

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Diseño de un dispositivo sincronizador de semáforos basado en pulsos de un reloj externo para grandes avenidas urbanas”

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN TELECOMUNICACIONES

Presentado por:

JHON LENON RAMÍREZ ANDRADE

JAIME ALFONSO VACA SERRANO

GUAYAQUIL – ECUADOR

AÑO: 2022

DEDICATORIA

El trabajo realizado en este proyecto de graduación lo dedico a mis padres, Wilfrido Ramírez y María Andrade y a mi hermano Andy Ramírez, quienes han sido partícipes durante estos años de preparación académica en los cuales me brindaron soporte y motivación día a día, con sus esfuerzos y sacrificios que siempre tendré presente. Este logro académico se lo dedico a ustedes.

Jhon Ramírez Andrade

AGRADECIMIENTOS

Quiero otorgar mi más sincero agradecimiento a esta prestigiosa institución académica, la Escuela Superior Politécnica del Litoral, y a su equipo de educadores que compartieron sus conocimientos intelectuales, así como también ética y valores. Al tutor PhD. Juan Carlos Avilés Castillo quien me orientó y aconsejó para la realización de este proyecto. Y al PhD. Washington Medina Moreira por su tiempo para la instrucción de la elaboración de este documento.

Jhon Ramírez Andrade

DEDICATORIA

Primero, le dedico este proyecto de tesis a Dios y luego a mi mamá, Jenny, quien ha hecho incontables sacrificios, a mi tía Lourdes y el resto de mi familia quienes siempre han estado ahí para apoyarme en todo momento. Finalmente, a mis amigos quienes me han ayudado a motivarme en los momentos más bajos permitiéndome salir adelante a pesar de las adversidades presentadas.

Jaime Vaca Serrano

AGRADECIMIENTOS

Agradezco de todo corazón a mi mamá Jenny quien ha estado en los momentos más complicados de mi vida, y que, sin ella, no estaría culminando mi etapa académica. También doy mi más sincero agradecimiento a la Escuela Superior Politécnica del Litoral por ser un lugar que me ha brindado muchos nuevos conocimientos con su equipo de docentes. Al tutor PhD. Juan Carlos Avilés Castillo por orientarme y guiarme durante el transcurso de todo este proyecto. Al PhD Washington Medina Moreira por sus instrucciones que me ayudaron a mejorar el desarrollo de este documento y del proyecto en general.

A mis amigos Mike, Karen y Cristhian por estar en los buenos y malos momentos con quienes además pude compartir parte de mi proceso universitario. Finalmente, a mi mejor amiga Valentina, quien se convirtió en la mejor compañera de carrera que pude pedir, y fue ese gran apoyo en los momentos más complicados de la carrera.

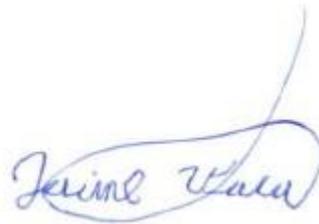
Jaime Vaca Serrano

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Jhon Lenon Ramírez Andrade* y *Jaime Alfonso Vaca Serrano* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Jhon Lenon Ramírez Andrade



Jaime Alfonso Vaca Serrano

TRIBUNAL DE EVALUACIÓN



Firmado electrónicamente por:
**WASHINGTON ADOLFO
MEDINA MOREIRA**

PhD. Washington Adolfo Medina Moreira

PROFESOR DE LA MATERIA INTEGRADORA



Firmado electrónicamente por:
**JUAN CARLOS AVILES
CASTILLO**

PhD. Juan Carlos Aviles Castillo

PROFESOR TUTOR DE TESIS

RESUMEN

En la presente tesis se realizará el diseño e implementación de un controlador de tráfico que se sincroniza mediante un pulso de reloj externo como respuesta a los problemas que presenta la ciudad de Guayaquil respecto a los atascamientos vehiculares producidos por sus semáforos que trabajan con tiempos constantes o que se encuentran mal sincronizados. El dispositivo trabajará de manera autónoma para no depender de la presencia de una persona para realizar los cambios flexibilizando su integración.

Este proceso de sincronización se iniciará una vez terminado el ciclo del semáforo y no de forma instantánea puesto que, presentar cambios bruscos en las luces del dispositivo podría ocasionar accidentes de tráfico y con ello, pérdidas humanas.

Finalmente, se tendrá un controlador de tráfico capaz de variar el tiempo de duración de los ciclos del semáforo de manera autónoma sincronizándose con el pulso externo sin alterar el orden de la secuencia de juego de las luces del dispositivo.

Palabras clave: sincronización, pulso externo, controlador de tráfico, ciclo, secuencia.

ABSTRACT

In the following thesis, the design and implementation of a traffic light prototype synchronized by an external clock pulse will be carried out in response to the problems that the city of Guayaquil experiences regarding traffic jams. Such difficulties are produced by traffic lights that work with constant times or because they work out of sync. The proposed device will work autonomously so as not to depend on the presence of a person to perform parameter changes, making its integration more flexible.

This synchronization process cannot be accomplished instantly. This process will start once the traffic light cycle has finished. Sudden changes in the device's lights could cause traffic accidents and human losses.

Finally, there will be a traffic controller capable of varying the duration time of the traffic light cycles autonomously, synchronized with an external pulse and without altering the order of the device's lights sequence.

Key Words: *synchronization, external pulse, traffic controller, cycle, sequence.*

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS.....	VI
SIMBOLOGÍA.....	VII
ÍNDICE DE ILUSTRACIONES.....	VIII
ÍNDICE DE TABLAS.....	X
CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.1 Descripción del Problema.....	1
1.2 Justificación.....	3
1.3 Objetivos.....	4
1.3.1 Objetivo General.....	4
1.3.2 Objetivos Específicos.....	4
1.4 Estado del Arte.....	4
1.5 Alcance.....	9
1.6 Metodología.....	10
CAPÍTULO II.....	11
2. MARCO TEÓRICO.....	11
2.1 Microprocesadores.....	11
2.1.1 Microprocesador Pentium.....	12
2.1.2 Xeon Dual Core.....	12
2.1.3 Microprocesador Intel Atom.....	13
2.1.4 Intel Core.....	13

2.1.4.1 Intel Core i3.....	14
2.1.4.2 Intel Core i5.....	14
2.1.4.3 Intel Core i7.....	15
2.1.4.4 Intel Core i9.....	15
2.2 Microcontroladores	16
2.2.1 Arduino.....	16
2.2.1.1 Principales Ventajas.....	17
2.2.1.2 Principales Desventajas.....	17
2.2.2 Raspberry Pi.....	18
2.2.3 ESP 8266	19
2.2.4 ESP 32	20
2.3 Lenguajes de Programación.....	21
2.3.1 Python	21
2.3.2 C++	22
2.3.3 Arduino.....	23
2.4 Placa Protoboard.....	23
2.5 Reloj de Tiempo Real o RTC.....	24
2.6 Resistores.....	25
2.7 Luces LED	26
2.8 Jumpers.....	27
CAPÍTULO III	28
3. ESCENARIO.....	28
3.1 Diseño de la Propuesta	28
3.1.1 Proceso de Alimentación.....	29
3.1.2 Proceso de Control.....	30
3.1.3 Proceso de Sincronización.....	30
3.2 Materiales Utilizados	30

3.2.1 Hardware.....	30
3.2.2 Software	32
3.3 Costos de Materiales.....	33
CAPÍTULO IV.....	35
4. ANÁLISIS DE RESULTADOS	35
4.1 Implementación del dispositivo.....	35
4.2 Funcionamiento del dispositivo.....	42
4.3 Pruebas realizadas.....	47
CONCLUSIONES Y RECOMENDACIONES	50
Conclusiones	50
Recomendaciones	50
BIBLIOGRAFÍA.....	52
ANEXOS.....	63
Anexo 1 – Código de programación (declaración de puertos de salida).....	64
Anexo 2 – Código de programación (tiempo de los eventos)	64
Anexo 3 – Código de programación (secuencia normal)	64
Anexo 4 – Código de programación (recepción de pulso externo)	66
Anexo 5 – Código de programación (proceso de sincronización).....	66
Anexo 6 – Código de programación (retoma secuencia normal)	68
Anexo 7 – Pruebas con pulso de llegada a los 14 sg	68
Anexo 8 – Pruebas con pulso de llegada a los 35 sg	70
Anexo 9 – Pruebas con pulso de llegada a los 28 sg	71

ABREVIATURAS

ATM	Agencia de Tránsito Municipal
GPS	Global Positioning System
IoT	Internet of Things
CV2X	Cellular Vehicle-to-Everything
IA	Inteligencia Artificial
SICESSIS	Sistema Centralizado de Sincronización de Semáforos Inalámbricos Solares
CI	Circuito Integrado
LED	Light Emitting Diode
RTC	Real Time Clock
DVD	Digital Versatile Disc
LAN	Local Area Network
CPU	Central Processing Unit
RAM	Random Access Memory
API	Application Programming Interface
SD	Secure Digital
TCP/IP	Transmission Control Protocol/Internet Protocol
SRAM	Static Random Access Memory
ROM	Read Only Memory
USB	Universal Serial Bus
GPIO	General Purpose Input/Output
PCB	Printed Circuit Board
SDA	Serial Data
SCL	Serial Clock

SIMBOLOGÍA

dB	Decibelio
sg	Segundos
Km/h	Kilómetros por hora
MHz	MegaHertz
KB	Kilobyte
KiB	Kibibyte
dBm	Decibelio milivatio
Mbps	Megabits por segundo

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Microprocesador Intel Pentium [38]	12
Ilustración 2: Microprocesador Intel Xeon [40]	13
Ilustración 3: Microprocesador Intel Atom [42]	13
Ilustración 4: Microprocesador Intel Core i3 [44]	14
Ilustración 5: Microprocesador Intel i5 [45]	15
Ilustración 6: Microprocesador Intel Core i7 [38]	15
Ilustración 7: Microprocesador Intel i9 [46]	16
Ilustración 8: Arduino UNO [49]	17
Ilustración 9: Raspberry Pi [51]	19
Ilustración 10: Módulo ESP 8266 [53]	20
Ilustración 11: Módulo ESP 32 [56]	21
Ilustración 12: Placa Protoboard [61]	24
Ilustración 13: Dispositivo RTC	25
Ilustración 14: Resistores [64]	26
Ilustración 15: Diodos Led [66]	26
Ilustración 16: Jumpers macho-hembra [67]	27
Ilustración 17: Semáforo ubicado en el sector de ‘Menestras del Negro’, Alborada	28
Ilustración 18: Diagrama de bloques del funcionamiento del semáforo	29
Ilustración 19: Diseño del prototipo de control de tráfico	35
Ilustración 20: Asignación de nomenclatura y designación de pines	37
Ilustración 21: Vista lateral del circuito en etapa de pruebas	38
Ilustración 22: Vista frontal del circuito en etapa de pruebas	38
Ilustración 23: Inicia la secuencia normal con verde1-vd1 y rojo2-rd2	38
Ilustración 24: Verde 1 - verde direccional 1 y rojo 2 - rojo direccional 2	39
Ilustración 25: Continuación de la secuencia amarillo1-vd1 y rojo2-rd2	39
Ilustración 26: Amarillo 1 - verde direccional 1 y rojo 2 - rojo direccional 2	40
Ilustración 27: Continuación de la secuencia rojo1-rd1 y verde2-vd2	40
Ilustración 28: Rojo 1 - rojo direccional 1 y verde 2 - verde direccional 2	41
Ilustración 29: Continuación de la secuencia rojo1-rd1 y amarillo2- vd2	41
Ilustración 30: Rojo 1 - rojo direccional 1 y amarillo 2 - verde direccional 2	42

Ilustración 31: Diagrama de flujo	44
Ilustración 32: Línea de tiempo de los eventos durante la sincronización	45
Ilustración 33: Pruebas de sincronización con diferentes tiempos de pulso	47
Ilustración 34: PCB (Placa de Circuito Impreso).....	49
Ilustración 35: Pistas de la PCB	49

ÍNDICE DE TABLAS

Tabla 1: Costo total de los materiales utilizados en el proyecto.....	33
---	----

CAPÍTULO I

1. INTRODUCCIÓN

El tráfico vehicular en las zonas urbanas alrededor del mundo según [1] es un inconveniente que ha ido en aumento indicando que solo en las dos ciudades más representativas de Ecuador, Guayaquil y Quito, se pierden 167 y 173 horas en congestiones vehiculares respectivamente ubicándose entre las ciudades con peor manejo de tráfico en Latinoamérica ubicándose Guayaquil en el sexto lugar y Quito en el noveno lugar.

Además, se indica en [2] que pasar por muchos problemas de tráfico puede llegar a costar una gran cantidad de dinero, como ha pasado en Los Ángeles, Estados Unidos, lugar en el cual se ha calculado una pérdida aproximada de \$9.700 millones anuales, siendo algunas de las razones la baja de rendimiento de los empleados y el mayor costo de transporte, lo que puede llevar a la quiebra a varios negocios.

Por otra parte, se tiene que este problema de aumento de tráfico vehicular se da debido a que, en los respectivos estudios no se toma en cuenta el sistema de transporte público, y solo se publican resultados del flujo de los automóviles tal y como indica [3] lo cual da lugar a análisis erróneos en varias ciudades tomando en cuenta esos datos publicados generando soluciones que no resuelvan la verdadera fuente del problema.

Por estos motivos mencionados, el presente proyecto propone una solución para sincronizar los semáforos de Ecuador teniendo en cuenta que la mala sincronización de estos también genera un conflicto vehicular en varias avenidas indicadas por [4] en el país.

1.1 Descripción del Problema

Durante el año 2018 en la ciudad de Guayaquil se registró un parque automotor estimado de 484.049 vehículos según un reporte periodístico basándose en datos de la Agencia de Tránsito Municipal de Guayaquil (ATM). Respecto a la cifra del año 2017, se observa un aumento de 62.248 automóviles [5] , es decir, un

incremento anual vehicular del 14.76% en una ciudad cuya población sobrepasa los 2.7 millones de habitantes.

El mismo reporte indica que la ATM entre el 2015 y 2018 ha obtenido un crecimiento promedio anual de 45.000 automóviles, sin embargo, el jefe de Estadística de la ATM, Christian Rochina, señala que podría haber 500.000 [5] carros circulando en la urbe si se consideran los autos provenientes de otras ciudades. Con esta información emitida, se puede tener previsto aproximadamente 664.049 automóviles circulando en la ciudad actualmente.

En otro reporte periodístico realizado en 2019 se observan datos que analizan la movilidad en 200 ciudades de 38 países ubicando a Guayaquil en el puesto 21 [6], mientras que en 2022 Estambul con 62% de nivel de congestión, Moscú con el 61%, Kiev con el 56% y Bogotá con el 55% [7] son las cuatro ciudades con mayor congestión vehicular del mundo.

Estos datos estadísticos revelan la evidente problemática que se presenta en varias ciudades a nivel mundial, como consecuencia, la congestión o tráfico vehicular genera malestar en los conductores ya que en Bogotá pierden 126 horas al año en estancamientos lo que significa el 55% [7] más del tiempo que tardarían en condiciones de recorrido normales.

Actualmente esta situación se refleja aún más con el retorno a clases presenciales post pandemia a partir de 6:30 a 8:00 y 17:00 a 18:30 [8] repartidas en diferentes zonas en Guayaquil.

Tal y como indica [9], en Quito los quiteños pasaban detenidos en semáforos alrededor de 28 horas en 2018, mientras que para el 2020, ese tiempo se triplicó generando gran malestar en sus habitantes. Mientras que, el artículo [10] apunta que, en México, sus ciudadanos pierden de dos a cuatro horas diarias solamente detenidos en semáforos.

Sobre este último caso, se explica que tanto tiempo perdido en semáforos puede llegar a generar pérdidas económicas debido a que varias empresas

públicas y privadas preferirían movilizarse a otros sectores donde no existe tanto congestión vehicular [10].

Mientras tanto, en una ciudad grande y avanzada en tecnología como Los Ángeles, ubicado en Estados Unidos, cada año, un conductor en promedio malgasta 102 horas al año y pierde alrededor del 12% [11] de su tiempo. Los vehículos detenidos en un mismo lugar durante un tiempo prolongado generan un problema ambiental puesto que, empeoran la calidad del aire, siendo el caso de que, un vehículo que se encuentra detenido en el tráfico genera un 40% [12] más de contaminación que uno que se mantenga en movimiento.

1.2 Justificación

Se reduciría el daño al aire generado en las zonas urbanas por los vehículos estancados largos periodos de tiempos en semáforos puesto que, tal y como menciona [12], estos generan una mayor cantidad de contaminación en comparación de cuando estos se encuentran en movimiento.

Adicionalmente, reduciendo la contaminación ambiental, se pueden evitar distintos problemas de salud que algunas personas padecen como el estrés generado al no poder movilizarse de manera continua en las avenidas y teniendo en cuenta su naturaleza impredecible según [13], esto puede convertirse en una conducta agresiva. Otro aspecto a mencionar son los problemas respiratorios que algunas personas pueden sufrir al estar rodeado de aire contaminado, en especial los menores de edad.

Se solucionará el problema de mala sincronización de semáforos en varias partes del país en donde los conductores apenas pueden avanzar algunos metros luego de un largo tiempo de espera en un semáforo solo para encontrarse con una luz roja en la siguiente cuadra causando embotellamientos al tener varias pausas en sus trayectos [4]. Estos casos se pueden ver en Guayaquil, según [4], en el cual este problema mencionado se hace notorio en horas pico en avenidas muy importantes para la movilización de los ciudadanos que llevan por nombre: Juan Tanca Marengo, Rodolfo Baquerizo Nazur y Francisco de Orellana.

El desarrollo de esta solución también tiene como motivo dar una nueva forma de sincronizar semáforos en Ecuador, en donde ya han hecho uso, en algunos sectores del país, de recursos [14] como la fibra óptica y enlaces con una central de control.

El artículo 1 del Código Orgánico del Ambiente detalla que, las personas tienen derecho de coexistir en un ambiente sano sin excesos de contaminación, además el artículo 3 del mismo Código establece en el numeral 7 que se deben buscar métodos para prevenir, reducir y controlar los impactos ambientales.

Finalmente, este trabajo cumple con el objetivo 11 de Desarrollo Sostenible en Ecuador en el cual [15] menciona que este se refiere a ciudades y comunidades sostenibles, teniendo como una de sus metas fijas la disminución del impacto ambiental negativo per cápita de las ciudades.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar e implementar un semáforo que va a sincronizarse por medio de un pulso de reloj externo que permita flexibilizar su integración en un sistema de control de tráfico urbano.

1.3.2 Objetivos Específicos

- Desarrollar un algoritmo capaz de modificar los tiempos de cambios de luces de tal manera que la sincronización se obtenga entre 3 o 4 ciclos del semáforo y leer instrucciones externas.
- Implementar un prototipo de semáforo.

1.4 Estado del Arte

Los semáforos son instrumentos que ayudan en el control de tráfico, ubicándose en las diferentes intersecciones y lugares donde se requiere regular el tránsito vehicular y peatonal, en 1868 se presentó su primer diseño en Londres [16] siendo John Peake Knight el responsable de este trabajo. Con la aparición de

nuevas tecnologías, este instrumento ha ido teniendo varios diseños hasta llegar a la actualidad con lo que se conoce como semáforo inteligente, mejorando el control del tráfico al ejecutar todo por medio de ciertos programas que se ejecutan desde un puesto de mando.

Existen varios estudios para semáforos inteligentes, en un proyecto de semaforización del área urbana [17], los reguladores de tráfico realizan una sincronización con la misma hora y fecha establecida para todos los dispositivos para lo cual emplean módulos GPS.

El uso de la sincronización automática tiene otra forma además del módulo GPS anteriormente mencionado, como es el caso en Bogotá, Colombia, en donde se implementó una central semafórica inteligente la cual sabrá a tiempo real cómo va el tráfico por sí solo, controlará a los semáforos inteligentes, los cuales contendrán sensores y estarán conectados a un medio de comunicaciones a través de fibra óptica [18], y finalmente dará prioridad a aquellas vías que tengan una mayor cantidad de vehículos modificando los ciclos del semáforo hasta que detecte que se tiene un número de vehículos en un nivel normal.

En México, se desarrolló en 2017 un sistema de semáforos inalámbricos el cual menciona que dicho sistema se comunica con los semáforos de un cuadrante el cual tendrá un denominado semáforo maestro y este último podrá comunicarse con los demás semáforos maestros de otros cuadrantes para mandar señales de como el tránsito se encuentra en el momento actual permitiendo crear una red de gestión de tránsito [19] tomando en cuenta que, según lo mencionado en [20], los congestionamientos viales hacen perder de dos a cuatro horas diarias a los ciudadanos.

El artículo [21] explica que, Australia en 2018 ha optado un método muy diferente al explicado el cual fue creado como respuesta a la llegada del auto con guía autónoma al país en 2015 consistiendo en instalar sensores en ciertos carriles para poder recoger datos relativos al tránsito que ayuden en la implementación de algoritmos matemáticos para especificar la duración de los ciclos de los semáforos

para luego hacer uso de la infraestructura de Internet para determinar la ubicación y movimientos de los vehículos con guía autónoma.

Según [22], la mayoría de los sistemas de semáforos actuales utiliza un bucle de cables debajo de las carreteras para crear un campo magnético permitiendo mandar una señal a los vehículos que pasan sobre la carretera y estén conectados al IoT para crear una comunicación entre estos con los semáforos y les informen el tiempo de duración de intersección de carreteras para así reducir los accidentes.

Por su parte, la empresa Ford Motor Company en el año 2022 ha tomado en cuenta la facilitación de rutas para los sistemas de emergencias como las ambulancias, camiones de bomberos y las patrullas, para lo cual ha utilizado una nueva tecnología llamada CV2X (Cellular Vehicle-to-Everything) [23] basada en que un vehículo mande una señal al semáforo especificado para que este se ponga en verde y una vez que el vehículo pase por el cruce, el semáforo volverá a funcionar normalmente siendo esto posible gracias a que dicha tecnología crea una plataforma unificada que conecta a los vehículos beneficiados con la infraestructura de las carreteras, otros vehículos y usuarios de las vías.

Existen también los llamados “semáforos inteligentes anti-ruido” que pueden ser encontrados en Bombay, ciudad ubicada en India [24]. Estos fueron implementados a inicios del 2020, tienen instalados unos medidores de decibelios y cuando el ruido de la avenida supera los 85 dB, lo cual ocurre si la bocina de muchos vehículos es usada a la vez, entonces el semáforo ubicado en la vía pasa automáticamente a rojo durante 90 segundos. Esta medida fue tomando en cuenta que en las ciudades superpobladas es difícil implementar un sistema que sincronice todos los semáforos al haber tantos vehículos con un limitado transporte público.

Se puede usar la inteligencia artificial (IA) que controle los procedimientos de los semáforos inteligentes siendo el caso en la Universidad de Anston ubicado en Birmingham, Inglaterra, donde sus investigadores en 2022 han implementado un software llamado Traffic 3D [25], el cual simula diferentes condiciones de tráfico, y si se equivoca, recibe una penalización hasta que no vuelva a repetir dicho error

por lo que buscará nuevas soluciones adaptándose a los diferentes procesos. Una vez que se observa que su desempeño ha mejorado, se prueba al sistema en una situación real utilizando lo aprendido.

El uso del método mencionado anteriormente en el artículo [25] ha probado ser efectivo siendo que, en 2018, la Universidad Carnegie Mellon realizó experimentos con una IA en la ciudad de Pittsburgh y sus resultados demostraron que hubo una reducción del 40% en el tiempo de espera de los vehículos y otras reducciones de emisiones de hasta el 26% [26] en los vehículos que pasaban por las calles gestionadas por su sistema.

Los peatones también forman parte de posibles congestionamientos de vehículos por lo que, en 2019, unos investigadores de la Universidad Tecnológica de Graz ubicado en Austria [27], desarrollaron unos semáforos inteligentes los cuales están suministrados mediante una cámara que ayuda a localizar a los transeúntes que se dirigen a la vía para luego dependiendo de la cantidad de personas que localiza, el sistema instalado en un ordenador que se encuentra incorporado en la señal de tráfico, mandar una señal al semáforo para señalarle si deja pasar a los vehículos o a los peatones.

Por otra parte, en una revista de Cuba durante el año 2020 analizan la sincronización de semáforos utilizando el programa Synchro [28] para lo cual coordinan las cadenas semaforicas tomando parámetros generales como el tiempo de ciclo, intervalos de verde, desfase y ancho de banda, de modo que se establecen variantes de sincronización con un ciclo natural de 100 y 94 segundos para velocidades de 80 y 70 Km/h respectivamente, contrastando en el ámbito nacional en El Triunfo [29] realizan el mismo análisis y simulación a través del software Synchro que se realizó en [28].

Pero se encuentra otra perspectiva y muy diferente en Alemania [30] donde se indica que esta tecnología se basa en la implementación de sensores de radar y cámaras de alta resolución con el fin de captar con mayor exactitud la situación real del tráfico vehicular, permitiendo obtener el número exacto de los vehículos que esperan en los cruces, estos datos se almacenan en la nube y con la ayuda de

algoritmos de aprendizaje calculan los periodos de funcionamiento de los semáforos.

Un proceso similar ocurre en [31] que analizan una comparativa de la semaforización tradicional contra la semaforización inteligente para la reducción del congestionamiento vehicular, emplean una tarjeta Arduino y una cadena de sensores a lo largo de la calle cuyo funcionamiento de sistema es detectar la mayor congestión vehicular precediendo a disminuir el tiempo de cambio del semáforo para agilizar la movilidad de una calle mientras que la otra calle tiene que aumentar el tiempo de cambio.

En otro escenario muy semejante al [31] se presenta en México [32], desarrollan un Sistema centralizado de sincronización de semáforos inalámbricos solares (SICESSIS), la diferencia radica que en este proyecto emplean los paneles solares como fuentes de alimentación de cada uno de los sistemas maestros ubicados en cada cruce y a su vez cada uno de ellos cuenta con un GPS (Sistema de Posicionamiento Global) para la ubicación y sincronización semafórica.

Mientras que la tesis [33] trabaja con un arquetipo de controlador de semáforos leds, el cual realiza el mismo trabajo que [31] pero con mayor cantidad de procesadores puesto que contiene tres sistemas, comunicación, control y GPS, que le permiten controlar el sistema completo, a través del sistema de comunicación se envían las tareas a realizar.

Aplicando una metodología diferente a las ya mencionadas, en la ciudad de Machala aparece el método de Webster [34] cuyo funcionamiento del semáforo se basa en la ejecución de planes constituidos por una serie de fases que tienen una duración determinada ejecutándose en un horario establecido, este método considera la acción del vehículo, así como también la presencia de peatones.

Una implementación en México resulta ser diferente a las ya conocidas en cuanto a dispositivos empleados ya que también considera el control de tráfico vehicular según la intersección con el mayor número de vehículos para la sincronización de los semáforos, pero lo que lo hace diferente es una cámara [35]

que viene con el dispositivo de control de tránsito encargara de la visualización y captura del flujo vehicular.

La solución escogida es mediante un semáforo que va a sincronizarse por medio de un pulso de reloj externo que permita flexibilizar su integración en un sistema que tenga como función el control de tráfico en distintas zonas urbanas, para lo cual se tendrá como fin la modificación de los tiempos de cambio de luces, así como también la lectura de instrucciones externas.

Esta solución se diferencia del estado del arte ya que los métodos utilizados para la sincronización de los semáforos se han basado en el empleo de módulos GPS, en la creación de aplicaciones móviles, en inteligencias artificiales con paneles solares como fuente de alimentación, pero no ha presentado la implementación de un semáforo que se sincronice por medio de pulsos de reloj externos, lo que significaría una nueva propuesta de solución para evitar los estancamientos que ocurren en las vías urbanas, teniendo como beneficio el costo de su implementación debido a que es económica.

1.5 Alcance

El mecanismo de solución que se propone consiste en la implementación de un semáforo que recibe instrucciones y un pulso de sincronización siendo ambos externos. En la práctica, estas instrucciones podrían ser recibidas mediante un sistema de comunicaciones inalámbrico.

Este trabajo requiere dispositivos como microprocesadores para el diseño y configuración de un circuito integrado (CI) con el objetivo de desarrollar un dispositivo a través del cual se pueda sincronizar los dispositivos de señalización de tránsito de manera que la circulación vehicular fluya dentro del área urbana.

En términos generales la solución planteada se describe en tres fases. La primera fase consiste en la alimentación, que se encarga de brindar la energía suficiente para el correcto funcionamiento del sistema.

La segunda fase corresponde al algoritmo que controla el funcionamiento del semáforo, es decir, tiene como objetivo regular los ciclos de activación de las luces para gestionar el tráfico vehicular.

Como tercera fase se presenta la sincronización del semáforo cuyo objetivo es su inicialización periódica a través de un pulso de reloj externo después de un determinado tiempo y a partir de un punto de referencia.

1.6 Metodología

El proyecto tendrá como enfoque a las zonas urbanas del país siendo que, estas zonas son aquellas donde se requiere mayormente una mejor sincronización de los semáforos. Por dicho motivo, se ha optado por el diseño e implementación de un semáforo que, a través de un pulso de reloj externo, pueda sincronizar tiempos con cualquier otro similar teniendo en cuenta que, el cambio deberá ser progresivo de tal manera que permita flexibilizar su integración en un sistema que controle el tráfico en diversas zonas urbanas.

Es por ello que se requiere implementar un algoritmo capaz de modificar los tiempos de cambio de las luces del semáforo de tal modo que el proceso de sincronización se realice en un intervalo de 3 o 4 ciclos y que pueda leer instrucciones externas, por lo que, se necesitará el empleo de microprocesadores. Teniendo este concepto en cuenta, se decidió usar un dispositivo que esté compuesto por microcontroladores como lo es el ESP 32 o el Arduino UNO, donde la selección final dependerá de los objetivos que se busquen cubrir.

Debido a que el algoritmo a desarrollar presenta varios requerimientos se ha considerado el lenguaje de programación que mejor defina las especificaciones de codificación, por lo que se ha tomado en cuenta a herramientas como Arduino, Python e inclusive Quartus por medio de códigos VHDL como posibles opciones.

En este punto del proyecto con el código de programación desarrollado y con el circuito diseñado se procede con la culminación con la implementación de un semáforo.

CAPÍTULO II

2. MARCO TEÓRICO

En esta sección se explican los conceptos relacionados a distintas herramientas como lo es el módulo ESP 32, el reloj a tiempo real o RTC que ayuda a realizar seguimientos de eventos que ocurren en distintos momentos, los microprocesadores puesto que, el ESP 32 está conformado por dichos dispositivos, también se habla de algunos lenguajes de programación que ayudan a realizar distintas funciones según sus características e indicaciones y finalmente se mencionan las diferentes características de varias herramientas como lo son las luces LED, resistores y la placa protoboard.

2.1 Microprocesadores

Los microprocesadores se pueden definir como la combinación entre una computadora y su respectiva tecnología semiconductora empezando su progreso a mediados de la década de los 50 hasta que las tecnologías anteriormente mencionadas se fusionaron a inicios de la década de los años 70.

Ambas tecnologías empezaron sus respectivos desarrollos mientras transcurría la Segunda Guerra Mundial cuando la computadora era usada específicamente para acciones militares para luego de dicho suceso ser modificada para tener una orientación a proyectos científicos y civiles [36]. Poco después que el transistor fuera creado en 1948, las primeras computadoras dactilares con un propósito general se manifestaron las cuales utilizaban tubos al vacío que se manejaban como elementos electrónicos activos.

El problema era que el uso de dichas computadoras exigía una gran cantidad de dispositivos electrónicos y muchos circuitos siendo que las calculadoras electrónicas de ese tiempo requerían de al menos 75 hasta un máximo de 100 circuitos integrados. Con el objetivo de que la arquitectura de las mencionadas computadoras se reduzca a un chip mucho más simplificado [36], se creó un circuito conocido como microprocesador el cual puede manejar una serie de datos binarios

de diferentes maneras, así como la capacidad de, a partir de un programa preestablecido, realizar un gran número de operaciones lógicas.

2.1.1 Microprocesador Pentium

Este microprocesador presentado en la Ilustración 1 trabaja con una velocidad de hasta 66 MHz conteniendo 3.1 millones de transistores, tiene hasta 8KB de caché interno destinado para datos y 8KiB dirigido a instrucciones. No fue llamado 586 por Intel porque no se puede tener como registrado a una marca que se componga solo por números [37]. Puede alcanzar velocidades de buses de 50, 60 y 66 MHz según sus versiones, su velocidad de reloj puede variar desde 166 MHz hasta 233 MHz brindando una mejor experiencia en la lectura de películas en DVD a comparación de sus antecesores.



Ilustración 1: Microprocesador Intel Pentium [38]

2.1.2 Xeon Dual Core

Estos microprocesadores están diseñados para aplicaciones de estaciones de trabajo y servidores que trabajen con dos procesadores de alto rendimiento, cuya presentación se muestra en la Ilustración 2. Son escalables para dos procesadores lo cual significa que brinda un gran rendimiento para aplicaciones que corren en sistemas operativos avanzados como Windows XP, Windows Server 2003, Linux y UNIX [39]. Puede alcanzar velocidades de buses desde los 667 MHz hasta los 1066 MHz dependiendo su versión, la frecuencia con la que trabaja va desde los 2.5 GHz hasta los 3.2 GHz además que trabaja con voltajes entre 1.075 voltios a 1.35 voltios.



Ilustración 2: Microprocesador Intel Xeon [40]

2.1.3 Microprocesador Intel Atom

Este dispositivo puede ejecutar hasta dos instrucciones por ciclo y dispone de un amplio rango de núcleos que permiten diferentes tipos de procedimientos que se le indique [41]. Puede tener desde solo 4 núcleos hasta 24 en un solo microprocesador, además viene con una LAN integrada que contiene hasta ocho puertos Ethernet con una capacidad mínima de 50 Gb, siendo que, su versión más actualizada también brinda un conmutador integrado que permite realizar el cifrado en línea reduciendo en gran medida la latencia al momento de implementar la seguridad en redes, este microprocesador se presenta en la Ilustración 3.



Ilustración 3: Microprocesador Intel Atom [42]

2.1.4 Intel Core

Fue el reemplazo de los modelos Pentium como el estándar principal para chips de gama alta. Existen varios modelos como lo son el Core i3 que se puede observar en la Ilustración 4, el Core i5 presentado en la Ilustración 5, el Core i7 mostrado en la Ilustración 6 y el Core i9 capturado en la Ilustración 7, siendo que

cada una nueva generación es desarrollada en un intervalo de dos y cada una permite nuevas funciones.

2.1.4.1 Intel Core i3

El i3 es la gama de baja de esta familia de microprocesadores permitiendo realizar tareas simples como revisar el correo electrónico, navegar a través del Internet y compartir archivos [43]. Contiene chips de hasta 6 núcleos, puede trabajar hasta una frecuencia máxima de 4.7 GHz y tener 12 MB de caché.



Ilustración 4: Microprocesador Intel Core i3 [44]

2.1.4.2 Intel Core i5

El i5 es la gama media permitiendo actividades como streaming de música y película, actividades de ofimática, así como reproducir gráficos en 4K para juegos y vídeos [43]. Esta generación contiene chips de hasta 8 núcleos con la capacidad de trabajar con una frecuencia máxima de 5 GHz y una memoria caché de 16 MB.



Ilustración 5: Microprocesador Intel i5 [45]

2.1.4.3 Intel Core i7

El i7 es considerada la gama alta de los Core que permiten realizar actividades de mayor peso como los videojuegos, programación de software, edición de imágenes y garantiza el máximo desempeño en cada una de las tareas que se le asigne [43]. Contiene chips de hasta 8 núcleos con la capacidad de trabajar a una frecuencia máxima de 5.3 GHz y 16 MB de memoria caché.



Ilustración 6: Microprocesador Intel Core i7 [38]

2.1.4.4 Intel Core i9

El i9 es la gama de mayor rendimiento de los Core que permite realizar actividades de gran complejidad como las creaciones de contenido en 3D, videojuegos más recientes que requieren una mayor capacidad de rendimiento. Además, contiene controles de overlocking permitiendo un mejor desempeño en las actividades que tenga designadas [43]. Esta generación contiene dos

tipos de núcleos: 4 núcleos portátiles y 8 núcleos de escritorio con la capacidad de trabajar a una frecuencia máxima de 5 GHz ofreciendo también la tecnología Wi-Fi 6 para tener un mejor rendimiento que las computadoras livianas.



Ilustración 7: Microprocesador Intel i9 [46]

2.2 Microcontroladores

Son un sistema de control integrado basado en un microprocesador en un solo chip de bajo costo que es capaz de realizar diferentes instrucciones que se le indique [47], siendo que, estas acciones dependerán del software con el cual se esté trabajando.

A diferencia de una CPU, estos dispositivos funcionan a velocidades de reloj más bajas debido al hecho que, no utilizan una memoria RAM, sino una memoria programable permitiendo además que, se tenga un consumo mucho menor a nivel energético [47]. Para acceder a su memoria que es una de tipo flash, es necesario acceder mediante los pines de entrada y salida del microcontrolador.

Los microcontroladores pueden ser programados utilizando diferentes lenguajes de programación como lo son C y C++, también utilizan cadenas de herramientas de compilación tradicionales [48].

2.2.1 Arduino

La tarjeta Arduino se puede considerar como una plataforma que sirve para la creación de prototipos basada en Hardware y Software libre, por lo que,

cualquier usuario es capaz de realizar cambios en los códigos realizados [48] y cuya tarjeta de trabajo se muestra en la Ilustración 8. Los diseños de estas tarjetas utilizan varios tipos de microprocesadores y controladores, además se encuentran equipadas por un conjunto de pins de entrada/salida digitales y analógicos que pueden ser conectados a otros dispositivos o circuitos.

Puesto que, su lenguaje está basado en frameworks, Arduino permite la interacción con otros programas en el ordenados, como por ejemplo Flash o hasta con APIs web como Twitter. [48]



Ilustración 8: Arduino UNO [49]

2.2.1.1 Principales Ventajas

- Es fácil de usar al utilizar una versión simplificada del lenguaje C y C++.
- Bajo costo, debido a que, su herramienta de desarrollo IDE es totalmente gratuita, siendo que se puede conseguir uno en \$7.
- Al ser de código abierto, se puede conseguir ayuda de personas con mayor conocimiento en programación para editar los errores de códigos detectados.
- Ofrece soporte multiplataforma, siendo que puede trabajar con Windows, Linux y macOS.

2.2.1.2 Principales Desventajas

- Solo pueden ejecutar un solo programa a la vez necesitando herramientas externas para lograr esta acción.

- No está hecho para trabajos muy completos debido a que, sus microcontroladores no están hechos para trabajar al máximo de su rendimiento al ser una placa hecha para principiantes.
- No tienen soporte de comunicaciones incorporado, es decir, para que tengan conexión a Wi-Fi o Bluetooth, se deben interconectar módulos externos de hardware para tener dichas funciones disponibles.
- Carece de soporte para lenguajes como Java, Python y JavaScript.
- Tienen un almacenamiento de memoria limitado siendo que, tiene apenas hasta 32 Kb de memoria flash.

2.2.2 Raspberry Pi

Es un dispositivo que fue diseñado por la Fundación Raspberry Pi, pero sus modelos posteriores fueron creados por Sony UK. Es considerado como una minicomputadora que tiene como principal característica el disponer de interoperabilidad con cualquier dispositivo de hardware de entrada y salida como un monitor, televisor o teclado [50]. Una de sus ventajas es que es de bajo costo, además de poder usar lenguajes de programación que ayudan a los principiantes a aprender de manera rápida y sencilla sobre la programación como lo son Python y Scratch, este dispositivo se visualiza en la Ilustración 9.

Para poder empezar a configurar con el Raspberry Pi, es necesario una tarjeta SD que permita el arranque del sistema [50]. Un detalle para destacar es que este sistema es compatible con el sistema operativo Linux, reduciendo la cantidad de memoria necesaria para ejecutar las instrucciones indicadas y dándole una mayor diversidad en la cantidad de acciones que se puede realizar con este dispositivo.

Dependiendo del modelo, puede trabajar a frecuencias de reloj desde 700 MHz hasta 1500 MHz, una memoria RAM desde 512 MB hasta 1 GB, tener la opción de poder conectarse mediante Bluetooth o Wif-fi a otros dispositivos y acceso a un puerto Ethernet [51].

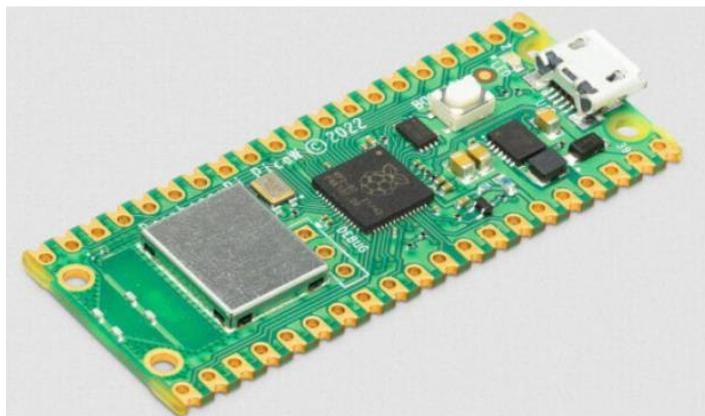


Ilustración 9: Raspberry Pi [51]

2.2.3 ESP 8266

Fue diseñado por una empresa china llamada Espressif Systems, siendo conocido como una forma de permitir que los microcontroladores puedan conectarse de manera inalámbrica a las redes Wi-Fi, además de realizar conexiones simples usando TCP/IP [52]. Este módulo apareció debido a la vulnerabilidad económica presentada en el Arduino MKR1000 que estaba en el mercado el cual representaba un gasto económico que muy pocos grupos podían sustentar, dando lugar a su primera versión llamada ESP01.

En general, el ESP 8266 es solo el procesador y la forma de construirlo sobre una placa impresa variará dependiendo de las versiones con las que se trabaje [52]. Este procesador mostrado en la Ilustración 10 trabaja con 80 MHz por defecto, pero le es posible llegar hasta los 160 MHz, además contiene 80 de kB de Data RAM, tiene 17 pines de entrada y/o salida los cuales como entrada pueden proporcionar una tensión desde 0 voltios hasta 3.3 voltios, en cambio como salida, pueden proporcionar hasta 12 voltios.

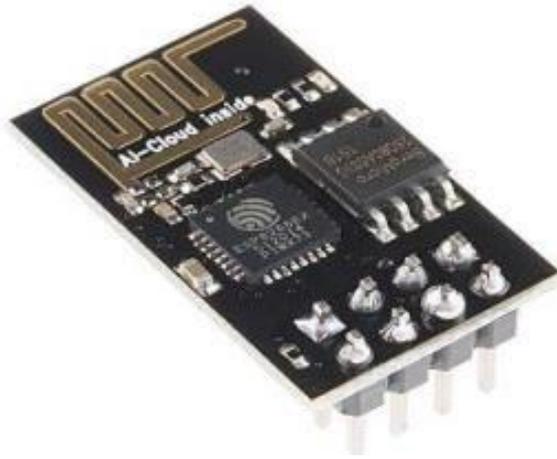


Ilustración 10: Módulo ESP 8266 [53]

2.2.4 ESP 32

Al igual que su versión anterior, este procesador también fue diseñado por la empresa china Espressif Systems [54]. Contiene un doble núcleo de 32 bits que le permite trabajar desde 160 MHz hasta 240 MHz, una memoria flash externa de hasta 16 MB, una memoria de data SRAM de 520 kB, 32 pines de entrada y/o salida, una memoria ROM de 448 kB, un convertidor analógico digital con capacidad de 12 bits que incluyen además 18 canales y 2 convertidores digital analógico con un almacenamiento de 8 bits. Así mismo, soporta velocidades de hasta 150 Mbps y una potencia de salida que alcanza un valor máximo de 20.5 dBm para garantizar el mayor rango físico posible.

En la Ilustración 11 se presenta este procesador que se caracteriza por tener integrado Bluetooth, Bluetooth LE y Wi-Fi lo cual garantiza una mayor capacidad de aplicaciones como puede ser conectarse a un dispositivo móvil mediante Bluetooth o tener físicamente un gran alcance y conexión de manera directa a Internet mediante Wi-fi por medio de un router [54]. Además, al tener dos núcleos, se podría hacer un proyecto que intercambie información a la nube y de manera simultánea esté administrando datos de un sensor con gran precisión.

Este módulo se considera como la versión mejorada del ESP 8266 al tener una mayor velocidad de wi-fi, una mayor cantidad de pines de entrada y salida, así mismo, el ESP 32 tiene integrado Bluetooth, algo que el ESP 8266 carece. También cabe recalcar que, el ESP 32 tiene una memoria ROM programable y

puede funcionar como un sensor de temperatura [55]. La única desventaja sería que el último módulo tiene un mayor precio económico, por lo tanto, si se trata de realizar un proyecto sin mucha complejidad, se podría pensar en usar el ESP 8266 para un menor gasto.

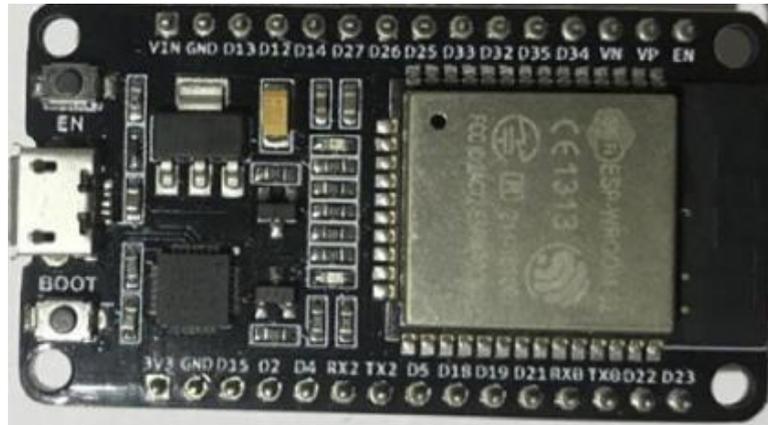


Ilustración 11: Módulo ESP 32 [56]

2.3 Lenguajes de Programación

Son un conjunto de instrucciones escritas que pueden estar en cualquier lenguaje disponible (como Python, C++ o Arduino) con el objetivo de poder realizar alguna tarea en específico [57]. Se componen de un conjunto de reglas semánticas que permiten expresar dichas instrucciones para luego ser interpretadas, y tienen como principal uso el desarrollo de aplicaciones móviles o de escritorio, sitios web o comunicación con dispositivos hardware y software que se encuentren presentes.

Cada uno de esos lenguajes tienen diferentes normas, especificaciones y límites que deben ser cumplidos para poder realizar las acciones que se les especifique teniendo en cuenta que cada una tiene sus diferentes ventajas y desventajas.

2.3.1 Python

Este lenguaje se usa para comúnmente programar la creación de sitios web, automatización de distintas tareas y análisis de datos [58]. Esto es posible debido a que, Python es un lenguaje de propósito general, por lo tanto, no está dirigido para un tipo de problema en específico.

Algunas de sus ventajas es el hecho de que es fácil para el programador encontrar errores en su código porque este programa ejecuta dicho código línea por línea, por lo tanto, es muy adoptado por principiantes en la programación [57]. Además, su uso es gratuito, tiene una sintaxis simple imitando el lenguaje natural siendo fácil de entender y tiene una comunidad grande y activa, por lo que se mantiene en constante actualización de sus librerías que le brindan nuevas funciones.

Sin embargo, no es muy recomendable para el desarrollo de aplicaciones y juegos móviles ni funciones que deban trabajar de manera continua debido a que, no es un lenguaje muy rápido al momento de ejecutar sus códigos al encontrarse trabajando con el intérprete [57].

2.3.2 C++

Es una extensión del lenguaje C con el objetivo de manipular objetos, teniendo una programación estructurada y orientada a ellos, siendo conocido como un lenguaje de programación multiparadigma [59]. A diferencia de Python que es un lenguaje interpretado, este lenguaje de programación es de un tipo estático, es decir, que todas las variables declaradas, así como la detección de errores serán verificados en el instante de la compilación de datos.

Algunos de sus beneficios es el hecho de ser portátil, por lo que puede ser adaptado en múltiples plataformas, así mismo, su lenguaje está bastante relacionado con el lenguaje C y Java, por lo que facilita a los programadores el cambio entre estos lenguajes en caso de ser necesario [59]. También se debe tener en cuenta su alto rendimiento debido al hecho que realiza llamadas directas al sistema operativo del dispositivo y se mantiene actualizado al contener una comunidad muy activa.

No obstante, tiene ciertas desventajas como lo es la enorme amplitud de su lenguaje siendo necesario muchas líneas de código para ejecutar ciertas acciones teniendo una curva de aprendizaje bastante alta [57], además de la necesidad de tener una compilación para cada plataforma, por lo que es difícil la depuración de su código en aplicaciones complejas.

2.3.3 Arduino

Además de ser una placa que se encuentra fundamentado en un microcontrolador, es también un conocido lenguaje de programación basado en el lenguaje C++ siendo muy similar a este [60], con la diferencia de que Arduino ofrece librerías para facilitar la programación de los pines que funcionan como entrada y salida de los distintos puertos de comunicación, así mismo, su plataforma está diseñada para no aceptar todas las funciones del lenguaje C++.

La principal diferencia con el lenguaje anteriormente mencionado es que todo el código se divide en mínimo 2 funciones principales [60] siendo estos:

- `setup()`._ Esta función es llamada cuando se comienza a programar.
- `loop()`._ Esta función es llamada de manera continua mientras el programa se encuentra corriendo el código.

2.4 Placa Protoboard

Llamada también placa de pruebas se observa en la Ilustración 12, es una herramienta que permite utilizar diferentes dispositivos sin la necesidad de soldar haciéndolo ideal para que los principiantes empiecen a familiarizarse con los circuitos e incluso los expertos en el tema hacen uso de esta placa porque en ocasiones realizan pruebas para iniciar proyectos a gran escala [61]. Vienen en varios tamaños y sus diseños pueden variar, pero su funcionamiento continúa siendo el mismo.

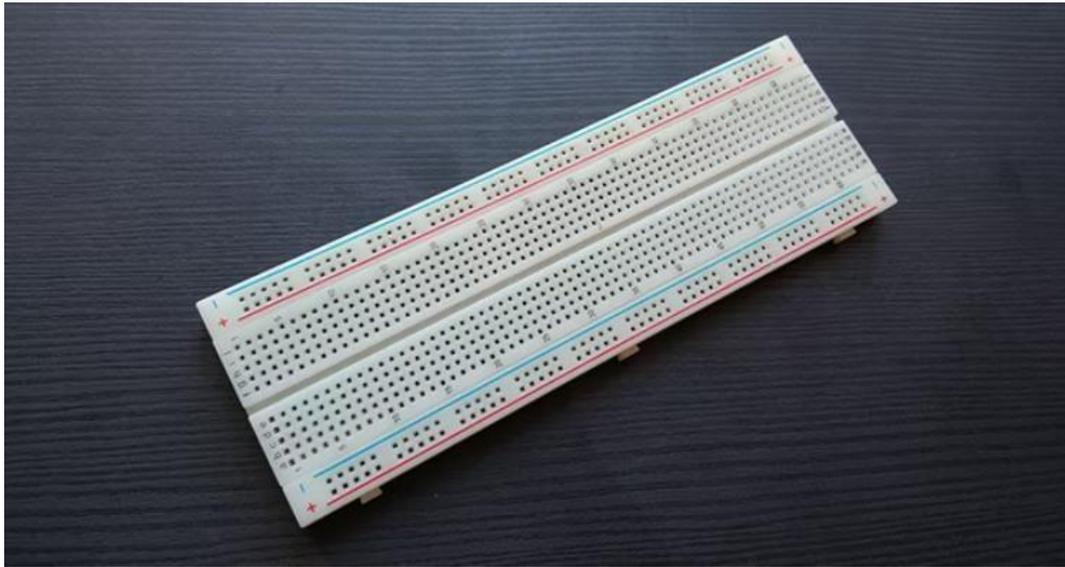


Ilustración 12: Placa Protoboard [61]

2.5 Reloj de Tiempo Real o RTC

Este dispositivo es un reloj que realiza un seguimiento preciso del tiempo incluso si la fuente de alimentación se encuentra apagada. Se encuentra compuesto por un controlador, un oscilador y un resonador de cuarzo incorporado [62]. Por lo mencionado anteriormente, el RTC es mayormente utilizado en aplicaciones en las cuales tienen un rol importante en el seguimiento preciso del tiempo actual, además proporciona la posibilidad de incorporar alarmas, temporizadores, funciones de interrupción y ayudan a reducir el consumo de energía del circuito con el que se encuentre trabajando, se lo puede observar en la Ilustración 13.

Todo lo mencionado es posible debido a que, el RTC tiene una función de interruptor de encendido para el funcionamiento de la fuente de alimentación permitiendo que el dispositivo tenga un conteo de tiempo preciso y continuo incluso si el circuito se encuentra apagado o en modo suspensión [62]. Todos los registros o acciones que realiza el dispositivo quedan guardados en su memoria RAM y se actualizan de manera periódica.

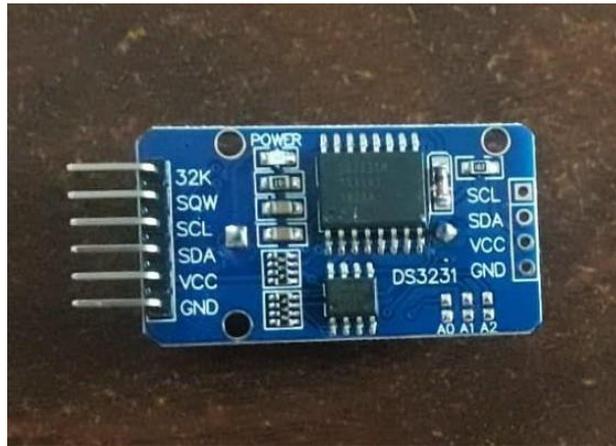


Ilustración 13: Dispositivo RTC

2.6 Resistores

Estos componentes se aprecian en la Ilustración 14 y son considerados de tipo eléctrico pasivo que permiten crear resistencia en el flujo de corriente eléctrica presente en el circuito diseñado. Tienen diferentes usos como limitar la cantidad de corriente que le llegan a otras herramientas en el circuito, generación de calor, división de voltaje y ajuste de tiempos constantes [63]. Las propiedades y coeficientes de las resistencias pueden variar dependiendo de la necesidad que tiene la persona en el proyecto con el que se encuentra trabajando.

Además, contienen un código de color debido al hecho que los modelos de estos componentes tienen varias bandas de colores que indican las diferentes características con las que trabaja el modelo adquirido [63], como lo puede ser la tolerancia o los dígitos significativos del valor de la resistencia.

Se las puede conectar en serie o en paralelo dependiendo del uso que se las quiera dar en el circuito especificado tomando en cuenta que, conectadas en serie, la corriente en cada resistencia es la misma, mientras que, en paralelo, es el voltaje el que contendrá el mismo valor en cada una de estas componentes [63].



Ilustración 14: Resistores [64]

2.7 Luces LED

Los LED que se presentan en la Ilustración 15 son dispositivos semiconductores que capaces de producir luz cuando existe corriente que fluya a través de ellos [65]. Normalmente son de un tamaño muy pequeño y el color que emiten se encuentra definido por la brecha de banda de energía de su respectivo semiconductor.

Estos dispositivos tienen la ventaja de consumir poca energía, no requieren mucho mantenimiento, pueden trabajar en varios rangos de voltaje, además que no están sujetos a fallos repentinos o quemados debido a que, no hay filamentos que puedan ser quemados o rotos [65]. Sin embargo, sus rendimientos dependen mucho del ambiente que los rodea porque al estar rodeados de un ambiente muy caliente, esto pueden llegar a sobrecalentarse necesitando un disipador de calor para asegurar un largo tiempo de vida en dichas circunstancias.



Ilustración 15: Diodos Led [66]

2.8 Jumpers

También conocidos como cables puentes, son un componente que posibilitan la apertura o cierre de un circuito eléctrico mediante sus terminales, se puede observarlos en la Ilustración 16. Estos son usados cuando dichos terminales no están adyacentes o se encuentran muy alejados de otros componentes del circuito [67]. Tienen la ventaja de que, son rápidos de conectar y desconectar porque se fijan solamente con la inserción de sus extremos permitiendo la coincidencia del pin macho con el pin hembra.

Existen diferentes tipos que se clasifican dependiendo de una característica específica:

Según su fisionomía:

- Hembra-hembra
- Macho-hembra
- Macho-macho

Según su conexión:

- Versiones con pinzas de cocodrilo en sus puntas
- Versiones con terminales aislados



Ilustración 16: Jumpers macho-hembra [67]

CAPÍTULO III

3. ESCENARIO

Este proyecto se basa en los ciclos de trabajo de los semáforos que se encuentran en el sector del restaurante 'Menestras del Negro' de la Alborada como se puede observar en la Ilustración 17. Se eligió esta ubicación porque es una carretera que no contiene una sola vía, sino que, por el contrario, tiene direccionales que indican a los vehículos por cual ruta pueden avanzar dependiendo el color que les muestren. En este caso, se tienen 3 posibles rutas por la cual los vehículos podrán transitar y los tiempos de los ciclos de cada color varía según la hora del día.



Ilustración 17: Semáforo ubicado en el sector de 'Menestras del Negro', Alborada

3.1 Diseño de la Propuesta

Tomando en cuenta los aspectos mencionados anteriormente respecto a los problemas que se generan al no tener semáforos en el sector correctamente sincronizados, se propone como solución un semáforo que, mediante pulsos externos, se irá sincronizando de manera progresiva, con el objetivo de agilizar el flujo de tráfico evitando cambios bruscos en sus ciclos para evitar accidentes, así como proporcionarle un código mediante el lenguaje de programación Arduino que ayude en su funcionamiento.

Esta solución contiene varios pasos a completar antes de llegar a una correcta sincronización tal y como se muestra en la Ilustración 18.

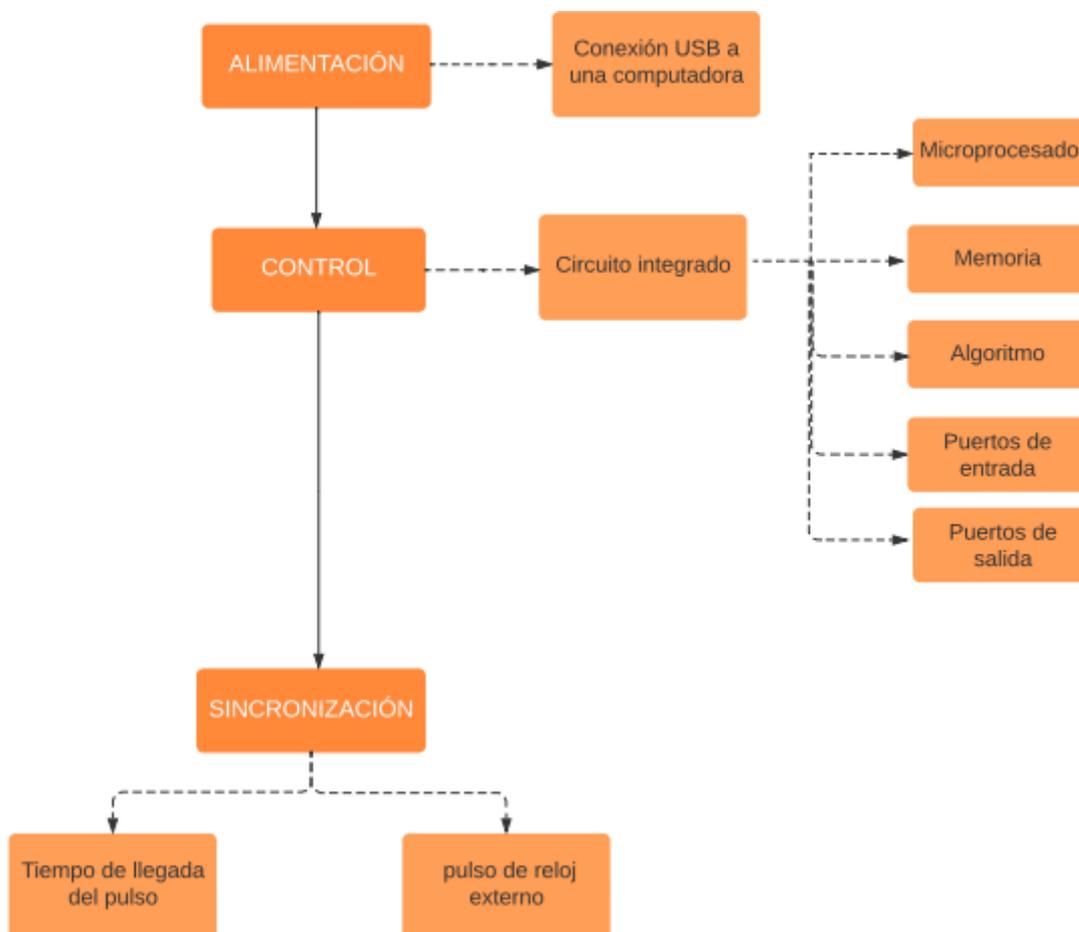


Ilustración 18: Diagrama de bloques del funcionamiento del semáforo

3.1.1 Proceso de Alimentación

En la etapa de alimentación para el diseño de la solución propuesta se puede realizar de tres formas diferentes, la primera es a través de un tomacorriente con un convertidor de voltaje, la segunda es por medio de baterías que según sus componentes de fabricación pueden convertirse en una muy buena fuente de energía, y finalmente la forma más habitual actualmente es a través de un puerto USB. Sin embargo, en este proyecto se ha decidido utilizar la tercera opción que es la alimentación por USB debido a la facilidad que presentan las tarjetas de desarrollo disponibles en el mercado electrónico.

3.1.2 Proceso de Control

La segunda etapa corresponde al proceso de control, el cual depende de un circuito integrado cuyas características necesarias comprenden la presencia de un microprocesador, una unidad de memoria con la capacidad suficiente para poder almacenar el algoritmo del funcionamiento de la solución propuesta, puertos de entrada orientados a la recepción de información y puertos de salida que puedan ser asignados a los dispositivos de acción que en este caso corresponde a las luces led.

3.1.3 Proceso de Sincronización

La última etapa corresponde al proceso de la sincronización del dispositivo, para que esto suceda es necesario la existencia de un pulso de reloj externo e independiente del circuito integrado encargado de la etapa anterior.

En primera instancia el dispositivo operará bajo la condición de la nula existencia de un pulso de reloj externo de sincronismo por lo que se trabajará en condiciones normales bajo una secuencia normal, en segunda instancia el dispositivo operará bajo la condición de la existencia de un pulso de reloj externo de sincronismo, en este punto la aplicación del algoritmo cumple un papel fundamental.

El algoritmo se encarga de registrar el tiempo en que llega el pulso de sincronismo, este valor es dividido para 4, que es el número de ciclos en que se alcanzará la sincronización, este tiempo luego es repartido entre el tiempo de duración de encendido del verde y amarillo, de modo que al cabo de cuatro ciclos el punto de inicio de semáforo se habrá trasladado al punto en que recibió el pulso y logrando con ello el objetivo del proyecto que es la sincronización a partir de un pulso externo.

3.2 Materiales Utilizados

3.2.1 Hardware

- Protoboard

Es la placa sobre la cual se ha trabajado el proyecto siendo necesaria debido a sus orificios que se encuentran ordenados por filas que posibilitan el ingreso de varios cables y otros elementos electrónicos permitiendo la creación de un circuito sin la necesidad de soldar los componentes que se han adquirido, además de que es un proyecto que no hace uso de una gran cantidad de corriente.

- ESP32

Se escogió el módulo ESP 32 por sobre cualquiera de los modelos de Arduino debido a que, este contiene una memoria mucho mayor que le permite trabajar por períodos más largos de tiempo, y siendo que este proyecto trabaja de manera continua y automática, es necesario tener la mayor cantidad de memoria posible. Además, el ESP 32 tiene en su interior un microprocesador que trabaja a una mayor velocidad de reloj siendo una característica importante en un proyecto en el cual se trabaja con los ciclos de trabajo de un dispositivo.

Así mismo, este trabajo no utiliza sistemas operativos como Linux, o hace uso de una tarjeta SD, por lo que un Raspberry Pi el cual es más una minicomputadora con funciones distintas a una ESP 32 no funcionaría de manera efectiva ya que, la mayoría de las funciones están dirigidas a ser controladas por un microcontrolador, que, en este caso, se encuentra dentro del ESP 32.

Por el lado del ESP 8266, cabe recalcar que el ESP 32 es su versión mejorada; tiene todas las características del primer dispositivo mencionado, además de varias mejoras. Solo se haría uso del ESP 8266 si se busca realizar un proyecto más simple y si se tiene como objetivo reducir el gasto.

- Reloj a tiempo real o RTC

Este dispositivo se caracteriza por entregar la fecha y hora exacta en caso de ser requerido siempre y cuando se utilice una batería como fuente de alimentación. Es el encargado de enviar los pulsos con los que el controlador de tráfico trabaja para poder lograr la sincronización.

- Luces LED

El uso que tienen es para simular los diferentes colores e indicaciones que el semáforo contiene. Para este caso, se están utilizando los leds de color rojo, amarillo y verde, siendo estos los colores clásicos de un semáforo normal, y uno de color azul para indicar que la acción que se está realizando ha sido concretada de manera exitosa.

- Jumpers

Son los elementos encargados de crear el cableado en el circuito del semáforo y sirven como puentes para la conexión entre los pines de la protoboard y los elementos que se están utilizando para trabajar como el ESP 32 siendo necesarios porque esta parte del proyecto no puede ser realizado por medio del software.

- Resistores

Son los encargados de limitar la cantidad de corriente a los distintos dispositivos presentes en el circuito, por ejemplo, si a los leds les llega demasiada corriente al estar conectados de manera directa a uno de los pines del ESP 32, pueden llegar a quemarse si se los expone por un tiempo prolongado y al necesitar este proyecto de muchas pruebas para encontrar posibles errores, es necesaria la presencia de estos resistores para prolongar la vida útil de los leds.

3.2.2 Software

- Arduino

Se ha escogido el lenguaje de programación Arduino debido a las facilidades y funciones que este ofrece en comparación con Python, siendo que este último es más utilizado para proyectos de tipo software, a diferencia del presente trabajo que utiliza varias herramientas de tipo hardware. Adicionalmente, Python no es recomendado para proyectos que trabajen de manera continua, esto

debido al hecho de que no es un lenguaje que se destaque por ser muy rápido al momento de correr sus códigos.

Por el lado del lenguaje C++, cabe mencionar que el lenguaje Arduino está basado en dicho lenguaje, solo con el detalle de que tiene ciertas funciones agregadas para trabajar con componentes de tipo hardware. El ESP 32 tiene como característica el poder trabajar con varios lenguajes. Adicionalmente, Arduino no cuenta con todas las características del lenguaje C++, por lo que es más simple en su uso siendo esta una ventaja al no tratarse de un proyecto tan complejo.

3.3 Costos de Materiales

Los componentes utilizados para este proyecto fueron elegidos debido a su bajo costo en el mercado, así como su fácil implementación para la construcción de un circuito. Así mismo, se tuvo en cuenta si se podían adquirir en lugares cercanos o si era necesario realizar un pedido de un lugar externo al país con anticipación, pero para este caso todos los componentes pudieron ser obtenidos en locales cercanos ubicados en el país.

Tabla 1: Costo total de los materiales utilizados en el proyecto

Componentes	Precio individual (\$)	Cantidad utilizada en el proyecto	Total a pagar por componente (\$)
Protoboard	4.20	1	4.20
ESP 32	12.50	1	12.50
Módulo RTC	7.50	1	7.50
Luces LED	0.10	11	1.10
Jumpers	0.10	19	1.90
Resistores	0.02	10	0.20
Costo total			\$27.40

Como se puede notar en la Tabla 1, el gasto total no es elevado, lo que permite la realización del proyecto a un precio accesible. Cabe recalcar que la parte

del software, que en este caso es el lenguaje de programación Arduino, no es parte del costo como tal porque se puede utilizar de forma gratuita al ser un lenguaje de acceso libre.

CAPÍTULO IV

4. ANÁLISIS DE RESULTADOS

En esta última sección, se presentan los diversos resultados obtenidos del proyecto posterior a las etapas de investigación y diseño, obteniendo el diagrama esquemático final del circuito implementado conforme a las características requeridas, y acompañado de las pruebas respectivas que validan su funcionamiento.

Es preciso mencionar que este proyecto se centra única y exclusivamente en el desarrollo de un controlador de tráfico que se sincroniza con la llegada de un pulso de reloj externo. El proyecto no considera el tema de la recepción de información enviada vía inalámbrica desde una central de control de tránsito, de modo que se ha acordado simular la recepción de dicho pulso con el módulo RTC.

4.1 Implementación del dispositivo

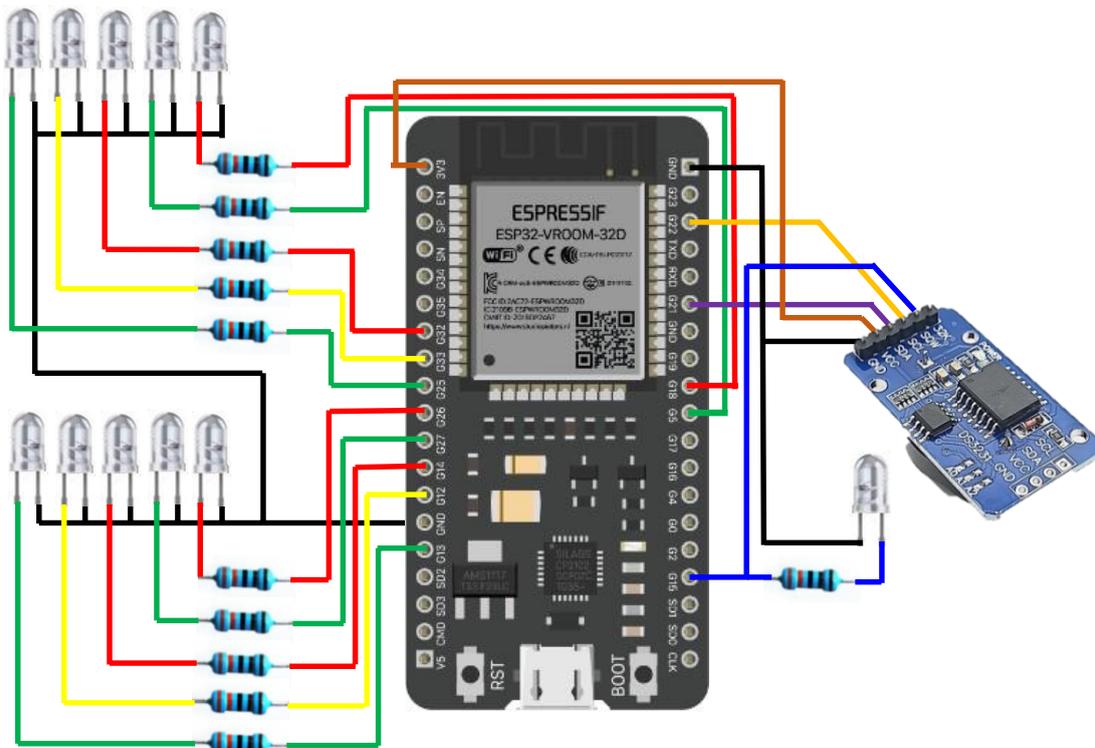


Ilustración 19: Diseño del prototipo de control de tráfico

En la Ilustración 19 se presenta el diseño de la topología implementada para el prototipo de control de tráfico, este circuito está constituido por un módulo microcontrolador ESP32-WROOM-32, un módulo RTC DS3231 y 11 luces led con sus respectivas resistencias; 10 de ellas ya se mencionaron anteriormente, por lo que la restante es un led azul utilizado para la visualización del pulso de reloj externo utilizado para la sincronización.

El dispositivo de la Ilustración 18 será sincronizado a partir de un pulso de reloj externo enviado desde el módulo RTC (Real Time Clock) y que para su verificación se ha colocado un led azul con la frecuencia de 1 Hz, es decir, que llegará un pulso cada segundo.

El prototipo implementado consiste en un semáforo de dos caras que se los ha denominado semáforo 1 y semáforo 2. Cada cara contiene 5 luces que corresponden a los colores mostrados a continuación y que han sido abreviados utilizando una particular nomenclatura, como se puede observar en la Ilustración 20:

- Verde 1: verde1
- Amarillo 1: amarillo1
- Rojo 1: rojo1
- Verde direccional 1: vd1
- Rojo direccional 1: rd1
- Verde 2: verde2
- Amarillo 2: amarillo2
- Rojo 2: rojo2
- Verde direccional 2: vd2
- Rojo direccional 2: rd2

```
int verde1 = 13;
int amarillo1 = 12;
int rojo1 = 14;
int vd1 = 27;
int rd1 = 26;

int verde2 = 25;
int amarillo2 = 33;
int rojo2 = 32;
int vd2 = 18;
int rd2 = 19;
```

Ilustración 20: Asignación de nomenclatura y designación de pines

El ciclo del semáforo está configurado con una duración de 60 segundos que se reparten en cuatro tiempos o eventos. Los eventos t1 y t3 tienen una duración de 27 segundos y los eventos t2 y t4 3 segundos de duración, de manera que la sumatoria de estos tiempos es igual a la duración del ciclo.

Los eventos que constituyen el ciclo conforman una secuencia de encendido de las luces como se detalla a continuación:

- Evento 1: verde1 – vd1 y rojo2 – rd2
- Evento 2: amarillo1 – vd1 y rojo2 – rd2
- Evento 3: rojo1 – rd1 y verde2 – vd2
- Evento 4: rojo1 – rd1 y amarillo2 – vd2

Estos eventos siguen una secuencia ordenada desde el primero hasta el cuarto, y para que cumpla con el funcionamiento de un semáforo, una vez culminado el evento 4, se reinicia el ciclo.

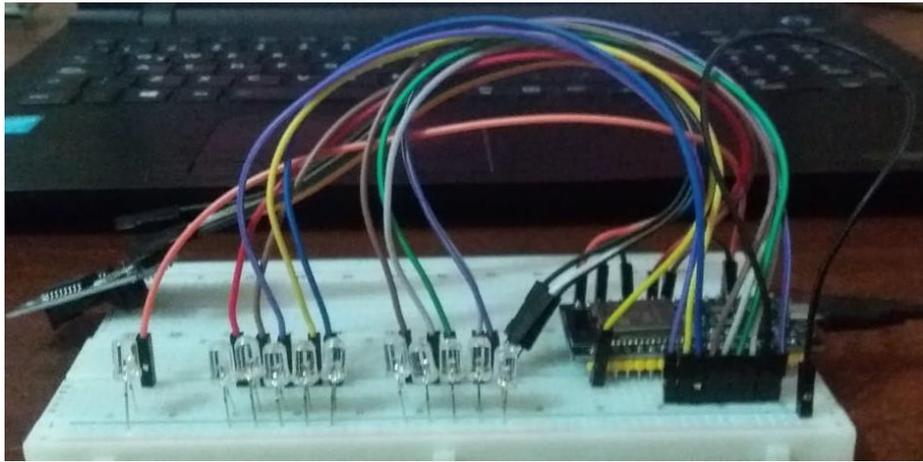


Ilustración 21: Vista lateral del circuito en etapa de pruebas

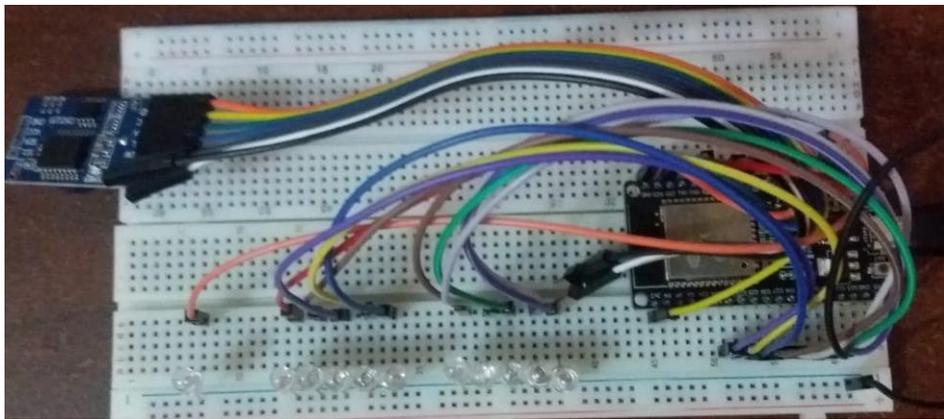


Ilustración 22: Vista frontal del circuito en etapa de pruebas

En las Ilustraciones 21 y 22 se presenta una vista lateral y frontal del circuito implementado en la placa de pruebas durante el montaje del circuito para sus respectivas pruebas de funcionamiento.

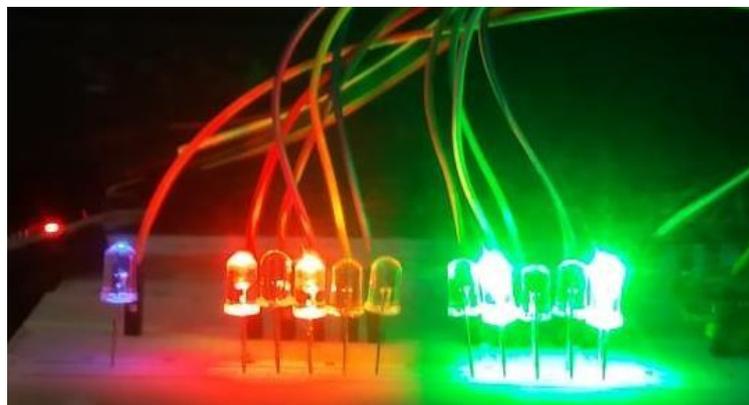


Ilustración 23: Inicia la secuencia normal con verde1-vd1 y rojo2-rd2

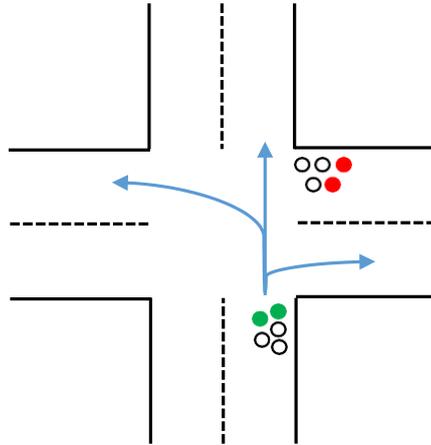


Ilustración 24: Verde 1 - verde direccional 1 y rojo 2 - rojo direccional 2

En esta Ilustración 23 se muestra el inicio del ciclo del semáforo que corresponde al primer evento relacionado a t_1 con la ejecución de las respectivas luces, verde 1 con verde direccional 1 en el semáforo 1 y rojo 2 con rojo direccional 2 en el semáforo 2, este juego de luces permite habilitar tres opciones en las que un conductor puede dirigirse a la izquierda, derecha y de frente como se indica en la Ilustración 24.

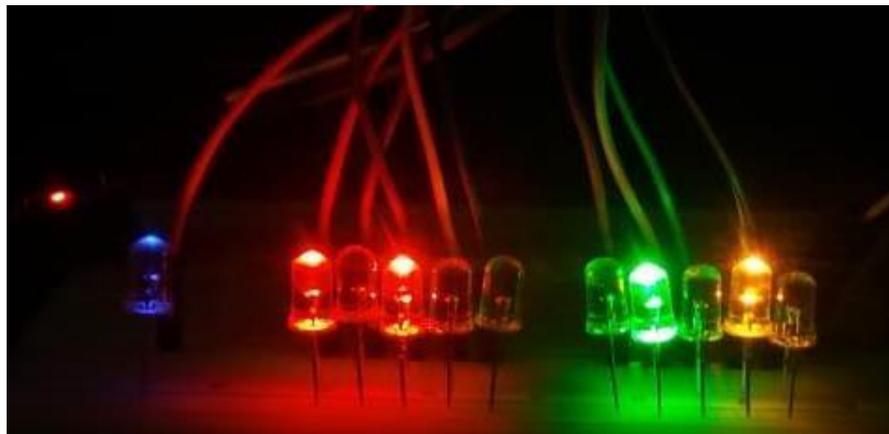


Ilustración 25: Continuación de la secuencia amarillo1-vd1 y rojo2-rd2

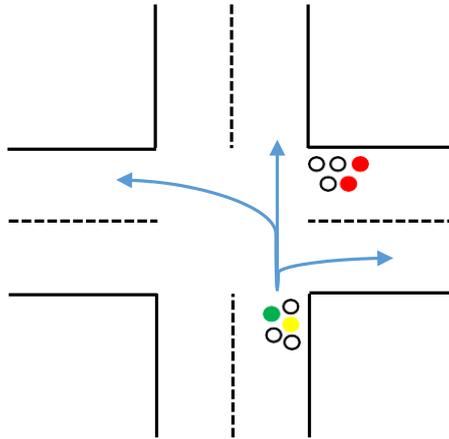


Ilustración 26: Amarillo 1 - verde direccional 1 y rojo 2 - rojo direccional 2

Con la Ilustración 25 se muestra la continuación del segundo evento relacionado a t_2 con el juego de luces amarillo 1 con verde direccional 1 en el semáforo 1 y rojo2 con rojo direccional 2 en el semáforo 2, permitiendo habilitar tres opciones en las que un conductor puede dirigirse como izquierda, derecha y de frente como se indica en la Ilustración 26.

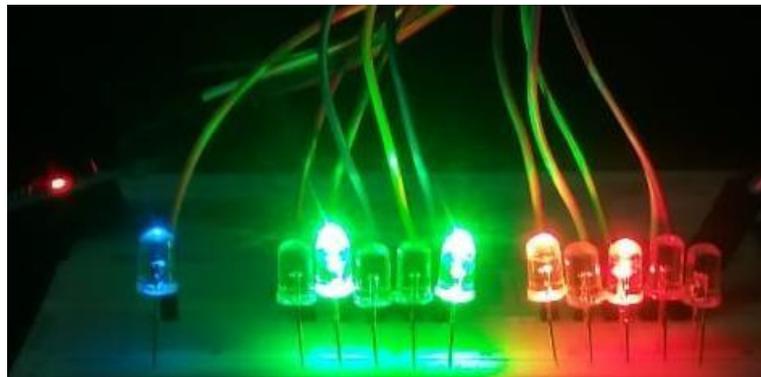


Ilustración 27: Continuación de la secuencia rojo1-rd1 y verde2-vd2

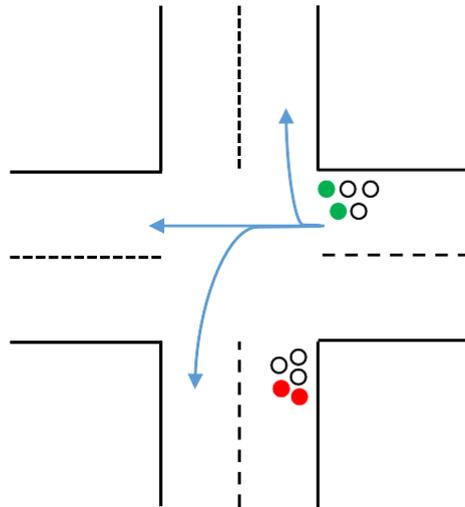


Ilustración 28: Rojo 1 - rojo direccional 1 y verde 2 - verde direccional 2

El tercer evento relacionado a t_3 corresponde a las luces rojo 1 con rojo direccional 1 en el semáforo 1 y verde 2 con verde direccional 2 en el semáforo 2, este juego de luces capturado se evidencia en la Ilustración 27 de modo que, se habilitan tres opciones en las que un conductor puede dirigirse como izquierda, derecha y de frente como se indica en la Ilustración 28.

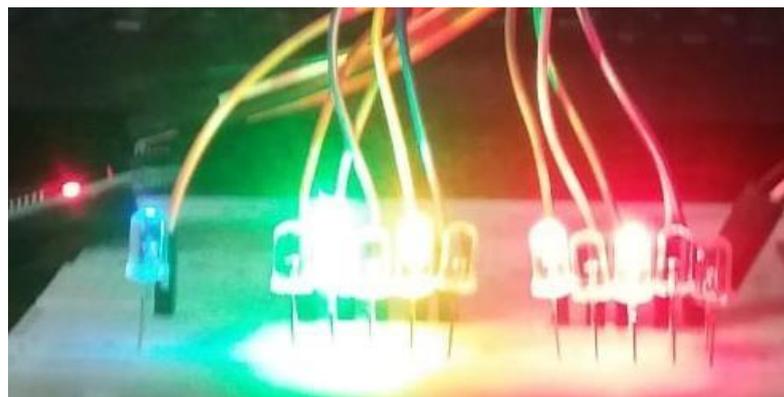


Ilustración 29: Continuación de la secuencia rojo1-rd1 y amarillo2- vd2

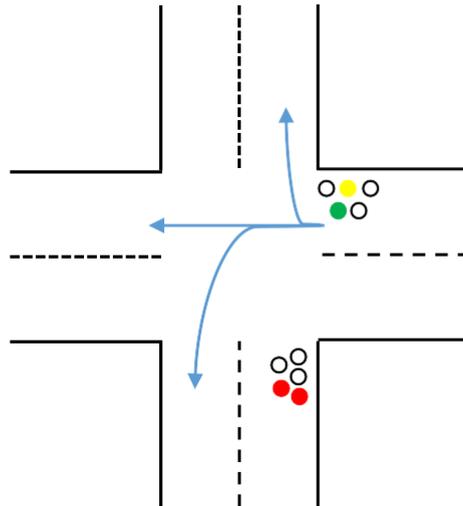


Ilustración 30: Rojo 1 - rojo direccional 1 y amarillo 2 - verde direccional 2

Para terminar con el cuarto evento relacionado a t4, se captura el juego de luces rojo 1 con rojo direccional 1 en el semáforo 1 y amarillo 2 con verde direccional 2 como se evidencia en la presente Ilustración 29, permitiendo habilitar tres opciones en las que un conductor puede dirigirse a la izquierda, derecha y de frente como se indica en la Ilustración 30.

Adicionalmente, se puede evidenciar que en los 4 eventos se ha capturado también el led azul que corresponde al pulso de sincronismo, es decir, en las Ilustraciones 23, 25, 27 y 29.

Cabe mencionar que, el módulo RTC envía constantemente pulsos cada segundo, por lo que, con el propósito de demostrar el funcionamiento de la secuencia del semáforo, se acordó configurarlo para que pasado determinado tiempo el microprocesador active el GPIO 15 encargado de la lectura del pulso externo, permitiendo de esta manera que se active la segunda parte de la implementación que es la sincronización del juego de luces con la llegada del pulso externo

4.2 Funcionamiento del dispositivo

El funcionamiento del prototipo se resume en un diagrama de flujo mostrado en la Ilustración 31, el proceso inicia con la alimentación del sistema operativo de tiempo real RTOS (Real Time Operation System), es decir, se conecta a una fuente

de energía el microcontrolador ESP32-WROOM-32 mediante puerto USB, e inicia instantáneamente la secuencia de eventos del semáforo.

En el microcontrolador ESP32 se lee constantemente el GPIO 15 en espera de recibir un pulso externo de sincronización, dependiendo de este pulso se escoge entre dos opciones, la primera opción es que el GPIO 15 no registra dicho pulso, es decir, que está recibiendo un 0 lógico de modo que el dispositivo debe continuar con la secuencia de los semáforos normalmente, y la segunda opción es que el mismo GPIO 15 registra el pulso, es decir, que esta ocasión se ha detectado un 1 lógico de modo que el dispositivo continuará hasta terminar el ciclo en que se recibe el pulso de sincronismo.

Este punto es crucial para la realización de este proyecto, ya que, a partir del instante en que termina el ciclo comienza la sincronización del dispositivo controlador de tráfico, para lograr el objetivo, en el microcontrolador ESP32 se registra el tiempo exacto en que se recibió el pulso externo de sincronismo entre 0 y 60 segundos.

La sincronización del prototipo con la llegada del pulso externo se cumple al terminar la ejecución de cuatro nuevos ciclos del semáforo con duración variable dependiendo del tiempo registrado, esto indica que se afecta al tiempo de duración de encendido de las luces, pero se ha configurado que este cambio se produzca solamente para las luces rojo y verde mas no para la amarilla.

La variación de los nuevos ciclos se relaciona con el valor del tiempo registrado a la llegada del pulso, éste valor se divide para el número de ciclos que le toma sincronizarse, 4 ciclos, y como solamente cambian los ciclos de las luces verde y rojo del semáforo, el resultado de la razón anterior se divide para 2, entonces este valor calculado se suma a los eventos t_1 y t_3 en los cuatro ciclos, consiguiendo con ello que el punto de inicio del semáforo alcance el punto en donde llegó el pulso externo y sincronizarse exitosamente. Para una mayor comprensión se explicará con una ilustración más adelante.

Finalmente, el dispositivo retoma el ciclo de eventos con la misma duración que se mencionó inicialmente.

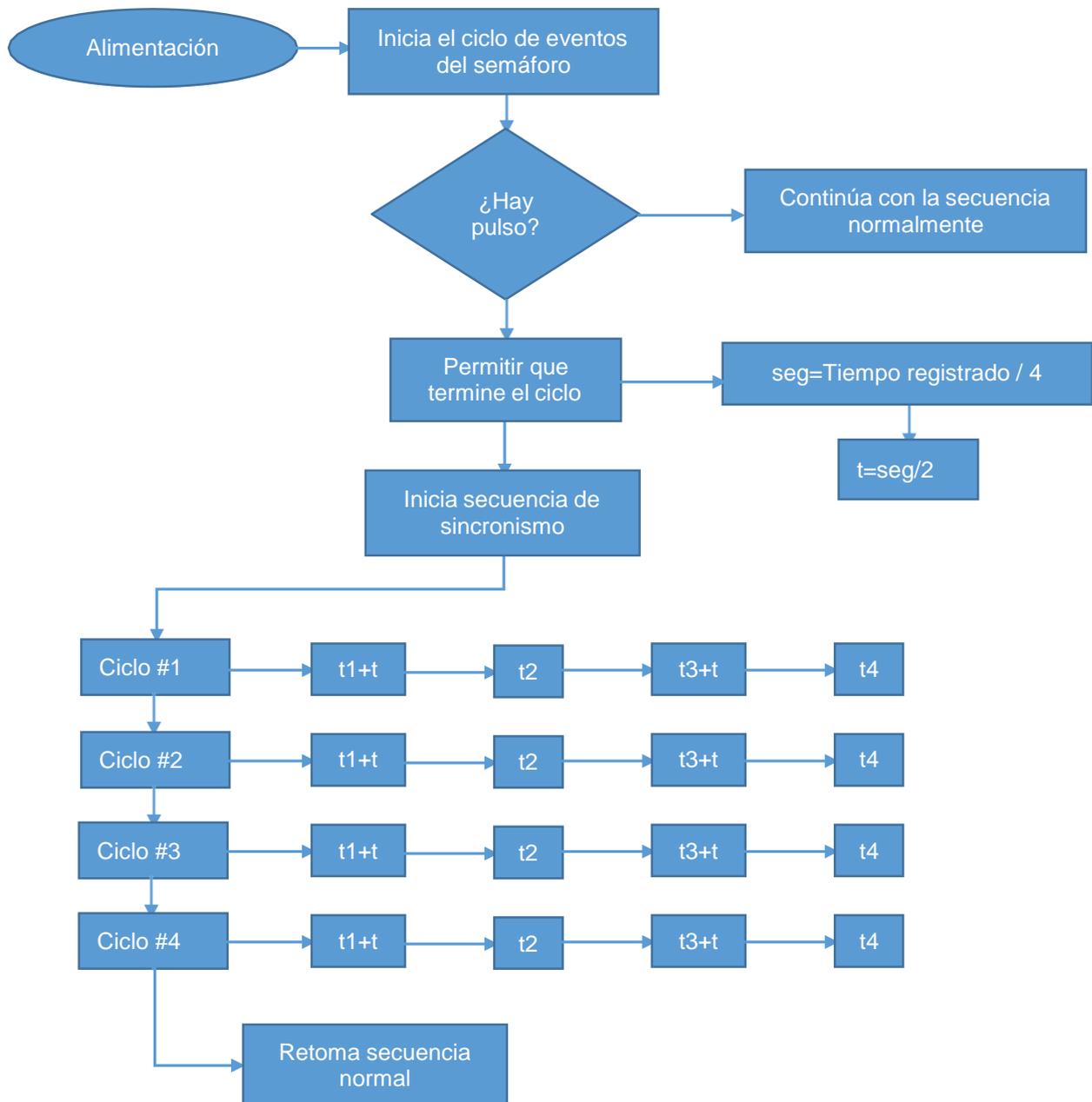


Ilustración 31: Diagrama de flujo

Este diagrama de flujo puede ser visto en código de programación en partes en el Anexo 1, Anexo 2, Anexo 3, Anexo 4, Anexo 5 y Anexo 6 correspondiente al algoritmo desarrollado en Arduino.

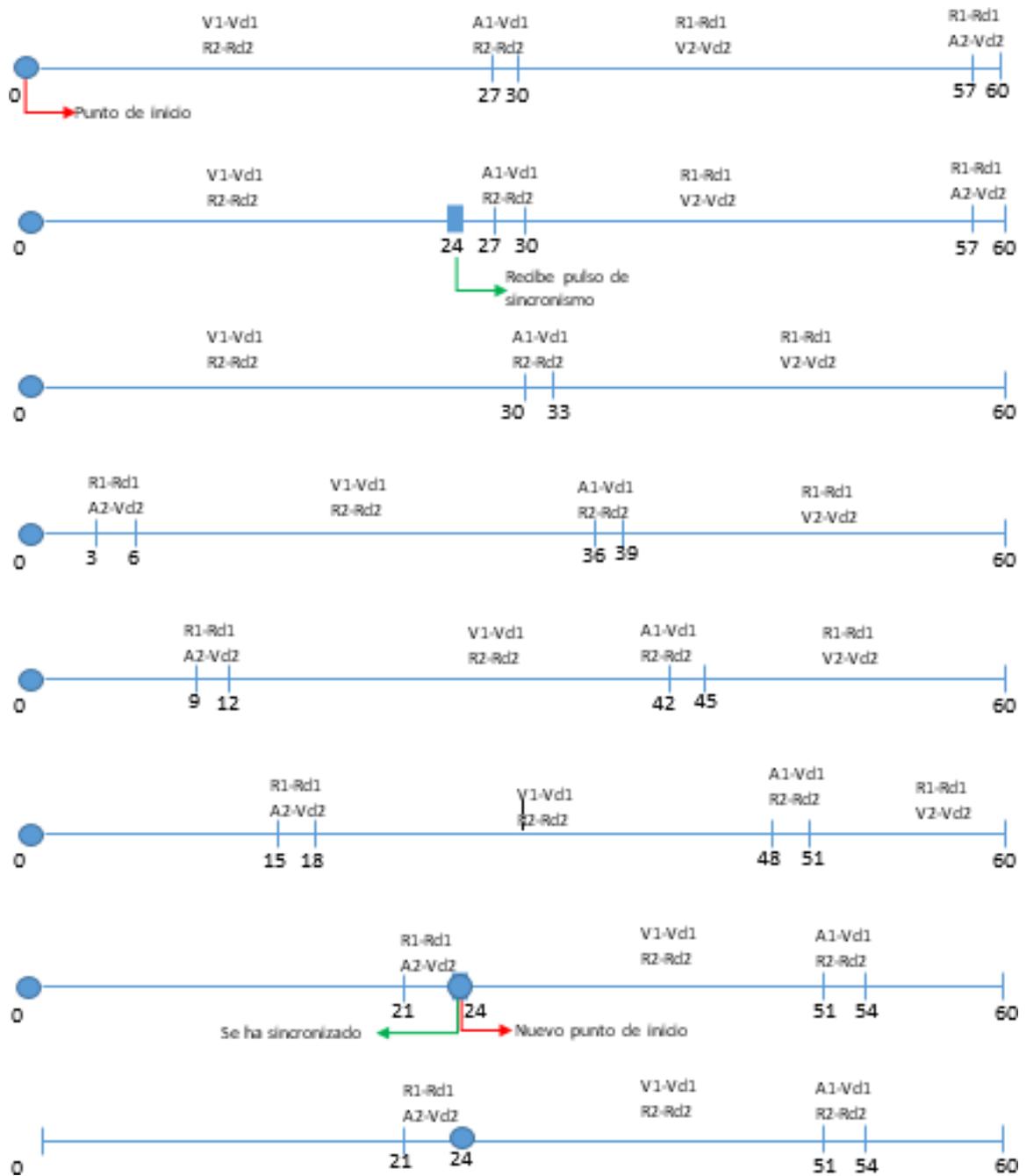


Ilustración 32: Línea de tiempo de los eventos durante la sincronización

Para un mayor entendimiento del funcionamiento del dispositivo se presenta la Ilustración 32 como ayuda visual para el caso presentado. La primera línea de tiempo muestra la secuencia normal del sistema sin irregularidades, pero en la segunda línea se nota la llegada de un pulso de sincronismo a los 24 segundos, sin embargo, el ciclo no ha variado.

Continuando con la tercera línea se aprecia que la duración de los eventos t_1 y t_3 ha cambiado de 27 a 30 segundos, esto se debe a que, el algoritmo divide los 24 segundos para 4 eventos obteniendo 6 segundos que a su vez se dividen entre 2, de modo que 3 segundos se suman a los eventos t_1 y t_3 durante los siguientes cuatro ciclos del semáforo.

Nótese que en cada ciclo se suma 6 segundos, por lo tanto, al terminar con los cuatro ciclos ha transcurrido 4 minutos con 24 segundos y como se puede evidenciar en la séptima línea de tiempo, el punto de inicio del semáforo se ha trasladado 24 segundos con lo que logra la sincronización con el pulso externo. Una vez sincronizado el dispositivo continúa con los mismos tiempos iniciales de t_1 y t_3 .

4.3 Pruebas realizadas

Semáforo 1	Semáforo 1	Semáforo 1
Semáforo 2	Semáforo 2	Semáforo 2
14	35	28
Sincronizar con pulso	Sincronizar con pulso	Sincronizar con pulso
Iniciar secuencia pulso	Iniciar secuencia pulso	Iniciar secuencia pulso
28750	31375	30500
28750	31375	30500
Iniciar secuencia pulso	Iniciar secuencia pulso	Iniciar secuencia pulso
28750	31375	30500
28750	31375	30500
Iniciar secuencia pulso	Iniciar secuencia pulso	Iniciar secuencia pulso
28750	31375	30500
28750	31375	30500
Iniciar secuencia pulso	Iniciar secuencia pulso	Iniciar secuencia pulso
28750	31375	30500
28750	31375	30500
Retoma secuencia normal	Retoma secuencia normal	Retoma secuencia normal
Semáforo 1	Semáforo 1	Semáforo 1
Semáforo 2	Semáforo 2	Semáforo 2

Ilustración 33: Pruebas de sincronización con diferentes tiempos de pulso

Con el propósito de validar el funcionamiento del prototipo, se realizaron varias pruebas de concepto del funcionamiento del dispositivo controlador de tráfico diseñado, de las cuales se pueden apreciar 3 en la Ilustración 33, para ello se tomaron tres tiempos diferentes en la llegada del pulso externo de sincronismo, 14 segundos, 35 segundos y 28 segundos.

Con la ayuda del algoritmo se tomó cada uno de ellos para la ejecución de los respectivos cálculos obteniendo los nuevos tiempos de duración para los eventos t1 y t3.

En la primera prueba se captura el pulso externo de sincronismo en el primer ciclo del semáforo a los 14 sg, siendo 1.750 sg el tiempo que se adiciona a los eventos t1 y t3, teniendo como resultado 28.750 sg, es decir, que por ciclo se suma 3.500 sg por lo que al cabo de los 4 ciclos se habrá sumado finalmente 14 sg logrando sincronizarse con el pulso externo.

Como ya se ha mencionado, el ciclo del semáforo es de 60 sg y el tiempo que tarda en sincronizarse es de 4 ciclos más el tiempo de llegada del pulso, en

este caso es 14 segundos, esto es 4 min y 14 sg de modo que el semáforo iniciará en verde a los 5 min y 14 segundos como se muestra en el Anexo 7 en la línea 20 del cronómetro.

En la segunda prueba se captura el pulso externo de sincronismo en el primer ciclo del semáforo a los 35 sg, siendo 4.375 sg el tiempo que se adiciona a los eventos t1 y t3, teniendo como resultado 31.375 sg, es decir, que por ciclo se suma 8.750 sg por lo que al cabo de los 4 ciclos se habrá sumado finalmente 35 sg logrando sincronizarse con el pulso externo.

El tiempo de duración del ciclo del semáforo es de 60 sg y el tiempo que tarda en sincronizarse es de 4 ciclos más el tiempo de llegada del pulso, en este caso es 35 segundos, esto es 4 min y 35 sg de modo que el semáforo iniciará en verde a los 5 min y 35 segundos como se muestra en el Anexo 8 en la línea 20 del cronómetro.

Para terminar con la tercera prueba se captura el pulso externo de sincronismo en el segundo ciclo del semáforo a los 28 sg, siendo 3.500 sg el tiempo que se adiciona a los eventos t1 y t3, teniendo como resultado 30.500 sg, es decir, que por ciclo se suma 7 sg por lo que al cabo de los 4 ciclos se habrá sumado finalmente 28 sg logrando sincronizarse con el pulso externo.

El tiempo de duración del ciclo del semáforo es de 60 sg y el tiempo que tarda en sincronizarse es de 4 ciclos más el tiempo de llegada del pulso, en este caso es 28 segundos, esto es 4 min y 28 sg de modo que, el semáforo iniciará en verde a los 6 min y 28 segundos como se muestra en el Anexo 9 en la línea 24 del cronómetro.

Es preciso mencionar que, la captura de estos tiempos en los Anexos 7, 8 y 9 no son exactos debido a la impresión humana al registrarlos con cada cambio de luces, esto no implica un mal funcionamiento del prototipo.

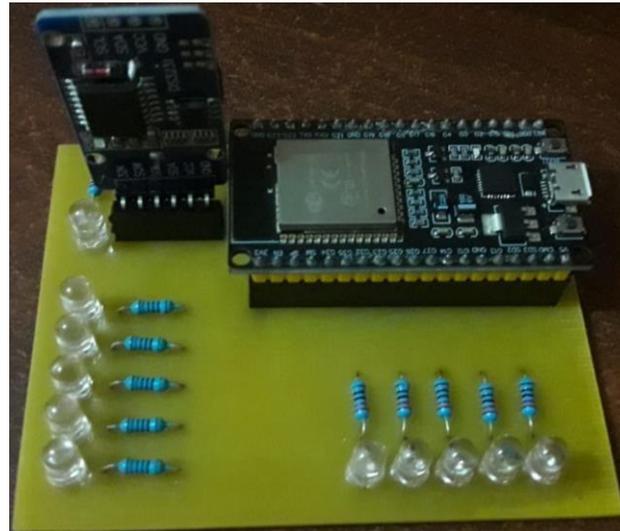


Ilustración 34: PCB (Placa de Circuito Impreso)

Finalmente, tras la realización de las pruebas se procede con la construcción de la placa de circuito impreso del prototipo diseñado, que proviene del inglés Printed Circuit Board (PCB), obteniendo como resultado la placa observada en la Ilustración 34, para lo cual se utilizó el software Proteus que permite la manipulación de los materiales para posicionarlos y a su vez la creación de las pistas como se muestra en la siguiente Ilustración 35.

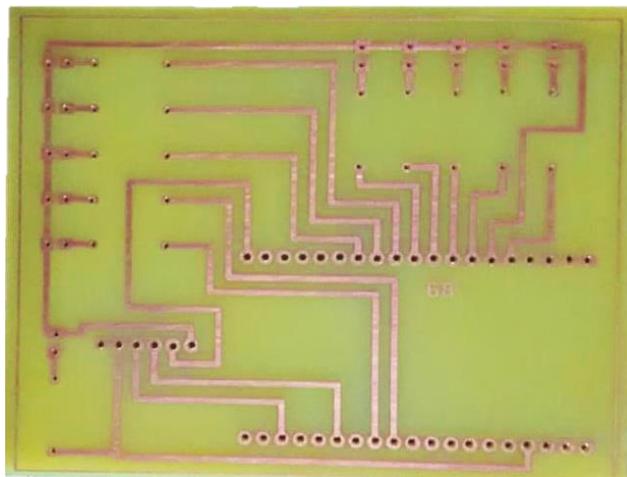


Ilustración 35: Pistas de la PCB

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Se logró implementar un algoritmo capaz de sincronizar los semáforos de manera que no ocurran cambios bruscos al momento de modificar los tiempos de la secuencia de luces para de esa manera evitar accidentes de tráfico en caso de que en un momento dado alguno de los semáforos cambie de un color a otro inesperadamente no siguiendo el orden predeterminado.

Se realizó la sincronización de los semáforos en 4 ciclos haciendo uso de la lectura de pulsos de reloj externos que llegan de manera automática presentando la ventaja de no requerir de que un técnico realice las modificaciones in situ de manera manual, permitiendo una mayor flexibilidad e integración en un sistema que controle el tráfico urbano. El diseño propuesto proporciona un ahorro de personal que reduce los costos de operación del sistema.

La implementación del controlador del semáforo fue exitosa ya que se comprobó el cumplimiento de los requerimientos de la sincronización en el menor tiempo posible y de la forma menos perceptible. Por facilidad de mediciones y manipulación, se incorporaron los materiales necesarios a una placa de circuito impreso.

Recomendaciones

Consultar detalladamente la información o el datasheet de los equipos y materiales empleados para la correcta conexión de elementos. Se presentó el caso de duda con el microprocesador ESP32-WROOM-32 para la comunicación serial SDA y SCL en donde se tenía la asignación de pines 21 y 22 en forma específica, diferente a lo que asumió originalmente.

Revisar la configuración de los 38 pines del microcontrolador ESP32-WROOM-32 debido a que en su configuración de fábrica la mayoría están habilitados como pero muy pocos están disponibles como entradas.

Tener precaución de utilizar brazaletes de aterrizamiento cuando se trabaja con microprocesadores, debido a que estos podrían dañarse por la estática al entrar en contacto con las manos.

BIBLIOGRAFÍA

- [1] EXTRA, «Quito y Guayaquil, entre las ciudades con el peor tráfico de Latinoamérica,» EXTRA, 21 Febrero 2019. [En línea]. Available: <https://www.extra.ec/noticia/actualidad/transito-congestion-vehicular-estudio-mundial-2649535.html>. [Último acceso: 9 Octubre 2022].
- [2] Mundo, BBC, «Las ciudades y países con el tráfico más congestionado de América Latina,» BBC News Mundo, 21 Febrero 2017. [En línea]. Available: <https://www.bbc.com/mundo/noticias-39045803>. [Último acceso: 9 Octubre 2022].
- [3] C. Barría, «Cuál es la ciudad con el peor tráfico vehicular de América Latina (y cómo podría mejorar su problema),» BBC News Mundo, 8 Marzo 2019. [En línea]. Available: <https://www.bbc.com/mundo/noticias-47473793>. [Último acceso: 9 Octubre 2022].
- [4] K. H. Diana Sotomayor, «Las luces y los agentes son la rémora del volante en Guayaquil y Quito,» EXPRESO, Julio 17 2022. [En línea]. Available: <https://www.expreso.ec/guayaquil/luces-agentes-son-remora-volante-quito-131778.html>. [Último acceso: 9 Octubre 2022].
- [5] EL UNIVERSO, «Unos 45.000 carros por año entran a rodar en Guayaquil,» EL UNIVERSO, 01 Febrero 2019. [En línea]. Available: <https://www.eluniverso.com/guayaquil/2019/02/01/nota/7166981/45000-carros-ano-entran-rodar-guayaquil/>. [Último acceso: 11 Octubre 2022].
- [6] K. Pesantes, «Guayaquil, una de las 25 ciudades con más tráfico en el mundo, opta por trabajo comunitario en lugar de multas,» PRIMICIAS, 07 Junio 2019. [En línea]. Available: <https://www.primicias.ec/noticias/sociedad/guayaquil-traffic-trabajo-comunitario-multas/>. [Último acceso: 15 Octubre 2022].

- [7] B. X. Becerra, «Bogotá es la cuarta ciudad del mundo con peor tráfico vehicular según Traffic Index,» LA REPÚBLICA, 18 Marzo 2022. [En línea]. Available: <https://www.larepublica.co/globoeconomia/bogota-cuarta-ciudad-en-el-mundo-con-el-peor-trafico-vehicular-segun-nuevo-ranking-3325102>. [Último acceso: 15 Octubre 2022].
- [8] EL UNIVERSO, «Los horarios de mayor tráfico vehicular en Guayaquil y Samborondón,» EL UNIVERSO, 17 Mayo 2022. [En línea]. Available: <https://www.eluniverso.com/guayaquil/comunidad/los-horarios-de-mayor-trafico-vehicular-en-guayaquil-y-samborondon-nota/>. [Último acceso: 15 Octubre 2022].
- [9] La Hora, «¿Sabe cuántas horas pasa un quiteño en el tráfico?,» 25 Marzo 2022. [En línea]. Available: <https://www.lahora.com.ec/pais/quito-perdida-tiempo-trafico/#:~:text=Pues%20si%20vive%20en%20Quito,%2C%20representa%202%2C6%20d%C3%ADas..> [Último acceso: 9 Octubre 2022].
- [10] I. Arriaga, «Una Década de impluso a la movilidad,» *Transporte & Ciudad: Foro de la movilidad*, p. 8, 2018.
- [11] S. Gómez, «Estas son las ciudades con peor tráfico en Estados Unidos,» UNIVISION, 8 Febrero 2019. [En línea]. Available: <https://www.univision.com/noticias/ranking-de-autos/estas-son-las-ciudades-con-peor-trafico-en-estados-unidos-fotos>. [Último acceso: 10 Octubre 2022].
- [12] CERO ACCIDENTES, «Qué saber sobre los efectos respiratorios de contaminantes en el aire causados por el tráfico,» Mayo 31 2022. [En línea]. Available: <https://www.ceroaccidentes.pe/que-saber-sobre-los-efectos-respiratorios-de-contaminantes-en-el-aire-causados-por-el-trafico/>. [Último acceso: 24 Octubre 2022].
- [13] A. Frakt, «Atascado y estresado: cómo afecta el tráfico a tu salud,» New York Times, 7 Febrero 2019. [En línea]. Available: <https://www.nytimes.com/es/2019/02/07/espanol/consecuencias-trafico->

salud.html#:~:text=Otro%20de%20los%20efectos%20en,tambi%C3%A9n%20pu
ede%20someterse%20a%20mediciones.. [Último acceso: 10 Octubre 2022].

- [14] EL UNIVERSO, «Por fases arranca el plan que busca mejor sincronización de semáforos en Guayaquil,» EL UNIVERSO, 26 Noviembre 2017. [En línea]. Available: <https://www.eluniverso.com/guayaquil/2017/11/26/nota/6497695/fases-arranca-plan-que-busca-mejor-sincronizacion-semaforos/>. [Último acceso: 14 Octubre 2022].
- [15] LAS NACIONES UNIDAS, «Objetivos de Desarrollo Sostenible,» 24 Mayo 2022. [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/cities/>. [Último acceso: 25 Octubre 2022].
- [16] SICE, «El semáforo: 150 años de historia,» 06 Febrero 2019. [En línea]. Available: <https://www.sice.com/actualidad/el-semaforo-150-anos-de-historia->. [Último acceso: 15 Octubre 2022].
- [17] R. Maiquez, «PROYECTO DE SEMAFORIZACION DEL AREA URBANA DE LA CIUDAD DE SHUSHUFINDI,» Enero 2017. [En línea]. Available: https://www.compraspublicas.gob.ec/ProcesoContratacion/compras/PC/bajarArchivo.cpe?Archivo=IG31SadpUru_4eelB1aGq7C9vQ85Blq_iXUTuhq_Yh0,. [Último acceso: 18 Octubre 2022].
- [18] C. Castiblanco Herrera, «Detrás del Alcalde,» Bogotá, 3 Junio 2019. [En línea]. Available: <https://bogota.gov.co/yo-participo/blogs/semaforizacion-inteligente-en-bogota>. [Último acceso: 16 Octubre 2022].
- [19] Nexu, «Diseñan en México un semáforo inteligente,» 25 Abril 2017. [En línea]. Available: <https://www.nexu.mx/blog/disenan-en-mexico-un-semaforo-inteligente/>. [Último acceso: 14 Octubre 2022].
- [20] E. E. MAYA JARQUIN, «CONTROL INTELIGENTE PARA SEMÁFOROS DE LA AV. CHAPULTEPEC DE LA CIUDAD DE MÉXICO,» Enero 2018. [En línea].

Available: <https://tesis.ipn.mx/xmlui/handle/123456789/27563?show=full>. [Último acceso: 15 Octubre 2022].

- [21] ANSA Latina, «Web y semáforos inteligentes,» 9 Febrero 2018. [En línea]. Available: https://www.ansalatina.com/americalatina/noticia/tecnologia/2018/02/09/internet-ayuda-a-los-semaforos-inteligentes_840adc65-cf71-4f5f-bfb7-5beff04f7873.html. [Último acceso: 16 Octubre 2022].
- [22] M. J. B. Ing. Teresa Rodriguez, «IoT para la semaforización inteligente en la ciudad de Guayaquil,» Polo del Conocimiento, 17 Noviembre 2021. [En línea]. Available: <https://polodelconocimiento.com/ojs/index.php/es/article/view/3313>. [Último acceso: 19 Octubre 2022].
- [23] E. Castro, «Ford trabaja en el desarrollo de semáforos inteligentes para vehículos de emergencia,» Mexico Industry, 31 Marzo 2022. [En línea]. Available: <https://mexicoindustry.com/noticia/ford-trabaja-en-el-desarrollo-de-semaforos-inteligentes-para-vehiculos-de-emergencia>. [Último acceso: 17 Octubre 2022].
- [24] J. A. Pascual Estapé, «Computer Hoy,» 23 Marzo 2020. [En línea]. Available: <https://computerhoy.com/noticias/motor/semaforos-anti-ruido-607079>.
- [25] DW Made for minds, «Este nuevo sistema de semáforos inteligentes con IA podría hacer que los atascos sean un recuerdo lejano,» 16 Mayo 2022. [En línea]. Available: <https://www.dw.com/es/este-nuevo-sistema-de-sem%C3%A1foros-inteligentes-con-ia-podr%C3%ADa-hacer-que-los-atascos-sean-un-recuerdo-lejano/a-61817980>. [Último acceso: 17 Octubre 2022].
- [26] V. NADAL, «Adiós al atasco mañanero: semáforos inteligentes que observan el tráfico y aprenden de él,» EL PAIS, 31 Diciembre 2018. [En línea]. Available: https://elpais.com/retina/2018/12/19/innovacion/1545224931_036265.html. [Último acceso: 19 Octubre 2022].

- [27] H. Barcia, «Viena instala semáforos inteligentes que se activan cuando deseas cruzar,» 14 Noviembre 2019. [En línea]. Available: <https://www.lavanguardia.com/vida/20191114/471597940456/viena-instala-semaforos-inteligentes-que-se-activan-cuando-deseas-cruzar.html>. [Último acceso: 19 Octubre 2022].
- [28] M. Alba y O. Hernández, «Análisis de sincronización de semáforos utilizando el programa Synchro,» *Revista Infraestructura Vial*, vol. 22, nº 39, pp. 1-11, 2020.
- [29] F. Lozado y A. Méndez, «DISEÑO DE UN PROTOTIPO PARA EL CONTROL DE TRÁFICO VEHICULAR EN EL CANTÓN EL TRIUNFO, PROVINCIA DEL GUAYAS,» Mayo 2019. [En línea]. Available: <https://repositorio.unemi.edu.ec/bitstream/123456789/4493/1/DISE%C3%91O%20DE%20UN%20PROTOTIPO%20PARA%20EL%20CONTROL%20DE%20TR%C3%81FICO%20VEHICULAR%20EN%20EL%20CANT%C3%93N%20EL%20TRIUNFO,%20PROVINCIA%20DEL%20GUAYAS.pdf>. [Último acceso: 17 Octubre 2022].
- [30] J. L. Soto, «Alemania ya prueba los semáforos inteligentes,» *EL MOTOR*, 28 Febrero 2022. [En línea]. Available: <https://motor.elpais.com/tecnologia/alemania-ya-prueba-los-semaforos-inteligentes/>. [Último acceso: 17 Octubre 2022].
- [31] J. C. Piña Pacheco y G. M. Zúñiga López, «ANÁLISIS COMPARATIVO DEL SISTEMA TRADICIONAL DE SEMAFORIZACIÓN VS UNA PROPUESTA DE SEMAFORIZACIÓN INTELIGENTE, PARA LA REDUCCIÓN DEL CONGESTIONAMIENTO VEHICULAR, EN LA CIUDAD DE GUAYAQUIL,» 20 Septiembre 2017. [En línea]. Available: <http://repositorio.ug.edu.ec/bitstream/redug/23908/1/TESIS%20SEMAFORIZACI%C3%93N%20I.pdf>. [Último acceso: 17 Octubre 2022].
- [32] A. Gutiérrez Solís, «Semáforos inalámbricos, solares y sincronizados,» 08 Abril 2017. [En línea]. Available:

<http://www.cienciamx.com/index.php/tecnologia/tic/14352-semaforos-inalambricos-solares-y-sincronizados>. [Último acceso: 17 Octubre 2022].

- [33] O. Bidegain, «Diseño y construcción de un prototipo de controlador de semáforos leds,» Noviembre 2019. [En línea]. Available: <http://rinfi.fi.mdp.edu.ar/bitstream/handle/123456789/410/OBidegain-TFG-IE-2019.pdf?sequence=1&isAllowed=y>. [Último acceso: 17 Octubre 2022].
- [34] W. Zambrano Zambrano, «SEMAFORIZACIÓN EN LAS INTERSECCIONES DE LAS CALLES BOLÍVAR Y JUAN MONTALVO COMO MEDIDA DE SEGURIDAD DEL CANTÓN PASAJE,» 27 Febrero 2020. [En línea]. Available: <http://repositorio.utmachala.edu.ec/bitstream/48000/15430/1/ECFIC-2020-IC-DE-00013.pdf>. [Último acceso: 17 Octubre 2022].
- [35] G. Campos, «EVALUACIÓN PARA LA DE SEMÁFOROS IMPLEMENTACIÓN INTELIGENTES DE B OULEVARD EN LOS CRUCEROS LUIS DONALDO COLOSIO,» Diciembre 2020. [En línea]. Available: <https://rinacional.tecnm.mx/bitstream/TecNM/2416/1/TESIS%20PERIFERICO%20-%20GEMA%20CAMPOS.pdf>. [Último acceso: 17 Octubre 2022].
- [36] A. Mera, «Microprocesador,» 14 Marzo 2019. [En línea]. Available: https://www.academia.edu/4044468/Microprocesador_1_MicroprocesadorCambri dge_Consultants_XAP_V%C3%A9ase_tambi%C3%A9n_Multin%C3%BAcleo_Mi crocontrolador_CPU_Conjunto_de_instrucciones_Arquitectura_de_computadores _Socket_de_CPU_Hardware. [Último acceso: 27 Noviembre 2022].
- [37] Intel, «Intel Pentium Processor,» 10 Junio 2017. [En línea]. Available: <https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/pentium-desktop-processor-brief.pdf>. [Último acceso: 27 Noviembre 2022].

- [38] C. Mendoza, «Microprocesadores,» Sutori, 4 Abril 2019. [En línea]. Available: <https://www.sutori.com/en/story/microprocesadores--Jx1UcJJSPiriQFfR5prjyen9>. [Último acceso: 29 Noviembre 2022].
- [39] Flagship Technologies Inc., «Intel Xeon 5000/5100 Dual-Core CPUs,» 17 Septiembre 2019. [En línea]. Available: <https://store.flagshiptech.com/intel-xeon-5000-5100-dual-core-cpus/>. [Último acceso: 27 Noviembre 2022].
- [40] Profesional Review, «Intel Xeon,» 12 Octubre 2021. [En línea]. Available: <https://www.profesionalreview.com/intel/intel-xeon/>. [Último acceso: 29 Noviembre 2022].
- [41] Intel, «Presentamos los procesadores Intel Atom® C5000, P5300 y P5700,» 6 Junio 2022. [En línea]. Available: <https://www.intel.la/content/www/xl/es/products/docs/processors/atom/c5000-p5300-p5700-product-brief.html>. [Último acceso: 27 Noviembre 2022].
- [42] Rs Components, «Intel CH80566EE025DW , Atom Microprocessor Z5XX 32bit IA-32 533MHz 437-Pin FCBGA8,» 18 Diciembre 2021. [En línea]. Available: <https://ph.rs-online.com/web/p/microprocessors/6827387>. [Último acceso: 2022 Noviembre 2022].
- [43] HP Tech Takes, «A Complete Guide to Intel Processors,» 30 Abril 2021. [En línea]. Available: <https://www.hp.com/us-en/shop/tech-takes/intel-processor-guide>. [Último acceso: 28 Noviembre 2022].
- [44] Nimavi, «Procesador CPU Intel Core i3 12100 4.30Ghz /12va.,» 31 Agosto 2022. [En línea]. Available: <https://nimavi.com/producto/procesador-cpu-intel-core-i3-12100-4-30ghz-12va-generacion-intel-uhd-730-socket-1700>. [Último acceso: 29 Noviembre 2022].

- [45] Profesional Review, «Intel Core i5,» 30 Octubre 2018. [En línea]. Available: <https://www.profesionalreview.com/intel/intel-core-i5/>. [Último acceso: 2022 Noviembre 2022].
- [46] Profesional Review, «Intel Core i9,» 30 Octubre 2018. [En línea]. Available: <https://www.profesionalreview.com/intel/intel-core-i9/>. [Último acceso: 29 Noviembre 2022].
- [47] M. H. El-Saba, «Introduction to Microcontrollers & Embedded Systems,» 2020. [En línea]. Available: https://www.researchgate.net/publication/349812349_Introduction_to_Microcontrollers_Embedded_Systems. [Último acceso: 30 Noviembre 2022].
- [48] Alfredo Moreno, Sheila Córcoles, «Aprende Arduino en un fin de semana,» 9 Agosto 2017. [En línea]. Available: <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/Arduinounfinseman.pdf>. [Último acceso: 30 Noviembre 2022].
- [49] Arduino.cl, «Arduino UNO,» 19 Febrero 2019. [En línea]. Available: <https://arduino.cl/arduino-uno/>. [Último acceso: 29 Noviembre 2022].
- [50] D. Pavlovic, «What is Raspberry Pi and Should I Try It Out?,» 23 Febrero 2020. [En línea]. Available: <https://www.hp.com/us-en/shop/tech-takes/what-is-raspberry-pi#:~:text=Raspberry%20Pi%20is%20a%20series,available%20at%20highly%20competitive%20prices..> [Último acceso: 5 Diciembre 2022].
- [51] E. Pérez, «Raspberry Pi Pico W: la nueva versión con WiFi rompe la gama de precios y ahora solo cuesta 6 euros,» 30 Junio 2022. [En línea]. Available: <https://www.xataka.com/makers/raspberry-pi-pico-w-nueva-version-wifi-rompe-gama-precios-ahora-solo-cuesta-6-euros#:~:text=La%20nueva%20Raspberry%20Pi%20Pico%20W%20ofrece%20c>

onectividad%20802.11n%20y,por%2021%20mm%20de%20ancho.. [Último acceso: 5 Diciembre 2022].

- [52] José Rentería, Raúl Ruelas, «Módulo ESP8266 y sus aplicaciones en el internet de las cosas,» *Revista de Ingeniería Eléctrica*, pp. 1-4, 2017.
- [53] T. Perets, «INVESTIGATION OF WI-FI (ESP8266) MODULE AND APPLICATION TO AN AUDIO,» Agosto 2019. [En línea]. Available: https://www.researchgate.net/publication/352750304_INVESTIGATION_OF_WI-FI_ESP8266_MODULE_AND_APPLICATION_TO_AN_AUDIO_SIGNAL_TRANSMISSION. [Último acceso: 30 Noviembre 2022].
- [54] DescubreArduino, «ESP32 vs ESP8266 ¿Cuales son las diferencias entre ambos módulos?,» 30 Septiembre 2022. [En línea]. Available: <https://descubrearduino.com/esp32-vs-esp8266/>. [Último acceso: 28 Noviembre 2022].
- [55] DescubreArduino, «¿Qué es el módulo ESP32?,» 30 Septiembre 2022. [En línea]. Available: <https://descubrearduino.com/esp32-vs-esp8266/>. [Último acceso: 29 Noviembre 2022].
- [56] V. Ozan, *Developing IoT Projects with ESP 32*, Birmingham: Packt, 2021.
- [57] javaTpoint, «Programming Language,» 14 Octubre 2020. [En línea]. Available: <https://www.javatpoint.com/programming-language>. [Último acceso: Diciembre 6 2022].
- [58] Coursera, «What Is Python Used For? A Beginner's Guide,» 14 Noviembre 2022. [En línea]. Available: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>. [Último acceso: Diciembre 6 2022].

- [59] Á. Robledano, «Qué es C++: Características y aplicaciones,» OpenWebinars, 22 Junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-cpp/>. [Último acceso: 6 Diciembre 2022].
- [60] «Arduino Editor- Web Editor,» 25 Abril 2020. [En línea]. Available: <https://create.arduino.cc/projecthub/lina-tech-explorations/the-basics-of-arduino-programming-program-structure-functi-f5fb2c>. [Último acceso: 6 Diciembre 2022].
- [61] I. BUCKLEY, «What Is a Breadboard and How Do You Use One?,» 22 Junio 2022. [En línea]. Available: <https://www.makeuseof.com/tag/what-is-breadboard/>. [Último acceso: 30 Diciembre 2022].
- [62] ECS Inc. International, «WHAT IS A REAL TIME CLOCK (RTC)?,» 17 Mayo 2022. [En línea]. Available: <https://ecsxtal.com/what-is-a-real-time-clock-rtc>. [Último acceso: 29 Noviembre 2022].
- [63] EE Power, «What Is a Resistor?,» 19 Agosto 2020. [En línea]. Available: <https://eepower.com/resistor-guide/resistor-fundamentals/what-is-a-resistor/>. [Último acceso: 30 Diciembre 2022].
- [64] C. Francois, «What is a Fixed Resistor?,» 19 Diciembre 2022. [En línea]. Available: <https://www.aboutmechanics.com/what-is-a-fixed-resistor.htm>. [Último acceso: 30 Diciembre 2022].
- [65] S. John, «Different Types of in Light Emitting Diodes (LED),» *IJRASET*, vol. 6, p. 1, 2018.
- [66] BARCELONA LED, «Características y Ventajas de los Diodos LED,» 30 Abril 2020. [En línea]. Available: <https://www.barcelonaed.com/blog/informacion-led/caracteristicas-y-ventajas-de-los-diodos-led/>. [Último acceso: 29 Noviembre 2022].

[67] Hardware, «Cable Jumper: qué es, para qué sirve, y dónde comprar,» 1 Febrero 2022. [En línea]. Available: <https://www.hwlibre.com/cable-jumper/>. [Último acceso: 30 Diciembre 2022].

ANEXOS

Anexo 1 – Código de programación (declaración de puertos de salida)

```
// Asignación de pines digitales para semáforo 1
```

```
int verde1 = 13;
```

```
int amarillo1 = 12;
```

```
int rojo1 = 14;
```

```
int vd1 = 27;
```

```
int rd1 = 26;
```

```
// Asignación de pines digitales para semáforo 2
```

```
int verde2 = 25;
```

```
int amarillo2 = 33;
```

```
int rojo2 = 32;
```

```
int vd2 = 18;
```

```
int rd2 = 19;
```

Anexo 2 – Código de programación (tiempo de los eventos)

```
// Tiempos de duración para los 4 eventos
```

```
int t1= 27000;
```

```
int t2 = 3000;
```

```
int t3 = 27000;
```

```
int t4 = 3000;
```

Anexo 3 – Código de programación (secuencia normal)

```
// Secuencia normal sin existencia de pulso de sincronismo
```

```
Serial.println("Semáforo 1");// Semáforo 1
```

```
digitalWrite(verde1, HIGH);
```

```
digitalWrite(amarillo1, LOW);
```

```
digitalWrite(rojo1, LOW);
```

```
digitalWrite(vd1, HIGH);
```

```
digitalWrite(rd1, LOW);
```

```
Serial.println("Semáforo 2");// Semáforo 2
```

```
digitalWrite(verde2, LOW);
```

```
digitalWrite(amarillo2, LOW);
```

```
digitalWrite(rojo2, HIGH);
```

```
digitalWrite(vd2, LOW);  
digitalWrite(rd2, HIGH);  
vTaskDelay(t1 / portTICK_RATE_MS);
```

```
digitalWrite(verde1, LOW);// Semáforo 1  
digitalWrite(amarillo1, HIGH);  
digitalWrite(rojo1, LOW);  
digitalWrite(vd1, HIGH);  
digitalWrite(rd1, LOW);  
digitalWrite(verde2, LOW);// Semáforo 2  
digitalWrite(amarillo2, LOW);  
digitalWrite(rojo2, HIGH);  
digitalWrite(vd2, LOW);  
digitalWrite(rd2, HIGH);  
vTaskDelay(t2 / portTICK_RATE_MS);
```

```
digitalWrite(verde1, LOW);// Semáforo 1  
digitalWrite(amarillo1, LOW);  
digitalWrite(rojo1, HIGH);  
digitalWrite(vd1, LOW);  
digitalWrite(rd1, HIGH);  
digitalWrite(verde2, HIGH);// Semáforo 2  
digitalWrite(amarillo2, LOW);  
digitalWrite(rojo2, LOW);  
digitalWrite(vd2, HIGH);  
digitalWrite(rd2, LOW);  
vTaskDelay(t3 / portTICK_RATE_MS);
```

```
digitalWrite(verde1, LOW);// Semáforo 1  
digitalWrite(amarillo1, LOW);  
digitalWrite(rojo1, HIGH);  
digitalWrite(vd1, LOW);  
digitalWrite(rd1, HIGH);
```

```

digitalWrite(verde2, LOW);// Semáforo 2
digitalWrite(amarillo2, HIGH);
digitalWrite(rojo2, LOW);
digitalWrite(vd2, HIGH);
digitalWrite(rd2, LOW);
vTaskDelay(t4 / portTICK_RATE_MS);

```

Anexo 4 – Código de programación (recepción de pulso externo)

```

// Lectura del pulso de sincronismo
int u=digitalRead(15);// Lee el pin digital 15

// Cuando el GPIO 15 recibe un 1 lógico, cnt cambia a 1 y se registra el tiempo que
// transcurrió hasta la llegada del pulso externo de sincronismo en seg
if (u==1){
    cnt=1;
    seg=reg;
}
// Se realiza el cálculo de tiempo que se adiciona a los eventos t1 y t3
int t=((seg*1000)/4)/2;

```

Anexo 5 – Código de programación (proceso de sincronización)

```

// Calcula el tiempo que falta para que termine el ciclo en el que llegó el pulso
vTaskDelay((60000-(seg*1000)) / portTICK_RATE_MS);
// Detiene el funcionamiento de la secuencia normal
vTaskSuspend(Suspend1);
Serial.println("Sincronizar con pulso");
// Se ejecuta la secuencia de sincronización con el tiempo adicional
// durante 4 ciclos
while(1){
    while(n<=3){
        Serial.println("Iniciar secuencia pulso");// Se imprime una vez por ciclo
        // Se muestra el nuevo tiempo para los eventos t1 y t3

```

```
Serial.println(t1+t);
Serial.println(t3+t);
digitalWrite(verde1, HIGH);// Semáforo 1
digitalWrite(amarillo1, LOW);
digitalWrite(rojo1, LOW);
digitalWrite(vd1, HIGH);
digitalWrite(rd1, LOW);
digitalWrite(verde2, LOW);// Semáforo 2
digitalWrite(amarillo2, LOW);
digitalWrite(rojo2, HIGH);
digitalWrite(vd2, LOW);
digitalWrite(rd2, HIGH);
vTaskDelay((t1+t) / portTICK_RATE_MS);
```

```
digitalWrite(verde1, LOW);// Semáforo 1
digitalWrite(amarillo1, HIGH);
digitalWrite(rojo1, LOW);
digitalWrite(vd1, HIGH);
digitalWrite(rd1, LOW);
digitalWrite(verde2, LOW);// Semáforo 2
digitalWrite(amarillo2, LOW);
digitalWrite(rojo2, HIGH);
digitalWrite(vd2, LOW);
digitalWrite(rd2, HIGH);
vTaskDelay(t2 / portTICK_RATE_MS);
```

```
digitalWrite(verde1, LOW);// Semáforo 1
digitalWrite(amarillo1, LOW);
digitalWrite(rojo1, HIGH);
digitalWrite(vd1, LOW);
digitalWrite(rd1, HIGH);
digitalWrite(verde2, HIGH);// Semáforo 2
digitalWrite(amarillo2, LOW);
```

```

digitalWrite(rojo2, LOW);
digitalWrite(vd2, HIGH);
digitalWrite(rd2, LOW);
vTaskDelay((t3+t) / portTICK_RATE_MS);

digitalWrite(verde1, LOW);// Semáforo 1
digitalWrite(amarillo1, LOW);
digitalWrite(rojo1, HIGH);
digitalWrite(vd1, LOW);
digitalWrite(rd1, HIGH);
digitalWrite(verde2, LOW);// Semáforo 2
digitalWrite(amarillo2, HIGH);
digitalWrite(rojo2, LOW);
digitalWrite(vd2, HIGH);
digitalWrite(rd2, LOW);
vTaskDelay(t4 / portTICK_RATE_MS);
n++;

```

Anexo 6 – Código de programación (retoma secuencia normal)

```

// Se imprime cuando termina la sincronización
Serial.println("Retoma secuencia normal");
// Se readuna la secuencia normal
sec();

```

Anexo 7 – Pruebas con pulso de llegada a los 14 sg

06	01:32.29	00:02.96
05	01:29.33	00:28.80
04	01:00.53	00:02.89
03	00:57.64	00:27.12
02	00:30.52	00:03.01
01	00:27.51	00:27.51

Ilustración A7.1: Captura de los eventos parte 1

12	03:07.74	00:03.16
11	03:04.58	00:28.77
10	02:35.81	00:03.10
09	02:32.71	00:28.70
08	02:04.01	00:02.82
07	02:01.19	00:28.90

Ilustración A7.2: Captura de los eventos parte 2

18	04:42.70	00:02.87
17	04:39.83	00:28.79
16	04:11.04	00:02.91
15	04:08.13	00:28.64
14	03:39.49	00:03.23
13	03:36.26	00:28.52

Ilustración A7.3: Captura de los eventos parte 3

24	06:14.58	00:02.93
23	06:11.65	00:26.65
22	05:45.00	00:03.58
21	05:41.42	00:26.55
20	05:14.87	00:03.25
19	05:11.62	00:28.92

Ilustración A7.4: Captura de los eventos parte 4

Anexo 8 – Pruebas con pulso de llegada a los 35 sg

06	01:34.82	00:02.87
05	01:31.95	00:31.99
04	00:59.96	00:02.68
03	00:57.28	00:27.13
02	00:30.15	00:02.86
01	00:27.29	00:27.29

Ilustración A8.1: Captura de los eventos parte 1

12	03:17.96	00:02.89
11	03:15.07	00:31.42
10	02:43.65	00:03.14
09	02:40.51	00:31.20
08	02:09.31	00:03.00
07	02:06.31	00:31.49

Ilustración A8.2: Captura de los eventos parte 2

18	05:01.08	00:02.96
17	04:58.12	00:31.41
16	04:26.71	00:02.97
15	04:23.74	00:31.40
14	03:52.34	00:02.98
13	03:49.36	00:31.40

Ilustración A8.3: Captura de los eventos parte 3

24	06:35.40	00:02.78
23	06:32.62	00:27.18
22	06:05.44	00:02.96
21	06:02.48	00:27.05
20	05:35.43	00:02.97
19	05:32.46	00:31.38

Ilustración A8.4: Captura de los eventos parte 4

Anexo 9 – Pruebas con pulso de llegada a los 28 sg

06	01:30.43	00:02.98
05	01:27.45	00:27.30
04	01:00.15	00:03.09
03	00:57.06	00:26.65
02	00:30.41	00:03.31
01	00:27.10	00:27.10

Ilustración A9.1: Captura de los eventos parte 1

12	03:07.44	00:02.90
11	03:04.54	00:30.62
10	02:33.92	00:03.06
09	02:30.86	00:30.42
08	02:00.44	00:02.96
07	01:57.48	00:27.05

Ilustración A9.2: Captura de los eventos parte 2

18	04:47.92	00:03.05
17	04:44.87	00:30.36
16	04:14.51	00:02.93
15	04:11.58	00:30.69
14	03:40.89	00:02.98
13	03:37.91	00:30.47

Ilustración A9.3: Captura de los eventos parte 3

24	06:28.41	00:03.05
23	06:25.36	00:30.44
22	05:54.92	00:03.04
21	05:51.88	00:30.46
20	05:21.42	00:02.58
19	05:18.84	00:30.92

Ilustración A9.4: Captura de los eventos parte 4

28	07:28.36	00:02.99
27	07:25.37	00:27.09
26	06:58.28	00:02.70
25	06:55.58	00:27.17

Ilustración A9.5: Captura de los eventos parte 5