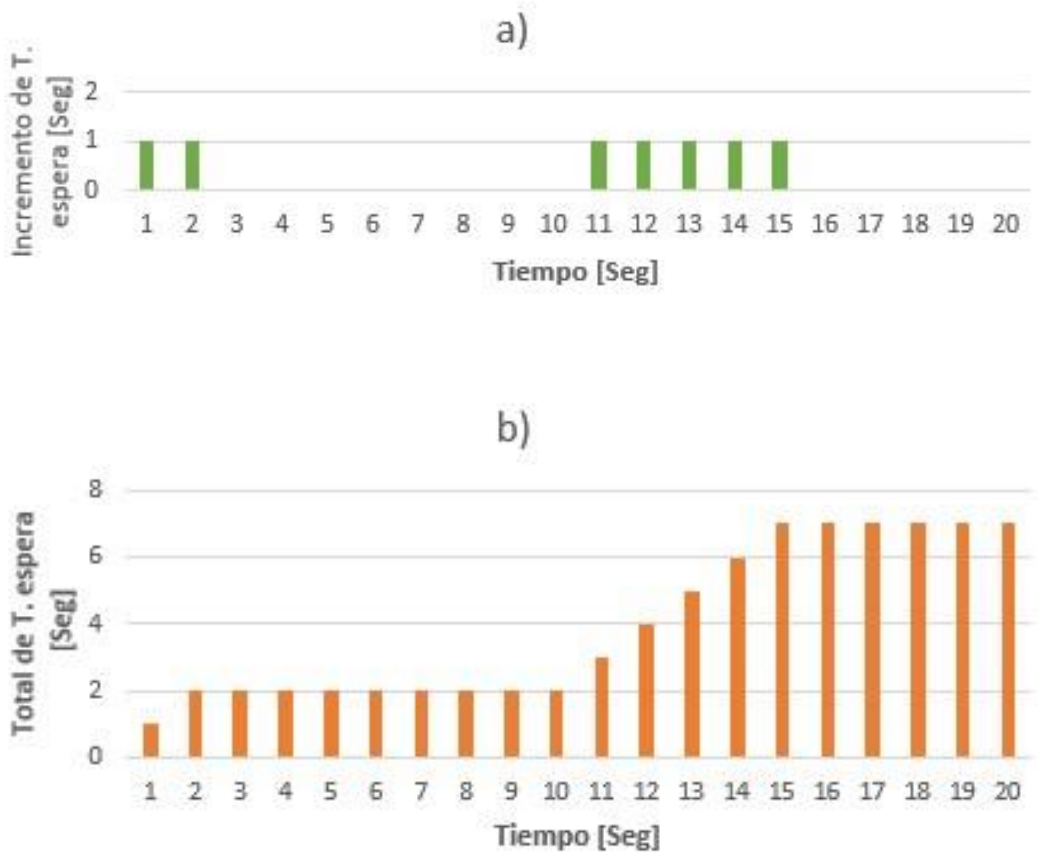
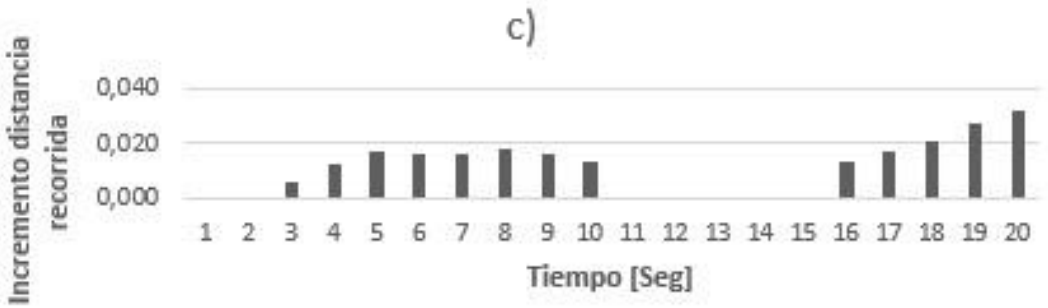


Figura 4.8: Gráficas a) y b) donde detalla el incremento del tiempo de espera en función de la velocidad de quiebre 12 km/h



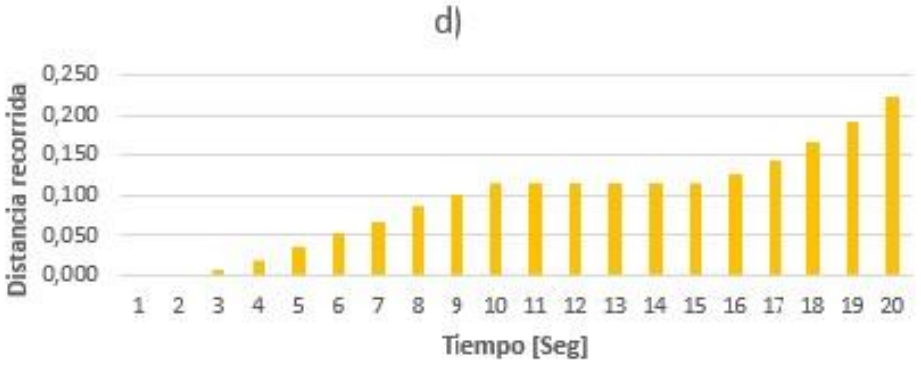


Figura 4.9: Gráficas c) y d) de distancia recorrida incrementan cuando supera la velocidad de cruce de 12 km/h

4.3.2.5. Costo de la carrera con un taxímetro homologado y con el dispositivo

Para esta prueba se necesitó la ayuda de un taxi y su respectivo taxímetro homologado, equipo que superó correctamente las pruebas de control. En la Figura 4.10 se puede observar una captura de la herramienta Google Earth, detallando la ruta de prueba, de un punto A hacia un punto B. El recorrido tiene una distancia de 7,26 km, donde se pudo alcanzar una velocidad mayor a 80 Km/h en una recta sin tráfico, consta de 3 paradas por semáforos en rojo y una curva cerrada.

Figura 4.10: Recorrido de prueba para el taxímetro y el prototipo

En la Tabla 4.5, se presenta un resumen de datos del costo de la carrera obtenida entre la comparación de un taxímetro homologado y el prototipo, la muestra se tomó minuto a minuto de una recopilación de información mayor que se tomó segundo a segundo en la ruta determinada de prueba, que duró 12 minutos en recorrer 7,26 km.

Tabla 4.5: Comparación de taxímetro homologado con el prototipo

Tiempo [Min]	Taxímetro	Prototipo	Error
12:46:25	$1,25	$1,25	$0.00
12:47:25	$1,25	$1,25	$0.00
12:48:25	$1,32	$1,32	$0.00
12:49:25	$1,70	$1,71	$0.01
12:50:25	$1,96	$1,97	$0.01
12:51:25	$2,11	$2,12	$0.01
12:52:25	$2,38	$2,39	$0.01
12:53:25	$2,70	$2,71	$0.01
12:54:25	$3,02	$3,04	$0.02
12:55:25	$3,25	$3,27	$0.02
12:56:25	$3,48	$3,50	$0.02
12:56:25	$3,69	$3,71	$0.02

Se tuvo que realizar pequeños cambios en los impulsos por kilómetro del dispositivo ya que estos varían de acuerdo al modelo del vehículo y así no tener un error mayor a $0.02 en el costo total de la carrera.

4.3.3. Pruebas con la impresora

4.3.3.1. Pruebas de comunicación con la impresora

Para estas pruebas se tuvo cuidado de tener las mismas velocidades de comunicación entre el taxímetro y la impresora, en este caso velocidad de trasmisión serial es de 19200 bps.

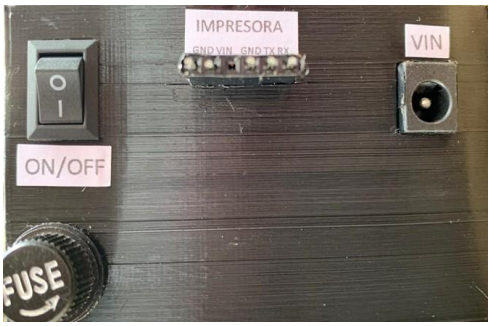


Figura 4.11: Terminales de conexión para la impresora

4.3.3.2. Pruebas de impresió n

Se realizó las pruebas de impresión con éxito, se conectó los dos conectores de tres pines de la impresora al dispositivo, luego de realizar un recorrido se imprimió el ticket o facura de la carrera realizada, observando que la impresora toma unos 5 segundos en imprimir el documento que está dirigido al usuario con los detalles completos de la carrera. En la Figura 4.12 se muestra la fotografía de la impresión del detalle de la carrera emitida por la impresora térmica con la que se realizó las pruebas de impresión.


TICKET DE CONTROL

IBARRA

Titular: David Guzman

PLACA: AAA-000

Tarifa utilizada: 1
INICIO DEL VIAJE: 6:29
FIN: 2020/1/6 6:29
Km recorrios: 1.06
Valor de tarifa: 1.25
Valor por KM: 0.34
Valor por minuto de espera 0.00

A PAGAR

$ 1.60

GRACIAS POR ELEGIRNOS

Figura 4.12: Impresión de ticket con detalle de la carrera

Tomando en cuenta que el dispositivo permité también imprimir otro tipo tickets como se explico en el Capítulo 3. Se realiza las impresiones teniendo como resultado las Figuras 4.13 y 4.14.

Figura 4.13: Impresión de tickets de parámetros y reporte diario

Figura 4.14: Impresión de tickets de reporte mensual y reporte total

4.3.4. Costos de fabricación

Para el diseño y la implementación del proyecto se utilizó varios elementos y dispositivos electrónicos, con dicho costo se puede realizar una comparación con dispositivos que realizan la misma tarea existente en el país. En la Tabla 4.6 se detalla el costo de fabricación del dispositivo.

Tabla 4.6: Costo de fabricación del dispositivo

Cantidad	Elemento	Valor
1	Arduino Mega	$20.00
1	Pantalla	$21.00
1	Reloj en tiempo real	$2.80
1	Módulo micro SD	$2.50
1	Módulo Wi-Fi	$5.00
1	Impresora térmica	$80.00
1	Componentes electrónicos	$16.00
1	Carcasa	$12.00
1	Otros	$10.00
1	Investigación	$40
Total		$209.3

Se observa que el campo de la investigación es de 40.00$ tomando en cuenta que no será el único dispositivo que se realizará, por otro lado el valor más alto obtenido de los distintos componentes del prototipo es la impresora térmica ya que es muy importante para la impresión de los recibos.

Conclusiones y Recomendaciones

Conclusiones

En este trabajo se desarrolló una plataforma de hardware libre la cual permite tomar las señales del sensor Vss de un vehículo, calculár el costo de una carrera e imprimir un recibo con los valores correspondientes.

Se determinó la funcionalidad mediante los manuales de usuario de seis taxímetros existentes en el mercado como: Tango XP, D10, F3 Plus, TX-10, RS-01 y TX40. Luego de leer sus principales caracteríticas se pudo concluir que los taxímetros mencionados anteriormente utilizan un panel visual conformado por displays y su costo es elevado. Por tal motivo el sistema desarrollado tiene una pantalla monocromática y un bajo costo de adquisición.

El hardware del taxímetro se diseñó con un Arduino Mega 2560 que es el encargado de ejecutar el algoritmo de control que permite el funionamiento del taxímetro, además en esta placa se conectó un reloj en tiempo real, un módulo micro SD para el almacenamiento de informacíon, una pantalla monocromática, un módulo Wi-Fi ESP8266, una botonera de pulsadores para la navegación. Todos estos elementos se encuentran ubicados de manera ordenada y adecuada dentro de una carcasa impresa en 3D.

La programación del microcontrolador se lo hizó a través del IDE de arduino el cual es una plataforma de uso sencillo y bastante completa para realizar este tipo de aplicaciones. Se usó

librerías para cada uno de los componentes que forman parte del hardware.

Las tablas realizadas en el capítulo 4, demuestran que el taxímetro diseñado es un dispositivo confiable con un buen desempeño y que cumple que la funcionalidad requerida.

Recomendaciones

- Se debe realizar una correcta calibración de los impulsos por kilómetro para que el calculo de la velocidad, distancia y tiempo sean lo más acertadas posibles.
- Verificar que la hora del dispositivo sea la correcta y sincronizada con la hora reloj del ordenador, si no lo está se debe actualizarla haciendo uso del software de configuración.
- Para conectar la impresora, identificar y enchufar correctamente los pines, caso contrario la impresora se quermará.

Trabajo futuro

Se debe utilizar componentes electrónicos SMD para optizar el espacio y hacer del dispositivo más pequeño.

Se puede implementar un módulo GPS el cual permitirá determinar la posición y también tener una mayor sicronización de la hora y fecha del dispositivo.

Bibliografía

[1] Resolución N°020-DIR-2013-ANT. Reglamento de aplicación para la homologación, instalación y uso del taxímetro en el servicio de transporte comercial en taxis convencionales y ejecutivos. Quito, Ecuador, 2013.

[2] M. Cuasapaz, "Construcción e implementación de un prototipo de un taxímetro digital con impresora facturadora utilizando tecnología GPS y el desarrollo de un firmware en un microcontrolador", Tesis Pregrado, Escuela Politécnica Nacional, Quito, Ecuador, Abril.2013.

[3] NTE INEN 2663:2013, Taxímetros. Requisitos metrológicos y técnicos, procedimientos de ensayo. Quito, Ecuador, 2013.

[4] B. Wu, and A. An, "FPGA Design and Implementation of Taximeter Anti -Fraud System", in IEEE Youth Academic Annual Conference of Chinese Association of Automation., pp. 654659, May.2018.

[5] C. Morales, "Diseño, construcción e implementación de un taxímetro con almacenamiento de viajes mediante GPS en el vehículo Chevrolet corsa wind 1.4", Tesis Pregrado, Universidad Técnica del Norte, Ibarra, Ecuador, Marzo.2016.

[6] L. Redrobán, "Diseño e implementaión de un prototipo para el control y tarifación del servicio de taxi a través de sensores inerciales", Tesis Pregrado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2018.

[7] Agencia Nacional de Tránsito (Diciembre, 2018). [Online]. Available: https://www.ant.gob.ec/index.php/descargable/file/6007-direcciones-de-empresas-distribuidoras-autorizadas-por-la-ant-al-19-12-2018

[8] Ful-Mar S.A.(2008-2019).[Online]. Available: http://www.ful-mar.com.ar/es/producto.php?producto=2

[9] Sumitrag (2013).[Online]. Available: https://www.ant.gob.ec/index.php/descargables/file/413-intelligent-taximeter-d10

[10] Digitax Automotive Electronics (2019). [Online]. Available: http://www.digitax.com/products/Taximetro-F3-Plus.html

[11] Optronic (2014).[Online]. Available: http://optroniconline.com/category/taximetros/

[12] Automatism and Advanced Techniques, (2018).[Online]. Available: https://www.ata-electronics.com/es/taximetros/taximetro-primus-rs-01/

[13] Taxitronic (2019).[Online]. Available: https://www.taxitronic.com/taximetros/tx40/

[14] E. García “Diseño y construcción de un dispositivo electrónico calculador de distancia y tarifas para taxis implementado en un banco de pruebas para la Escuela de Ingeniería Automotriz”, Tesis Pregrado, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, 2013.

[15] Motor Canales Mapfre (2019).[Online]. Available: https://www.motor.mapfre.es/consejos-practicos/consejos-de-conduccion/como-funciona-taximetro/

[16] Agencia Nacional de Tránsito, (2015). [Online]. Available: https://hoyenimbabura.com/2015/07/30/tarifataxisop/

[17] I. Sommerville “Ingeniería del Software”. 9na.ed. México: Pearson Educación. S.A., 2011, pp. 85-39.

[18] Requerimientos Funcionales y No Funcionales, (2018). [Online]. Available: https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a

[19] K. Ogata “Ingeniería de control moderna”. 5ta Madrid: Pearson Educación, S.A., 2010 ed. cap 3.

[20] G. Gridling, B. Weiss, “Introduction To Microcontrollers”, Vienna University of Technology, Austria 2007.

[21] Sensor de velocidad, (2019). [Online]. Available: https://www.autoavance.co/blog-tecnico-automotriz/139-fallas-en-sensor-de-velocidad-sensores-de-velocidad-efecto-hall/

[22] Transductor, (2014). [Online]. Available: https://es.wikipedia.org/wiki/Transductor

[23] Optoacoplador 4N25. [Online]. Available: https://tiendasistema16.wixsite.com/sistema16/product-page/optoacoplador-4n25

[24] Módulo de papel electrónico de 2.9 pulgadas, (2019). [Online]. Available: https://www.waveshare.com/wiki/2.9inch_e-Paper_Module

[25] Reloj y Calendario en arduino con los RTC ds1307 y ds3231, (2016). [Online]. Available: https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/

[26] International Standard ISO/IEC/IEEE 18880, Information technology Ubiquitous green community control network protocol. New York, USA 2015.

[27] ESP8266 todo lo que necesitas saber del mdulo WiFi para Arduino. Programador Fácil, (2017). [Online]. Available: https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/

[28] L Thayer . “Manual de usuario Impresora térmica con auto corte junto a arduino”. MCI Ingeniería Electronics. Providencia, Santiago, chile. 2016

[29] Leer y esribir en una tarjeta SD o micro SD con arduino, (2016). [Online]. Available: https://www.luisllamas.es/tarjeta-micro-sd-arduino/

[30] A. Sanchis, A. Ledezma. “Autámatas Finitos”. Universidad Carlos III de Madrid. Grado Ingeniería Informática. Madrid, España 2009.

[31] Librería del RTC para arduino. [Online]. Available: https://www.arduinolibraries.info/libraries/rt-clib

[32] Librería de la pantalla E-paper para arduino. [Online]. Available: https://github.com/waveshare/e-Paper

[33] Librería de la impresora térmica. [Online]. Available: https://github.com/adafruit/Adafruit-Thermal-Printer-Library

Apéndice

Este apéndice incluye el codigo del taxímetro desarrollado que incluyen las funciones que realiza la Plataforma Abierta para Desarrollo de Dispositivos de Tarifación vehicular: Sistema Embebido

.A. Software

.A.1. Código Arduino (taximetro.ino)

Programa 1: Código fuente del firmware desarrollado para el dispositivo

```
#include <SD.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
#include <SPI.h>
#include <SD.h>
#include "RTClib.h"
#include <Wire.h>
#include <Separador.h>
#include <epd2in9.h>
#include <epdpaint.h>
#define DEBUG true
#include "imagedata.h"

#include "Adafruit_Thermal.h"
#include "SoftwareSerial.h"

#define TX_PIN 6 // Arduino transmit  YELLOW WIRE  labeled RX on printer
#define RX_PIN 5 // Arduino receive   GREEN WIRE   labeled TX on printer

SoftwareSerial mySerial(RX_PIN, TX_PIN); // Declare SoftwareSerial obj first
Adafruit_Thermal printer(&mySerial);     // Pass addr to printer constructor
```

```
File myFile;
Separador s;
RTC DS3231 rtc;

#define COLORED      0
#define UNCOLORED    1
volatile int ISRenter = 0;
volatile int ISRatras = 0;
volatile int ISRabajo = 0;
volatile int ISRarriba = 0;
const int  ENTER = 8;//19     // the pin that the pushbutton is attached to
const int  ATRAS = 9;//3
const int  PULSOS = 2;
const int  ARRIBA = 7;//19
const int  DERECHA = 19;//7
const int  IZQUIERDA = 18;
const int ABAJO = 3;//9

long timeCounter = 0;
const int timeThreshold = 250;
int contador=0;
boolean Flaginicio = false;
boolean Flaginicio2 = false;
boolean Flaginicio3 = false;
boolean Flagtar = false;
boolean Flagtar1 = false;
boolean Flag = false;
boolean Flagparameters = false;
boolean Flagviewtar1 = false;
boolean Flagviewtar2 = false;
boolean Flagpantalla3 = false;
boolean Flagpantalla2 = false;
boolean Flagpantalla1 = false;
```

```
boolean Flagpantalla = false;
boolean Flaginformacion = false;
boolean Flagtarifa=false;
boolean Flagocupado=false;
boolean Flagpagar=false;
boolean Flagextras=false;
boolean Flagextras1=false;
boolean Flagconfiguracion= false;
boolean Flagoperacion=false;
boolean Flagg=false;
boolean estado= false;
boolean flag=false;
boolean toggle1 = 0;
boolean valoresiniciales=false;
boolean op_pa=false;

float contadortiempo=0.00;
int velocidad=0;
float kilometros=0.00;
float kilometroslibre=0.00;
float kilometrosocupado=0.00;
float contador1=0.00;
float total=0.00;
float total1=0.00;
float total2=0.00;
float total3=0.00;
float adicional =0.00;
float pulsos=0.00;
float pulsos1=0.00;
float pulsos2=0.00;


int enter = 0;    // enter for the number of button presses
int atras = 0;
```

```
int abajo = 0;
int derecha = 1;
int izquierda = 0;
int arriba = 0;
int buttonState = 0;          // current state of the button
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int buttonState5 = 0;
int buttonState6 = 0;
int buttonState7 = 0;
int lastButtonState = 0;      // previous state of the button
int lastButtonState1 = 0;
int lastButtonState2 = 0;
int lastButtonState3 = 0;
int lastButtonState4 = 0;
int lastButtonState5 = 0;
int lastButtonState6 = 0;
int lastButtonState7 = 0;
int Decena1, Unidad1;
int Decena2, Unidad2;
int Decena3, Unidad3;
int configuracion=0;
int n_tarifa=0;
unsigned char image[0];
Paint paint(image,0,0);     // width should be the multiple of 8
Epd epd;

char cc[20];
const char parametros[] PROGMEM = "Parametros";
const char tari[] PROGMEM = "Tarifas";
const char info[] PROGMEM = "Informacion";
const char impr[] PROGMEM = "Modo Configuracion";
```

```
const char tard[] PROGMEM = "Tarifa Diurna";
const char tarn[] PROGMEM = "Tarifa Nocturna";

const char tarm[] PROGMEM = "Tarifa minima:";
const char arranque[] PROGMEM = "Arranque:";
const char porkim[] PROGMEM = "Por Km:";
const char minu[] PROGMEM = "Min espera:";

const char nums[] PROGMEM = "Num Serie:";
const char constk[] PROGMEM = "Constante K:";
const char pla[] PROGMEM = "Placa:";
const char numre[] PROGMEM = "Num Registro:";

const char comp[] PROGMEM = "Compania:";
const char ciudad[] PROGMEM = "Ciudad:";
const char rucs[] PROGMEM = "Ruc:";
const char auts[] PROGMEM = "Aut.SRI:";
const char tit[] PROGMEM = "Titular:";

const char print1[] PROGMEM = "Parametros";
const char print2[] PROGMEM = "Reporte mensual";
const char print3[] PROGMEM = "Reporte total";
const char print4[] PROGMEM = "Reporte diario";

const char * const string_table[] PROGMEM =      // change "string_table" name to suit
{
  parametros, tari, info, impr, tard, tarn, tarm, arranque, porkim, minu,
  nums, constk, pla, numre,
  comp, ciudad, rucs, auts, tit,
  print1, print2, print3, print4
};

int eeAddress = 1; // Compa ia char [50]
int eeAddress1 = 51; //Ciudad char [30]
```

```
int eeAddress2 = 81; //Ruc char [13]
int eeAddress3 = 95; //aut. char [10]
int eeAddress4 = 106; //Propietario char [45]

int eeAddress5 = 150; // Numero de serie [15]
int eeAddress6 = 165; //Constante K float 4 bytes
int eeAddress7 = 170; //Velocidad Critica int 2 bytesi
nt eeAddress8 = 172; //Velocidad M xima int 2 bytesi
nt eeAddress9 = 174; //Placa char[10]
int eeAddress10 = 184; // Numero de registro char [10]

int eeAddress11=200; // Tarifa Diurna
int eeAddress12=205; //Arranque
int eeAddress13=210; //Valor por km
int eeAddress14=215;  //Minuto de espera
int eeAddress15=220;  //Extras

int eeAddress16=225; // Tarifa Nocturna
int eeAddress17=230; //Arranque
int eeAddress18=235; //Valor por km
int eeAddress19=240;  //Minuto de espera
int eeAddress20=245;  //Extras

int eeAddress21 = 250; //Llave Master [4]
int eeAddress22 = 255; //Chofer 1 [20]
int eeAddress23 = 275; //Contrase a 1 [4]
int eeAddress24 = 280; //Chofer 2 [20]
int eeAddress25 = 300; //Contrase a 2 [4]

int eeAddress26 = 310; // Dia de proxima revision int [2]
int eeAddress27 = 314; // Mes de proxima revision int [2]
int eeAddress28 = 318; // A o de proxima revision int [2]
int eeAddress29 = 322; // Fecha de ultima revision int [12]
```

```
int eeAddress30 = 340; // contador de eventos int [2]
int eeAddress31 = 345; // Persona de verificacion char [20]
int eeAddress32 = 400; // Persona de verificacion char [20]
int eeAddress33 = 455; // contador de eventos compara int[2]

int eeAddress34 = 450; // variable para distancia de viaje

int eeAddress50 = 500; // contador de eventos compara int[2]
int eeAddress51 = 505;

char tarifas [5];
int vlc;// vlm;
int aa, mes, dia, hora, minuto, segundo,eventos,eventos1;
int dr;//mr,ar;
int connectionId;
char compania[50];//char propietario[45];
char ciu[30];
char ruc[15];//char aut[11];//char numserie[15];char fecha[12];
char placa[10];//char numreg[10];
char master[5]; //char contra1[5]; char contra2[5];
char chofer1[20];//char chofer2[20];
char compania1[50]="";
unsigned long tiempo1=0;
unsigned long tiempo2=0;
unsigned long tiempoSegundos=0;

char B[5];
char C[5];
String cad="";
String cad1="";
int weekDay;
int h, m, a, b;
int hora_inicio;
int min_inicio;
```

```
int year_fin;
int mes_fin;
int dia_fin;
int hora_fin;
int min_fin;
int UltimaPocicion=0;
int yy,mm,dd,yl,ml,dl;
float tf,f,t,total_diario;
int totalBytes=0;
String cadena="";

int index=0;
float temp, temp1, temp2, arra, porkm, mines, ext,arra1, porkm1, mines1, ext1,td;
float arra2, porkm2, mines2, ext2, val1, val2, val3, val4;

void setup()
{
  noInterrupts();
  TCCR1A = 0;
  TCCR1B = 0;
  TCCR1B |= (1 << WGM12);
  TCCR1B |= (1 << CS12) | (1 << CS10);
  TCNT1 = 0;
  OCR1A = 46874;// = (16*10^6) / (1*1024) - 1 (must be <65536) 3s=47347 1s=15624 46874
  TIMSK1 |= (1 << OCIE1A);
  interrupts();
  pinMode(ENTER, INPUT_PULLUP);
  pinMode(ATRAS, INPUT_PULLUP);
  pinMode(ABAJO, INPUT_PULLUP);
  pinMode(DERECHA, INPUT_PULLUP);
  pinMode(IZQUIERDA, INPUT_PULLUP);
  pinMode(ARRIBA, INPUT_PULLUP);
  pinMode(PULSOS, INPUT_PULLUP);
```

```
attachInterrupt(digitalPinToInterrupt(PULSOS), interruptCount, LOW);

Serial.begin(115200);     //////For Serial monitor
Serial2.begin(115200);
Serial1.begin(115200);

Serial1.print("Iniciando SD ...");
if (!SD.begin(4))
{
 Serial1.println("No se pudo inicializar");
 return;
}
Serial1.println("inicializacion exitosa");

if (! rtc.begin())
{
  while (1);
}
if (rtc.lostPower())
 {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
 }

if (epd.Init(lut_full_update) != 0)
{
  return;
}
epd.ClearFrameMemory(0xFF);   // bit set = white, bit reset = black
epd.DisplayFrame();
epd.ClearFrameMemory(0xFF);   // bit set = white, bit reset = black
epd.DisplayFrame();

if (epd.Init(lut_partial_update) != 0)
```

```
  {
    return;
  }
  tiempo1=millis();
  EEPROM.get(eeAddress4,compania);
  //escritura();
  mySerial.begin(19200);
  printer.begin();
}
ISR(TIMER1_COMPA_vect)
{
 if (toggle1)
 {
    toggle1 = 0;
    flag=true;
  }
  else
  {
    toggle1 = 1;
    flag=true;
  }
}
void interruptCount()
{
  pulsos++; //velocidad
  pulsos1++; //kmocupado
  kilometros=pulsos1/2500;

  Serial.print("Pulsos1: ");
  Serial.println(pulsos1);
}
void(*resetFunc)(void)=0;
void loop()
{
```

```
  ENter();
  ATras();
  left();
  right();
  ABajo();
  ARriba();
//   Serial.print("Enter: ");
//   Serial.println(ISRenter);
//   Serial.print("Abajo: ");
//   Serial.println(ISRabajo);
//   Serial.print("Arriba: ");
//   Serial.println(ISRarriba);
//   Serial.print("Derecha: ");
//   Serial.println(derecha);
//   Serial.print("Izquierda: ");
//   Serial.println(izquierda);
//   Serial.print("Atras: ");
//   Serial.println(ISRatras);
if(estado==true)
      {
        if (flag==true)
            {
                velocidad =(pulsos*3600)/(3*2500); //2400
                flag=false;
                pulsos=0;
            }
         if(kilometros<=2)
           {
            total3=total;
           }
       else if (kilometros>2)
       {
          if (velocidad>12)
          {
```

```
            adicional=((kilometros-2)*val1);
            total1=adicional;
          }
          if (velocidad<12)
          {
            tiempo();
            total2=contadortiempo;
          }
          total3=(total+total1+total2);
        }
      }
//        Serial.print(contadortiempo);
            Serial.print("kilometros: ");
            Serial.println(kilometros);
            Serial.print("velocidad: ");
            Serial.println(velocidad);
//              Serial.print("inicial: ");
            //Serial.println(total); //valor por tarifa 1.25 o 1.50
            //Serial.print("por km: ");
            //Serial.println(total1); //valor por km
            //Serial.print("min espera: ");
            //Serial.println(total2); //valor por minuto de espera
            //Serial.print("Total: ");
           // Serial.println(total3); //suma de todo
//              Serial.print("pulsos: ");
//              Serial.println(pulsos);
if (Flag == true)
        {
          if (epd.Init(lut_partial_update) != 0)
          {
          return;
          }
          epd.ClearFrameMemory(0xFF);   // bit set = white, bit reset = black
          epd.SetFrameMemory(IMAGE_DATA);
```

```
        Flag = false;
    }

if (Flagpantalla == true)
    {
      if (ISRabajo <= -1 || ISRabajo >= 4)
      {
        ISRabajo = 0;
      }
        ISRarriba = 0;
        izquierda = 0;
        derecha=1;
    }
  if (Flagparameters==true)
  {
        ISRabajo = 0;
  }
  if (Flagpantalla1==true)
  {
        ISRarriba = 0;
        izquierda = 1;
        derecha=1;
  }
  if (Flaginformacion==true)
  {
        ISRarriba = 0;
        ISRabajo=2;
  }
  if (Flagtar==true)
  {
    izquierda = 1;
    derecha = 1;
       if (ISRabajo == -1 || ISRabajo == 0)
          {
```

```
            ISRabajo = 1;
          }
      if (ISRabajo == 3)
          {
            ISRabajo = 1;
          }
  }
if (Flagocupado == true)
{
   if (izquierda >= 0)
  {
    izquierda = 1;
    derecha=2;
  }
  if(derecha>=3)
  {
    derecha=2;
  }
  if(ISRatras>=1)
  {
    ISRatras=0;
    ISRenter=1;
  }
  ISRabajo=0;
  ISRarriba=1;
}
if (Flagviewtar1 == true)
{
   ISRabajo = 1;
   ISRarriba = 0;
}
if (Flagviewtar2 == true)
{
   ISRabajo = 2;
```

```
        ISRarriba = 0;
    }
if (Flagpagar == true)
    {
       if (izquierda >= 0)
      {
        izquierda = 0;
      }
      if (ISRenter >= 0)
      {
        ISRenter = 0;
      }
      if (derecha >= 0)
      {
        derecha = 1;
      }
    }
 if (op_pa==true)
      {
        izquierda = 0;
        derecha = 3;
           if (ISRabajo == -1)
              {
                ISRabajo = 0;
              }
          if (ISRabajo == 4)
              {
                ISRabajo = 0;
              }
          if (ISRarriba >= 1)
              {
                ISRarriba = 0;
              }
          if (izquierda >= 1)
```

```
            {
              izquierda = 0;
              derecha=3;
            }
     }
// ////////////////////////////////////////////////
 if (ISRenter == 0 && derecha == 1 || ISRenter == 0 && derecha == 4)
 {
   epd.SetFrameMemory(IMAGE_DATA);
   inicio1();
 }
 if (ISRenter == 0 && derecha == 2)
 {
   inicio2();
 }
 if (ISRenter == 0 && derecha == 3)
 {
   inicio3();
 }

 if (ISRenter == 1 && derecha == 1 && ISRabajo == 0|| ISRenter == 1 && derecha == 1 &
 {
   pantalla();
 }
 if (ISRenter == 1 && derecha == 1 && ISRabajo == 1)
 {
   pantalla1();
 }
 ////// pantalla2 //////
 if (ISRenter == 1 && derecha == 1 && ISRabajo == 2)
 {
   pantalla2();
 }
 ////// pantalla3 //////
```

```
if (ISRenter == 1 && derecha == 1 && ISRabajo == 3)
{
  pantalla3();
}
if (ISRenter == 2 && derecha == 1 && ISRabajo == 0 && izquierda==0)
{
  parameters();

}
if (ISRenter == 2 && derecha == 1 && ISRabajo == 1 && izquierda==1|| ISRenter == 2 &
{
  tariffs();
}
if (ISRenter == 2 && derecha == 1 && ISRabajo == 2 && izquierda==1)
{
  tariffs1();
}
if (ISRenter == 3 && derecha == 1 && ISRabajo == 1&& izquierda==1)
{
  //view TARIFA DIURNA
  viewtariff1();
}
if (ISRenter == 3 && derecha == 1 && ISRabajo == 2&& izquierda==1)
  //TARIFA NOCTURNA
{
  viewtariff2();
}
   if (ISRenter == 2 && derecha == 1 && ISRabajo == 2 && izquierda==0)
  {
    informacion();
  }

  if (ISRenter == 1 && derecha == 2 && ISRabajo == 0 && izquierda==1)
  {
```

```
        ocupado();
    }
    if (ISRenter == 2 && derecha == 2 && ISRabajo == 0 && izquierda==1)
    {
      pagar();
    }
    if (ISRenter == 1 && derecha == 3 && ISRabajo == 0 && izquierda==0||ISRenter == 1
    {
      opcion_parametros();
    }
    if (ISRenter == 2 && derecha == 3 && ISRabajo == 0 && izquierda==0)
    {
      impresion_parametros();
    }

    if (ISRenter == 1 && derecha == 3 && ISRabajo == 1 && izquierda==0)
    {
      opcion_diario();

    }

    if (ISRenter == 2 && derecha == 5 && ISRabajo == 1 && izquierda==1)
    {
//      impresion_diario();
    }
    if (ISRenter == 1 && derecha == 3 && ISRabajo == 2 && izquierda==0)
    {
      opcion_mensual();
    }
    if (ISRenter == 2 && derecha == 3 && ISRabajo == 2 && izquierda==0)
    {
  //    impresion_mensual();
    }
    if (ISRenter == 1 && derecha == 3 && ISRabajo == 3 && izquierda==0)
```

```
    {
      opcion_total();
    }
    if (ISRenter == 2 && derecha == 3 && ISRabajo == 3 && izquierda==0)
    {
    //  impresion_total();
    }

  if (Flagg==true)
  {
    escritura();
    Flagg=false;
  }
}
void ENter()
{
  buttonState = digitalRead(ENTER);
  if (buttonState != lastButtonState)
  {
    if (buttonState == LOW)
    {
      ISRenter++;
      Flag=true;
    }
  }
  lastButtonState = buttonState;
}
void ATras()
{
  buttonState1 = digitalRead(ATRAS);
  if (buttonState1 != lastButtonState1)
  {
    if (buttonState1 == LOW)
    {
```

```
        ISRatras++;
        ISRenter--;
        Flag=true;
        if(ISRenter==-1)
        {
          ISRenter=0;
        }
      }
    }
    lastButtonState1 = buttonState1;
}
void ABajo()
{
    buttonState2 = digitalRead(ABAJO);
    if (buttonState2 != lastButtonState2)
    {
      if (buttonState2 == LOW)
      {
        ISRabajo++;
      }
    }
    lastButtonState2 = buttonState2;
}
void ARriba()
{
    buttonState5 = digitalRead(ARRIBA);
    if (buttonState5 != lastButtonState5)
    {
      if (buttonState5 == LOW)
      {
        ISRarriba++;
        ISRabajo--;
        if(ISRabajo==-1)
        {
```

```
        ISRabajo=0;
      }
    }
  }
  lastButtonState5 = buttonState5;
}
void right()
{
    buttonState3 = digitalRead(DERECHA);
  if (buttonState3 != lastButtonState3)
  {

    if (buttonState3 == LOW)
    {
      derecha++;
    }
  }
  lastButtonState3 = buttonState3;
}
void left()
{
  buttonState4 = digitalRead(IZQUIERDA);
  if (buttonState4 != lastButtonState4)
  {
    if (buttonState4 == LOW)
    {
      izquierda++;
      derecha--;
      if (izquierda == -1)
      {
        izquierda = 0;
      }
      if (derecha == 0 || derecha ==-1)
      {
```

```
        derecha = 1;
      }
    }
  }
  lastButtonState4 = buttonState4;
}
void reloj()
{
  char retc[] = {'O', '\O'};
  char retc1[] = {'O', '\O'};
  char retc2[] = {'O', '\O'};
  char retc3[] = {'O', '\O'};
  retc[0] = Decena1 + 'O';
  retc1[0] = Unidad1 + 'O';
  retc2[0] = Decena2 + 'O';
  retc3[0] = Unidad2 + 'O';
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(86);
  paint.SetHeight(60);

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0,4, retc, &Font96, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 40, 20, paint.GetWidth(), paint.GetHeight());

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0,4, retc1, &Font96, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 40, 65, paint.GetWidth(), paint.GetHeight());

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0,4, ":", &Font96, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 40, 115, paint.GetWidth(), paint.GetHeight());

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0,4, retc2, &Font96, COLORED);
```

```
  epd.SetFrameMemory(paint.GetImage(), 40, 166, paint.GetWidth(), paint.GetHeight());

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0,4, retc3, &Font96, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 40, 216, paint.GetWidth(), paint.GetHeight());
}
void horario()
{
  DateTime now = rtc.now();
  int h = now.hour();
  int m = now.minute();
  Decena1 = h / 10;
  Unidad1 = (h - Decena1 * 10);
  Decena2 = m / 10;
  Unidad2 = (m - Decena2 * 10);
}

void letranegra()
{
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(30);
  paint.SetHeight(296);
  paint.Clear(COLORED);
}
void letranormal()
{
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(30);
  paint.SetHeight(296);
  paint.Clear(UNCOLORED);
}
void inicio1()
{
  epd.SetFrameMemory(IMAGE_DATA);
```

```
horario();
reloj();
paint.SetRotate(ROTATE_90);
paint.SetWidth(20);
paint.SetHeight(80);
paint.Clear(COLORED);
paint.DrawStringAt(5, 4, "Libre", &Font20, UNCOLORED);
epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight());
paint.SetRotate(ROTATE_90);
paint.SetWidth(20);
paint.SetHeight(120);
paint.Clear(UNCOLORED);
paint.DrawStringAt(7, 4, "Tarifac", &Font20, COLORED);
epd.SetFrameMemory(paint.GetImage(), 0, 80, paint.GetWidth(), paint.GetHeight());
paint.Clear(UNCOLORED);
paint.DrawStringAt(12, 4, "Imprim", &Font20, COLORED);
epd.SetFrameMemory(paint.GetImage(), 0, 190, paint.GetWidth(), paint.GetHeight());
epd.DisplayFrame();
Flagpantalla=false;
Flagpantalla1=false;
Flagocupado = false;
estado=false;
Flagpagar=false;
op pa = false;
ISRabajo = 0;
ISRarriba = 0;
izquierda = 0;
ISRatras = 0;
if (izquierda == 1)
  {
    izquierda = 0;
  }
    if (derecha == -1 || derecha == 0 || derecha >=4)
    {
```

```
        derecha = 1;
      }
  total3=0.00;
  total1=0.00;
  total2=0.00;
  pulsos1=0;
  contadortiempo=0.00;
  kilometros=0;
}
void inicio2()
{
  epd.SetFrameMemory(IMAGE_DATA);
  horario();
  reloj();
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(20);
  paint.SetHeight(80);
  paint.Clear(UNCOLORED);
  paint.DrawStringAt(5, 4, "Libre", &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight());
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(20);
  paint.SetHeight(120);
  paint.Clear(COLORED);
  paint.DrawStringAt(7, 4, "Tarifac", &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 80, paint.GetWidth(), paint.GetHeight());
  paint.Clear(UNCOLORED);
  paint.DrawStringAt(12, 4, "Imprim", &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 190, paint.GetWidth(), paint.GetHeight());
  epd.DisplayFrame();
  Flagpantalla=false;
  Flagpantalla1=false;
  Flagocupado = false;
  estado=false;
```

```
  total3=0.00;
  total1=0.00;
  total2=0.00;
  pulsos1=0;
  contadortiempo=0.00;
  calendario();
  ISRabajo = 0;
  ISRarriba = 0;
  ISRatras = 0;
  izquierda=1;
}
void inicio3()
{
  epd.SetFrameMemory(IMAGE_DATA);
  horario();
  reloj();
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(20);
  paint.SetHeight(80);
  paint.Clear(UNCOLORED);
  paint.DrawStringAt(5, 4, "Libre", &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight());
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(20);
  paint.SetHeight(120);
  paint.Clear(UNCOLORED);
  paint.DrawStringAt(7, 4, "Tarifac", &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 80, paint.GetWidth(), paint.GetHeight());
  paint.Clear(COLORED);
  paint.DrawStringAt(12, 4, "Imprim", &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 190, paint.GetWidth(), paint.GetHeight());
  epd.DisplayFrame();
  Flagpantalla=false;
  Flagpantalla1=false;
```

```
  Flagocupado = false;
  op_pa = false;
  ISRabajo = 0;
  ISRarriba = 0;
  ISRatras = 0;
  izquierda = 0;
  if (izquierda == 1)
    {
      izquierda = 0;
    }
   if (derecha == 4)
    {
      derecha = 1;
      izquierda = 0;
    }
}
void pantalla()
{
  epd.SetFrameMemory(IMAGE_DATA1);
//  letranegra();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[0])));
//  paint.DrawStringAt(10, 8, cc, &Font20, UNCOLORED);
//  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight());
//
//  letranormal();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[1])));
//  paint.DrawStringAt(10, 8, cc, &Font20, COLORED);
//  epd.SetFrameMemory(paint.GetImage(), 70, 0, paint.GetWidth(), paint.GetHeight());/
//
//  letranormal();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[2])));
//  paint.DrawStringAt(10, 8, cc, &Font20, COLORED);
//  epd.SetFrameMemory(paint.GetImage(), 35, 0, paint.GetWidth(), paint.GetHeight());/
//
```

```
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[3])));
//   paint.DrawStringAt(10, 8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 5, 0, paint.GetWidth(), paint.GetHeight());//
  epd.DisplayFrame();
  Flagpantalla=true;
  Flagpantalla1=false;
  Flagparameters=false;
  ISRarriba = 0;
  ISRatras = 0;
  izquierda = 0;
  derecha=1;
}
void pantalla1()
{
  epd.SetFrameMemory(IMAGE_DATA2);
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[0])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight());
//
//   letranegra();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[1])));
//   paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
//   epd.SetFrameMemory(paint.GetImage(), 70, 0, paint.GetWidth(), paint.GetHeight());/
//
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[2])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 35, 0, paint.GetWidth(), paint.GetHeight());/
//
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[3])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
```

```
//   epd.SetFrameMemory(paint.GetImage(), 5, 0, paint.GetWidth(), paint.GetHeight());//

  epd.DisplayFrame();
  Flagpantalla=false;
  Flagpantalla1=true;
  Flagtar=false;
  ISRarriba = 0;
  ISRatras = 0;
  izquierda = 1;
}
void pantalla2()
{
  epd.SetFrameMemory(IMAGE_DATA3);
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[0])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight());
//
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[1])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 70, 0, paint.GetWidth(), paint.GetHeight());/
//
//   letranegra();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[2])));
//   paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
//   epd.SetFrameMemory(paint.GetImage(), 35, 0, paint.GetWidth(), paint.GetHeight());/
//
//   letranormal();
//   strcpy_P(cc, (char*)pgm_read_word(&(string_table[3])));
//   paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//   epd.SetFrameMemory(paint.GetImage(), 5, 0, paint.GetWidth(), paint.GetHeight());//

  epd.DisplayFrame();
```

```
  valoresiniciales=true;
  Flagpantalla=true;
  Flagpantalla1=false;
  Flaginformacion=false;
  ISRatras = 0;
  izquierda = 0;
}
void pantalla3()
{
//  letranormal();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[0])));
//  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight());

//  letranormal();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[1])));
//  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//  epd.SetFrameMemory(paint.GetImage(), 70, 0, paint.GetWidth(), paint.GetHeight());/
//
//  letranormal();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[2])));
//  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
//  epd.SetFrameMemory(paint.GetImage(), 35, 0, paint.GetWidth(), paint.GetHeight());/
//
  letranegra();
//  strcpy_P(cc, (char*)pgm_read_word(&(string_table[3])));
//  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
//  epd.SetFrameMemory(paint.GetImage(), 5, 0, paint.GetWidth(), paint.GetHeight());//
//
  epd.SetFrameMemory(IMAGE_DATA4);
  epd.DisplayFrame();
  Flagpantalla1=false;
  Flagpantalla=true;
  ISRatras = 0;
```

```
  izquierda = 0;
}

void parameters()
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[10])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  (EEPROM.get(eeAddress5, ruc));
  paint.DrawStringAt(155, 5, ruc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[11])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  EEPROM.get(eeAddress6, td);
  cad=dtostrf(td, 4, 2, B);
  cad.toCharArray(B, 6);
  paint.DrawStringAt(185, 5, B, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[12])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  (EEPROM.get(eeAddress9, placa));
  paint.DrawStringAt(100, 5, placa, &Font20 , COLORED);
  epd.SetFrameMemory(paint.GetImage(), 35, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[13])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  (EEPROM.get(eeAddress10, placa));
  paint.DrawStringAt(200, 5, placa, &Font20 , COLORED);
```

```
  epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight()); // T

  epd.DisplayFrame();
  Flagparameters=true;
  ISRabajo = 0;
  ISRarriba = 0;
  izquierda = 0;
  ISRatras = 0;
}
void tariffs()
{
  epd.SetFrameMemory(IMAGE_DATA);

  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[4])));
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[5])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 50, 0, paint.GetWidth(), paint.GetHeight()); //
  epd.DisplayFrame();
  Flagtar=true;
  Flagviewtar1 = false;
  Flagviewtar2 = false;
  izquierda = 1;
  ISRarriba = 0;
  ISRatras = 0;
 }
void tariffs1()
{
  epd.SetFrameMemory(IMAGE_DATA);
```

```
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[4])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight()); //

  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[5])));
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 50, 0, paint.GetWidth(), paint.GetHeight()); //

  epd.DisplayFrame();
  Flagviewtar1 = false;
  Flagviewtar2 = false;
  izquierda = 1;
  ISRarriba = 0;
  ISRatras = 0;
  derecha=1;
}
void viewtariff1()
{
  //view TARIFA DIURNA
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[4])));
  paint.DrawStringAt(40, 5, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[6])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  EEPROM.get(eeAddress11, td);
  cad=dtostrf(td, 4, 2, B);
  cad.toCharArray(B, 6);
  paint.DrawStringAt(210, 5, B, &Font20 , COLORED);
```

```
epd.SetFrameMemory(paint.GetImage(), 75, 0, paint.GetWidth(), paint.GetHeight()); //

letranormal();
strcpy_P(cc, (char*)pgm_read_word(&(string_table[7])));
paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
EEPROM.get(eeAddress12, td);
cad=dtostrf(td, 4, 2, B);
cad.toCharArray(B, 6);
paint.DrawStringAt(140, 5, B, &Font20 , COLORED);
epd.SetFrameMemory(paint.GetImage(), 50, 0, paint.GetWidth(), paint.GetHeight()); //

letranormal();
strcpy_P(cc, (char*)pgm_read_word(&(string_table[8])));
paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
EEPROM.get(eeAddress13, td);
cad=dtostrf(td, 4, 2, B);
cad.toCharArray(B, 6);
paint.DrawStringAt(110, 5, B, &Font20 , COLORED);
epd.SetFrameMemory(paint.GetImage(), 25, 0, paint.GetWidth(), paint.GetHeight()); //

letranormal();
strcpy_P(cc, (char*)pgm_read_word(&(string_table[9])));
paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
EEPROM.get(eeAddress14, td);
cad=dtostrf(td, 4, 2, B);
cad.toCharArray(B, 6);
paint.DrawStringAt(170, 5, B, &Font20 , COLORED);
epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight()); //T
epd.DisplayFrame();
Flagviewtar1 = true;
Flagviewtar2 = false;
izquierda = 1;
ISRarriba = 0;
ISRatras = 0;
```

```
  derecha=1;
  ISRabajo = 1;

}
void viewtariff2()
//TARIFA NOCTURNA
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[5])));
  paint.DrawStringAt(40, 5, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[6])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  EEPROM.get(eeAddress16, td);
  cad=dtostrf(td, 4, 3, B);
  cad.toCharArray(B, 5);
  paint.DrawStringAt(210, 5, B, &Font20 , COLORED);
  epd.SetFrameMemory(paint.GetImage(), 75, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[7])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  EEPROM.get(eeAddress17, td);
  cad=dtostrf(td, 4, 3, B);
  cad.toCharArray(B, 5);
  paint.DrawStringAt(140, 5, B, &Font20 , COLORED);
  epd.SetFrameMemory(paint.GetImage(), 50, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[8])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
```

```
  EEPROM.get(eeAddress18, td);
  cad=dtostrf(td, 4,3, B);
  cad.toCharArray(B, 5);
  paint.DrawStringAt(110, 5, B, &Font20 , COLORED);
  epd.SetFrameMemory(paint.GetImage(), 25, 0, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[9])));
  paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
  EEPROM.get(eeAddress19, td);
  cad=dtostrf(td, 4, 3, B);
  cad.toCharArray(B, 5);
  paint.DrawStringAt(170, 5, B, &Font20 , COLORED);
  epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight()); //T
  epd.DisplayFrame();
  Flagviewtar1 = false;
  Flagviewtar2 = true;
  izquierda = 1;
  ISRarriba = 0;
  ISRatras = 0;
  derecha=1;
  ISRabajo = 2;
}
void informacion()
{
    epd.SetFrameMemory(IMAGE_DATA);
    letranormal();
    strcpy_P(cc, (char*)pgm_read_word(&(string_table[14])));
    paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
    (EEPROM.get(eeAddress, compania));
    paint.DrawStringAt(135, 5, compania, &Font20, COLORED);
    epd.SetFrameMemory(paint.GetImage(), 100, 0, paint.GetWidth(), paint.GetHeight());

    letranormal();
```

```
    strcpy_P(cc, (char*)pgm_read_word(&(string_table[15])));
    paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
    (EEPROM.get(eeAddress1, ciu));
    paint.DrawStringAt(110, 5, ciu, &Font20 , COLORED);
    epd.SetFrameMemory(paint.GetImage(), 75, 0, paint.GetWidth(), paint.GetHeight()); /

    letranormal();
    strcpy_P(cc, (char*)pgm_read_word(&(string_table[16])));
    paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
    (EEPROM.get(eeAddress2, ruc));
    paint.DrawStringAt(70, 5, ruc, &Font20 , COLORED);
    epd.SetFrameMemory(paint.GetImage(), 50, 0, paint.GetWidth(), paint.GetHeight()); /

    letranormal();
    strcpy_P(cc, (char*)pgm_read_word(&(string_table[17])));
    paint.DrawStringAt(10, 5, cc, &Font20, COLORED);
    (EEPROM.get(eeAddress3, ruc));
    paint.DrawStringAt(125, 5, ruc, &Font20 , COLORED);
    epd.SetFrameMemory(paint.GetImage(), 25, 0, paint.GetWidth(), paint.GetHeight()); /

    letranormal();
    strcpy_P(cc, (char*)pgm_read_word(&(string_table[18])));
    paint.DrawStringAt(10, 5, cc, &Font20 , COLORED);
    (EEPROM.get(eeAddress4, compania));
    paint.DrawStringAt(125,5, compania, &Font20 , COLORED);
    epd.SetFrameMemory(paint.GetImage(), 0, 0, paint.GetWidth(), paint.GetHeight()); //
    epd.DisplayFrame();
    Flaginformacion=true;
    ISRarriba = 0;
    izquierda = 0;
    ISRatras = 0;
    ISRabajo = 2;
}
void ocupado()
```

```
{
        epd.SetFrameMemory(IMAGE_DATA);
        control();
        paint.SetRotate(ROTATE_90);
        paint.SetWidth(20);
        paint.SetHeight(90);

        paint.Clear(COLORED);
        paint.DrawStringAt(12, 4, "Pagar", &Font20, UNCOLORED);
        epd.SetFrameMemory(paint.GetImage(), 0, 70, paint.GetWidth(), paint.GetHeight())
        epd.DisplayFrame();

        ISRabajo = 0;
        ISRarriba = 0;
        izquierda =1;
        ISRatras = 0;
        Flagocupado = true;
        estado=true;
}
void pagar()
{
        epd.SetFrameMemory(IMAGE_DATA);
        escritura1();
        impresion_viaje();
        inicio1();
        Flagpagar = true;
        Flagocupado=false;
        estado=false;
        ISRabajo = 0;
        ISRarriba = 0;
        izquierda = 1;
        ISRatras = 0;
}
void opcion_parametros()
```

```
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[19])));
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 10, paint.GetWidth(), paint.GetHeight()); /

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[22])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[20])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 35, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[21])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 5, 10, paint.GetWidth(), paint.GetHeight()); //
  epd.DisplayFrame();
  ISRabajo = 0;
  ISRarriba = 0;
  izquierda = 0;
  Flagpagar=false;
  op_pa=true;
 }
void opcion_diario()
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[19])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
```

```
  epd.SetFrameMemory(paint.GetImage(), 100, 10, paint.GetWidth(), paint.GetHeight()); /

  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[22])));
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[20])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 35, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[21])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 5, 10, paint.GetWidth(), paint.GetHeight()); //
  epd.DisplayFrame();
  izquierda = 0;
}
void opcion_mensual()
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[19])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 10, paint.GetWidth(), paint.GetHeight()); /

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[22])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 10, paint.GetWidth(), paint.GetHeight()); //

  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[20])));
```

```
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 35, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[21])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 5, 10, paint.GetWidth(), paint.GetHeight()); //
  epd.DisplayFrame();
  izquierda = 0;
}
void opcion_total()
{
  epd.SetFrameMemory(IMAGE_DATA);
  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[19])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 100, 10, paint.GetWidth(), paint.GetHeight()); /

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[22])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 70, 10, paint.GetWidth(), paint.GetHeight()); //

  letranormal();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[20])));
  paint.DrawStringAt(10,8, cc, &Font20, COLORED);
  epd.SetFrameMemory(paint.GetImage(), 35, 10, paint.GetWidth(), paint.GetHeight()); //

  letranegra();
  strcpy_P(cc, (char*)pgm_read_word(&(string_table[21])));
  paint.DrawStringAt(10,8, cc, &Font20, UNCOLORED);
  epd.SetFrameMemory(paint.GetImage(), 5, 10, paint.GetWidth(), paint.GetHeight()); //
  epd.DisplayFrame();
  izquierda = 0;
```

```
}
//void espsend(String d)
//{
//  String cipSend = "AT+CIPSEND=";
//  cipSend += connectionId;
//  cipSend += ",";
//  cipSend += d.length();
//  cipSend += "\r\n";
//  sendData(cipSend, 100, DEBUG);
//  sendData(d, 100, DEBUG);
//}
//////////////gets the data from esp and displays in serial monitor////////////////
//String sendData(String command, const int timeout, boolean debug)
//{
//  String response = "";
//  Serial2.print(command);
//  long int time = millis();
//  while ( (time + timeout) > millis())
//  {
//    while (Serial2.available() > 0)
//    {
//      char c = Serial2.read(); // read the next character.
//      response += c;
//    }
//  }
//  if (debug)
//  {
//    Serial.print(response); //displays the esp response messages in arduino Serial2
//  }
//  return response;
//}
void calendario()
{
   DateTime now = rtc.now();
```

```
weekDay = now.dayOfTheWeek();
h = now.hour();
m = now.minute();
if(weekDay == 6 || weekDay == 0)
{
 a=2;
}
else if(weekDay ==1 || weekDay ==2 || weekDay==3|| weekDay==4 || weekDay==5)
{
 if((h >= 6)||(h < 17))
 {
 a=1;
 }
 if((h >= 17)||(h<6))
 {
  a=2;
 }
}
EEPROM.get(eeAddress11, temp1);
EEPROM.get(eeAddress12, arra1);
EEPROM.get(eeAddress13, porkm1);
EEPROM.get(eeAddress14, mines1);
EEPROM.get(eeAddress15, ext1);

EEPROM.get(eeAddress16, temp2);
EEPROM.get(eeAddress17, arra2);
EEPROM.get(eeAddress18, porkm2);
EEPROM.get(eeAddress19, mines2);
EEPROM.get(eeAddress20, ext2);
 if(a==1)
{
n_tarifa=1;
temp=temp1;
arra=arra1;
```

```
    porkm=porkm1;
    mines=mines1;
    ext=ext1;
    }
   if(a==2)
    {
    n_tarifa=2;
    temp=temp2;
    arra=arra2;
    porkm=porkm2;
    mines=mines2;
    ext=ext2;
    }
     val1=arra+0.00;
     val2=porkm+0.00;
     val3=mines+0.00;
     val4=ext+0.00;
     total=temp+0.00;
}
void control()
{
  total3=total+total1+total2;
  cad=dtostrf(total3, 4, 2, B);
  cad.toCharArray(B, 6);
  paint.SetRotate(ROTATE_90);
  paint.SetWidth(86);
  paint.SetHeight(296);

  paint.Clear(UNCOLORED);
  paint.DrawStringAt(0, 4, B, &Font96,COLORED);
  epd.SetFrameMemory(paint.GetImage(), 40, 10, paint.GetWidth(), paint.GetHeight());
}

void tiempo()
```

```
{
    tiempo2=millis();
    if(tiempo2>(tiempo1+1000))
    {
      tiempo1=millis();
      contadortiempo=contadortiempo+val3/60;
    }
}
void escritura()
{
  DateTime now = rtc.now();
  EEPROM.get(eeAddress30,eventos);
  EEPROM.get(eeAddress33,eventos1);
  EEPROM.get(eeAddress31,compania);
  EEPROM.get(eeAddress32,compania1);
  String aar="";
  String aar1="";
  String ev="";
  String ev1="";
  ev=String(eventos);
  ev1=String(eventos1);
  aar=String(compania);
  aar1=String(compania1);

     if(aar != aar1 || ev!=ev1)

  {
    myFile = SD.open("Cambios.txt", FILE_WRITE);//abrimos  el archivo
    if (myFile)
    {
          myFile.print("Responsable=");
          myFile.print(compania);
          myFile.print(' ');
          myFile.print("Fecha=");
```

```
            myFile.print(now.year(), DEC);
            myFile.print('/');
            myFile.print(now.month(), DEC);
            myFile.print('/');
            myFile.print(now.day(), DEC);
            myFile.print(' ');
            myFile.print(now.hour(), DEC);
            myFile.print(':');
            myFile.print(now.minute(), DEC);
            myFile.print(':');
            myFile.print(now.second(), DEC);
            myFile.println();
            myFile.close();
            EEPROM.put(eeAddress32,compania);
            EEPROM.put(eeAddress33,eventos);
    }
  }
}
void impresion_parametros()
{
  EEPROM.get(eeAddress11, temp1);
  EEPROM.get(eeAddress14, mines1);
  EEPROM.get(eeAddress16, temp2);
  EEPROM.get(eeAddress19, mines2);
  EEPROM.get(eeAddress9, placa);
  EEPROM.get(eeAddress6, td);
   printer.justify('C');
   printer.doubleHeightOn();
   printer.println(F("PARAMETROS"));
   printer.doubleHeightOff();
   printer.setSize('S');
   printer.println(F("IBARRA"));
   printer.println(F("Titular: David Guzman"    ));
   printer.setSize('S');
```

```
  printer.println(F("_____"));

  printer.justify('L');
  printer.print(("Num. serie:          "));
  EEPROM.get(eeAddress5, ruc);
  printer.println((ruc));

  printer.print(F("CONSTANTE K:          "));
  printer.println((td));
  printer.print(F("PLACA:                "));
  printer.println((placa));
  printer.setSize('S');
  printer.println(F("----------------------"));
  printer.print(F("Num.REGISTRO:              "));
  (EEPROM.get(eeAddress10, placa));
  printer.println((placa));
  printer.print(F("TARIFA DIURNA(TF):         "));
  printer.println((temp1));
  printer.print(F("MINUTO DE ESPERA(TF):      "));
  printer.println((mines1));
  printer.print(F("TARIFA NOCTURNA(TN):       "));
  printer.println((temp2));
  printer.print(F("MINUTO DE ESPERA(TN):      "));
  printer.println((mines2));
  printer.justify('C');
  printer.println(F("GRACIAS POR ELEGIRNOS"));
  printer.feed(2);
  ISRenter=1;
}
void impresion_viaje()
{
  DateTime now = rtc.now();
  weekDay = now.dayOfTheWeek();
  year_fin= now.year();
```

```
mes_fin= now.month();
dia_fin= now.day();
hora_fin= now.hour();
min_fin= now.minute();
EEPROM.get(eeAddress11, temp1);
EEPROM.get(eeAddress14, mines1);
EEPROM.get(eeAddress16, temp2);
EEPROM.get(eeAddress19, mines2);
EEPROM.get(eeAddress9, placa);
EEPROM.get(eeAddress6, td);
EEPROM.get(eeAddress34, kilometros);
printer.justify('C');
printer.doubleHeightOn();
printer.println(F("TICKET DE CONTROL"));
printer.doubleHeightOff();
printer.setSize('S');
printer.println(F("IBARRA" ));
printer.println(F("Titular: David Guzman"    ));
printer.justify('L');
printer.print(F("PLACA:                    "));
printer.println((placa));
printer.setSize('S');
printer.println(F("--------------------------------"));
printer.justify('L');
printer.print(F("Tarifa utilizada:          "));
printer.println(n_tarifa);
printer.justify('L');
printer.print(F("INICIO DEL VIAJE:          "));
printer.print(h,DEC);
printer.print(':');
printer.println(m,DEC);
printer.justify('L');
printer.print(F("FIN:       "));
printer.print(year_fin);
```

```
  printer.print('/');
  printer.print(mes_fin);
  printer.print('/');
  printer.print(dia_fin);
  printer.print("          ");
  printer.justify('L');
  printer.print(hora_fin);
  printer.print(':');
  printer.println(min_fin);
 EEPROM.get(eeAddress34, kilometros);
  printer.print(F("Km recorrios:                "));
  printer.println((kilometros));
  printer.justify('L');
  printer.print(F("Valor de tarifa:             "));
  printer.println(total);
  printer.justify('L');
  printer.print(F("Valor por KM:                "));
  printer.println(total1);
  printer.justify('L');
  printer.print(F("Valor por minuto de espera "));
  printer.println(total2);
  printer.setSize('S');
  printer.println(F("--------------------------------"));
  printer.justify('C');
  printer.doubleHeightOn();
  printer.println(F("A PAGAR"));
  printer.print(F("$                            "));
  printer.println(total3);
  printer.justify('C');
  printer.setSize('S');
  printer.println(F("------------------------------"));
  printer.println(F("GRACIAS POR ELEGIRNOS"));
  printer.feed(3);
}
```

```
void escritura1()
{
  DateTime now = rtc.now();
   myFile = SD.open("KML.txt", FILE_WRITE); //abrimos  el archivo
    if (myFile)
    {
           myFile.print("KML: ");
           myFile.print(kilometros);
           myFile.print(' ');
           myFile.print("Fecha=");
           myFile.print(now.year(), DEC);
           myFile.print('/');
           myFile.print(now.month(), DEC);
           myFile.print('/');
           myFile.println(now.day(), DEC);
           myFile.close();
           EEPROM.put(eeAddress34,kilometros);
    }
  }
```

.B. Programas Utilizados

.B.1. Diagrama en yEd Graph Editor

En la Figura 3.2 se muestra el diagrama de estados desarrollado en yEd.

.B.2. Diseño del circuito y PCB en Eagle

La Figura 15 muestra el diseño del circuito implementado en el dispotivo y la Figura 16 muestra el diseño PCB, ambos han sido realizados en Eagle.

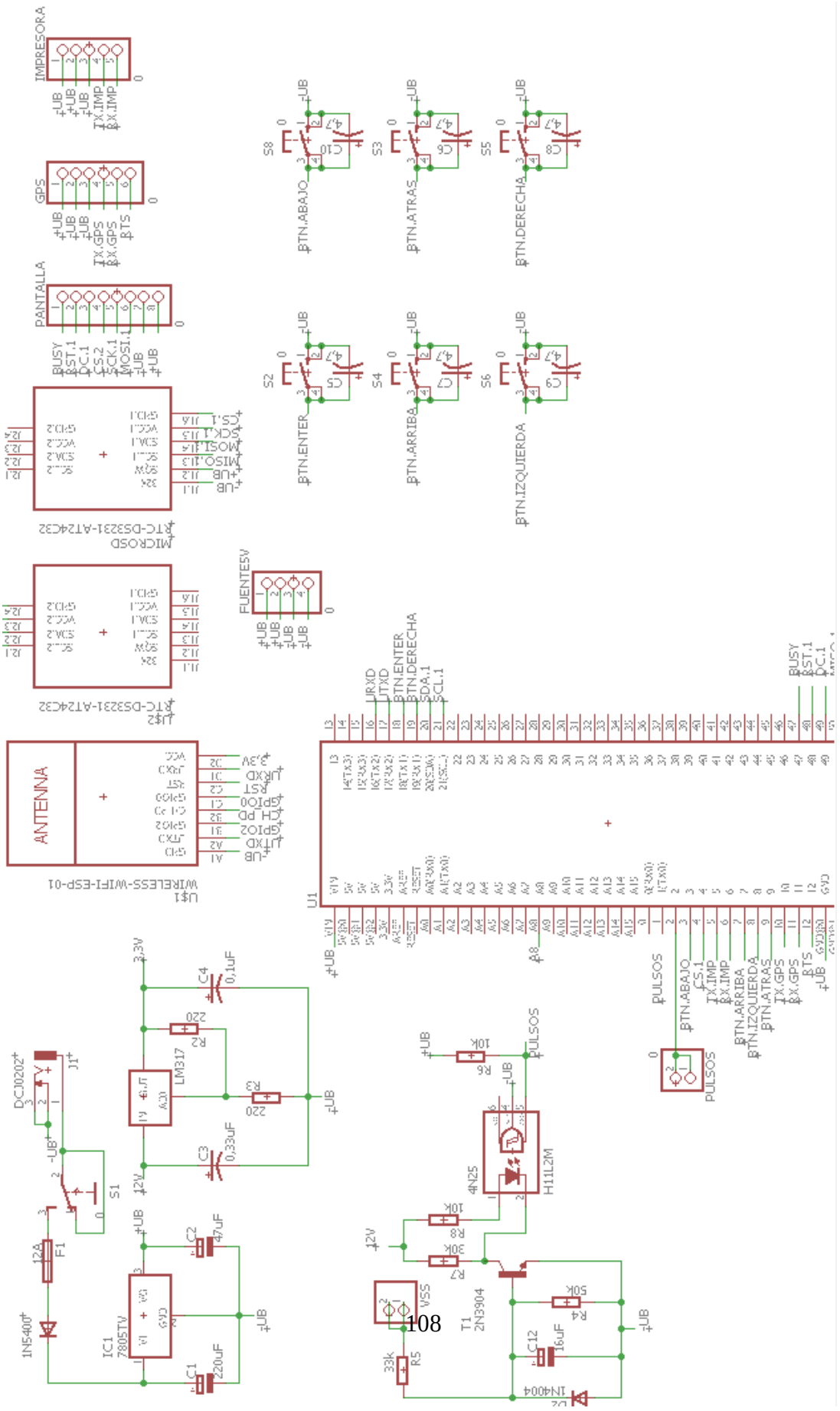


Figura 15: Diseño del circuito

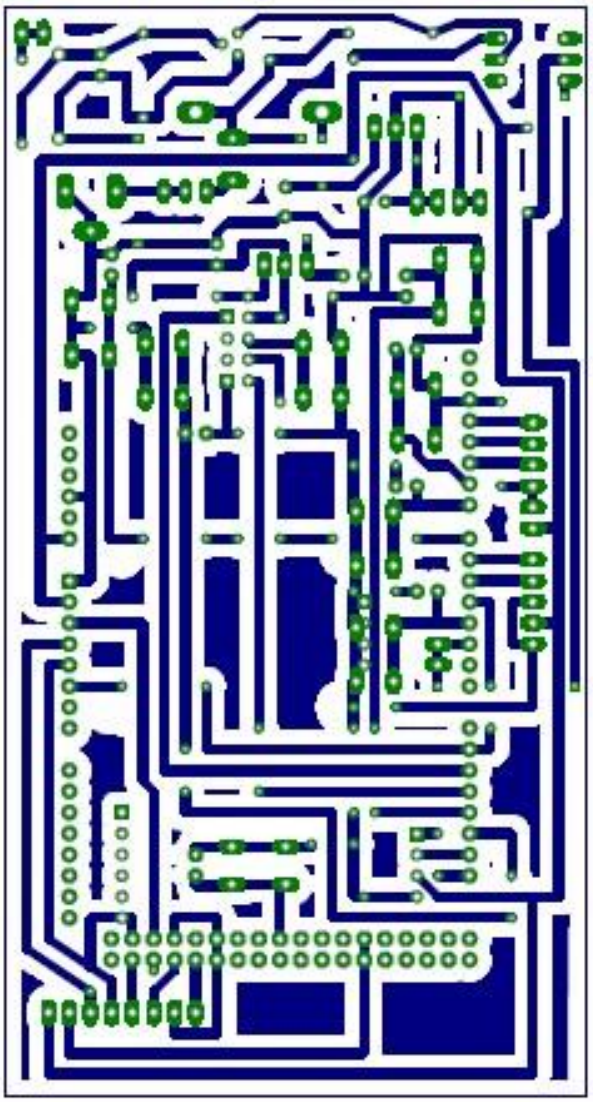


Figura 16: Diseño PCB

.B.3. Carcasa en Solid Works

En la Figura 4.1 se muestra la carcasa diseñada en solid works en donde se ubican y ordenan todos los elementos del hardware.

A continuación se muestran los planos de la carcasa.