UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

Sistema para conversión de semáforos convencionales en semáforos para no videntes

Autor:
Ing. Sebastián Alejandro Suárez

Director:
Esp. Ing. Sergio Renato de Jesús Meleán

Jurados:
Esp. Ing. Franco Bucafusco (FIUBA)
Esp. Ing. Diego Fernandez (FIUBA)
Esp. Ing. Marcelo E. Romeo (UNSAM, UTN-FRBA)

Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre octubre 2018 y agosto de 2019.

Resumen

En esta memoria se describe el diseño e implementación de un sistema para la inclusión de personas no videntes o con disminución visual, el cual advierte los cambios de luces de un semáforo vehicular a través de una red inalámbrica por medio de las vibraciones de un teléfono inteligente.

Para la elaboración del proyecto se aplicaron los conocimientos adquiridos a lo largo de la Carrera de Especialización en Sistemas Embebidos tales como la programación de microcontroladores, conceptos de sistemas operativos de tiempo real, procesamiento de eventos asíncronos, protocolos de comunicación y diseño de circuitos impresos.

Este trabajo fue desarrollado para la empresa Adox la cual está interesada en la creación de este equipo, siendo el código y hardware público a la comunidad.

Agradecimientos

A mi esposa Mary por su compañía, paciencia, consejo y apoyo incondicional a lo largo de toda la especializacion.

A mis padres José y Graciela que me forjaron como la persona que soy y me alentaron en el estudio.

A mi director Sergio por su tiempo y sabiduria en todo momento.

A Roberto Zelarayán por prestarme la EDU-CIAA.

A los profesores de la Carrera de Especialización en Sistemas Embebidos que me acompañaron en esta formación profesional.

Índice general

Índice de figuras

Índice de Tablas

Dedicado a... [OPCIONAL]

Capítulo 1

Introducción General

Este capítulo introduce al lector al tema abordado en este trabajo, su propósito y su alcance.

1.1. Motivacion

En Argentina 1 de cada 10 personas poseen algún tipo de discapacidad, siendo la que más prevalece la discapacidad motora, seguida por la visual, la auditiva y la mental [6].

En este marco nos enfocamos en las personas con discapacidad visual, las cuales se enfrentan diariamente con el desafío de calcular distancias en un lugar determinado, con el fin de evitar accidentes. El uso del bastón es el elemento empleado a la hora de desplazarse para evitar dicha problemática. Bajo este contexto el mayor inconveniente surge en la vía publica al querer cruzar una calle. Esto se debe a que en general no tienen ningún tipo de señal que les indique si está libre de autos circulando o si se produjo el cambio de semáforo. Para estas situaciones el bastón no es de utilidad, ya que no brinda ningún tipo de información. Si bien en algunas oportunidades pueden valerse de la buena intención de algún transeúnte es importante para toda persona poder valerse por sí misma y no depender de un tercero.

1.2. Objetivos

El objetivo de este proyecto fue desarrollar un prototipo abierto, autónomo y económico, que permita ser conectado a cualquier semáforo convencional, aprendiendo su comportamiento. Tomando como entradas la secuencia de luces del semáforo y como salida advierta a las personas con discapacidades visuales cuando puede o no cruzar la calle sin peligro. Esta advertencia se realiza por medio de de una señal sonora y/o vibraciones de su teléfono inteligente el cual se conecta a una red wifi [2] provista por el sistema.

Dicho prototipo fue realizado de manera abierta, es decir, que su código y hardware están disponibles para todas las personas que lo deseen.

1.3. Alcance

El desarrollo del presente proyecto incluyó:

- Desarrollo de un prototipo funcional.
- Desarrollo de una aplicación en android.
- Desarrollo de un protocolo para lograr la escalabilidad en las formas de comunicación con dispositivos de advertencia.
- Ajuste de nivel de sonido automáticamente según ruido ambiente.
- Posibilidad de activar el sistema por medio de un comando a distancia.

Capítulo 2

Introducción Específica

Este capítulo provee una introducción más detallada de todo el trabajo realizado. Se presenta al lector una explicación del funcionamiento de los semáforos, una vista general del sistema, requerimientos y una explicación de las tecnologías involucradas en el desarrollo.

2.1. Funcionamiento de los semaforos

Los semáforos, también conocidos técnicamente como señales de control de tráfico, son dispositivos de señales que se sitúan en intersecciones viales y otros lugares para regular el tráfico, y por ende, el tránsito peatonal [9].

Los semáforos se dividen en tres clases, que son:

- Vehicular: Tiene por objeto regular el tránsito de vehículos en las intersecciones. Está compuesto esencialmente por tres faros programados para que proyecten durante un tiempo determinado.
- Peatonal: Se hallan instalados en combinación con los vehiculares y tienen por objeto regular el paso de los peatones en intersecciones con alto volumen de tráfico.
- Direccional: Tiene como fin informar mediante flechas, el momento adecuado para girar.

En cuanto al funcionamiento del semáforo vehicular se puede decir que, cuando la luz es verde, significa que hay vía libre y se puede pasar. La luz amarilla advierte al conductor que se aproxima un cambio de luz. Al ver la luz roja se debe detener el auto, pues otro flujo de vehículos se interceptará en la dirección de su marcha.

En los semáforos peatonales, el significado es el siguiente: la silueta roja indica que el peatón no debe cruzar la calle, mientras que la silueta verde o blanca lo permite.

En los semáforos direccionales, la flecha roja prohíbe el giro, y la verde autoriza el cruce en ese sentido.

2.1.1. Secuencia

En base a la observación del funcionamiento de los semáforos vehiculares en distintas provincias de la República Argentina como ser Córdoba, Catamarca y Tucumán se detectaron las siguientes secuencias:

- Rojo, rojo-amarillo , verde, amarillo y rojo
- Rojo, amarillo, verde, amarillo y rojo
- Rojo, amarillo, verde y rojo

2.2. Esquema general del sistema

Se observa en la siguiente figura 2.1 una version simplificada del sistema.

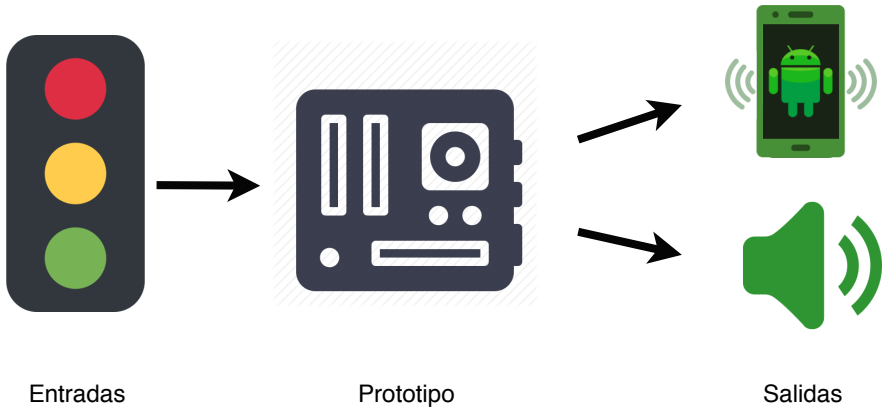

FIGURA 2.1: Esquema general del sistema.

Se toman como entradas las luces del semáforo, que se encuentran conectadas al prototipo a través de interfaces que detectan la tensión presente en los focos. Esta información se procesa y según el caso se envían las señales correspondientes. El sistema puede advertir a una persona con discapacidad visual de dos maneras diferentes:

1. Por medio del módulo Wi-Fi hacia una aplicación instalada previamente en su teléfono inteligente.
2. Mediante una bocina que emite una señal sonora. Para esto el prototipo toma el ruido ambiente y limita la intensidad de esta señal de manera directamente proporcional al ruido presente.

2.3. Requerimientos

De acuerdo a la necesidad que se necesita satisfacer y a lo expuesto, se encontraron los siguientes requerimientos:

R1. Hardware:

R1.1. Operar con cargas de entradas de 220 V, 50 Hz y 5 A.

R1.2. El hardware involucrado en la detección del cambio de luces debe estar totalmente aislado del módulo principal.

R2. Comunicación:

R2.1. El sistema debe proveer un red Wi-Fi que cumpla con las normas IEEE 802.11.

R2.2. Se debe proveer una señal sonora, su intensidad debe poder variar automáticamente dependiendo del ruido presente.

R3. Software embebido:

R3.1. El sistema deberá ser capaz de aprender la secuencia de cambio de luces.

R3.2. El sistema deberá ser capaz de detectar el semáforo fuera de servicio.

R3.3. Se deberá implementar un protocolo para la fácil escalabilidad de los distintos tipos de comunicación.

R3.4. El sistema deberá implementarse en base a un sistema operativo de tiempo real.

R4. Metodología de desarrollo:

R4.1. Se utilizará GIT como herramienta de control de versiones.

R4.2. Se utilizará Doxygen como herramienta para generar la documentación.

R4.3. Se realizarán pruebas unitarias para cada módulo.

R5. Aplicación móvil:

R5.1. Conectarse automáticamente a la red wifi que provee el sistema.

R5.2. Definir un protocolo de vibraciones según los mensajes del sistema.

2.4. Propuesta de implementación

Para abordar una solución integral a la problemática planteada y satisfacer los requerimientos enunciados, se propuso un sistema con componentes de hardware y software los cuales se describen de la siguiente manera.

Para el software se utilizaron las siguientes tecnologías:

- FreeRTOS como sistema operativo de tiempo real.
- sAPI como framework para manejo de periféricos [8].
- Colas para mensajes entre tareas.
- Eventos asincronicos.

En cuanto al hardware, los elementos se dividieron en tres secciones:

- Elementos de entrada.
- Plataforma de desarrollo.

- Elementos de salida.

2.4.1. Entradas

Detección de tension

Para censar si hay tensión presente en algún foco del semáforo vehicular, se utilizaron módulos de aislamiento con opto-acopladores para corriente alterna 220 V figura 2.2 en cada luz.

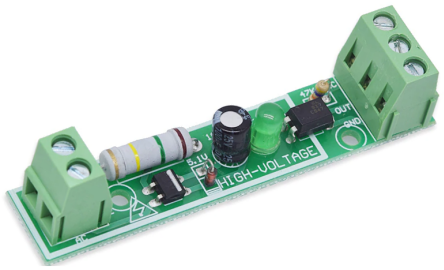

FIGURA 2.2: Módulo de aislamiento opto-acoplador de 1 bit

Detección de ruido ambiente

Para la detección del nivel de ruido ambiente presente se utilizo un modulo que tiene un micrófono y un circuito amplificador con un integrado LM393 figura 2.3.

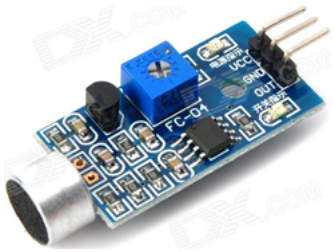

FIGURA 2.3: Módulo para la detección de sonido

2.4.2. Plataforma de desarrollo

Se optó por la EDU-CIAA-NXP figura 2.4 como base para el desarrollo del proyecto por ser una plataforma ya conocida, de amplia disponibilidad, económica y de hardware abierto. Dicha placa posee un microcontrolador LPC4337 de la empresa NXP, el cual cuenta con dos núcleos ARM Cortex-M; un Cortex-M4 y un Cortex-M0.

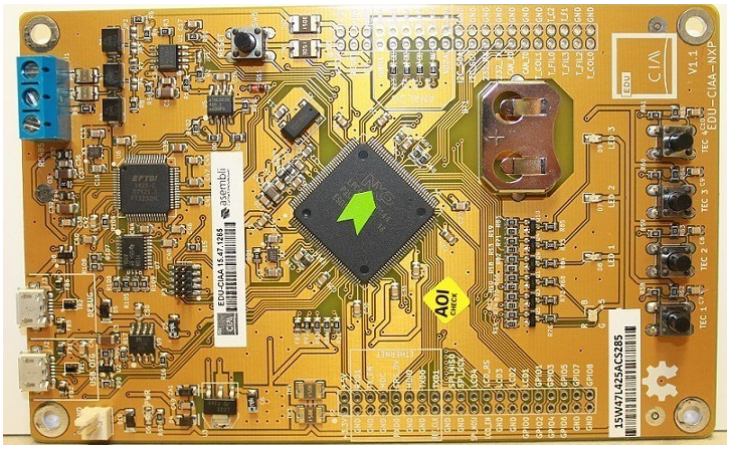

FIGURA 2.4: La EDU-CIAA posee un microcontrolador LPC4337 (dual core ARM Cortex-M4 y Cortex-M0).

2.4.3. Salidas

Modulo wifi

El ESP01[1] es un modulo que contiene un microcontrolador ESP8266[5] de bajo coste con Wi-Fi integrado fabricado por la empresa Espressif[3] figura 2.5. En cuanto a comunicación Wi-Fi, este modulo tiene comunicación integrada 802.11 b/g/n, incluidos modos Wi-Fi Direct (P2P)[11] y softAp[10]. Incluye una pila de TCP/IP completa, lo que libera de la mayor parte del trabajo de comunicación al procesador.

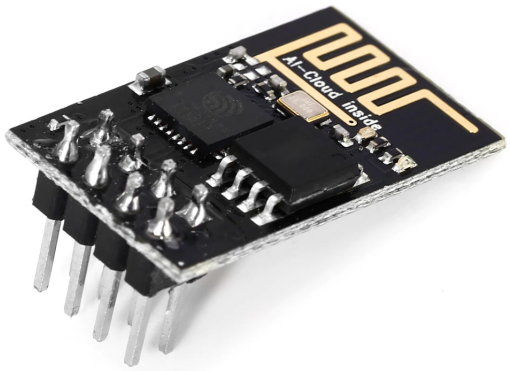

FIGURA 2.5: ESP8266 (ESP-01) 2.4GHz Módulo Inalámbrico.

Modulo generador de sonido

El módulo fue desarrollado específicamente para este proyecto en base a diagramas de la hoja de datos del fabricante figura 2.6 para el amplificador operacional LM386[7], el cual provee un amplificador de audio de baja potencia, capaz de funcionar con una fuente de alimentación simple entre 4 y 12 V. Este módulo genera una señal sonora a través de un parlante de $8\,\Omega$.

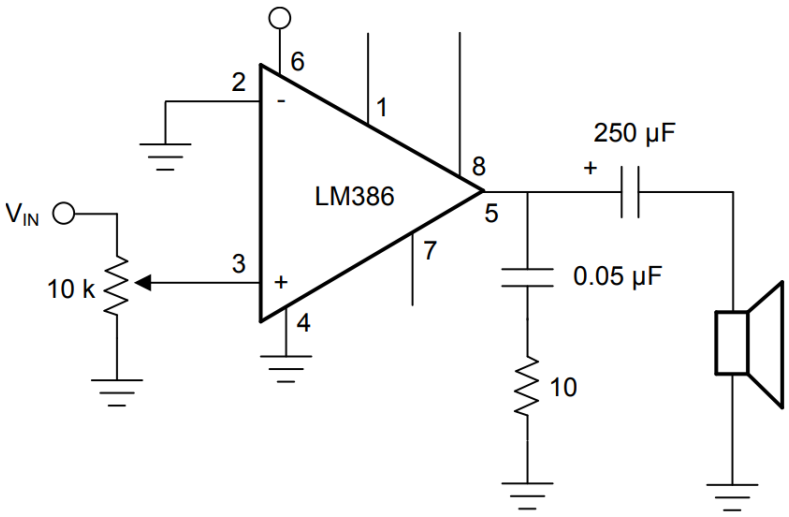

FIGURA 2.6: Circuito amplificador basado en el amplificador operacional LM386.

Capítulo 3

Diseño e Implementación

En este capítulo se presenta la arquitectura del *firmware* y el patrón de diseño usado para los módulos del sistema. Asimismo, se detallan aspectos funcionales de cada módulo y se fundamentan las elecciones de los distintos componentes de hardware utilizados.

El código fuente asociado puede consultarse en la siguiente URL: https://github.com/alesuarez/soniforo

3.1. Hardware

3.1.1. Arquitectura

Como se menciono en el capitulo anterior el sistema utiliza distintos módulos para poder manejar eventos externos y reaccionar en consecuencia, en la figura 3.1 se puede observar los módulos y su tipo de conexión a la placa principal EDU-CIAA.

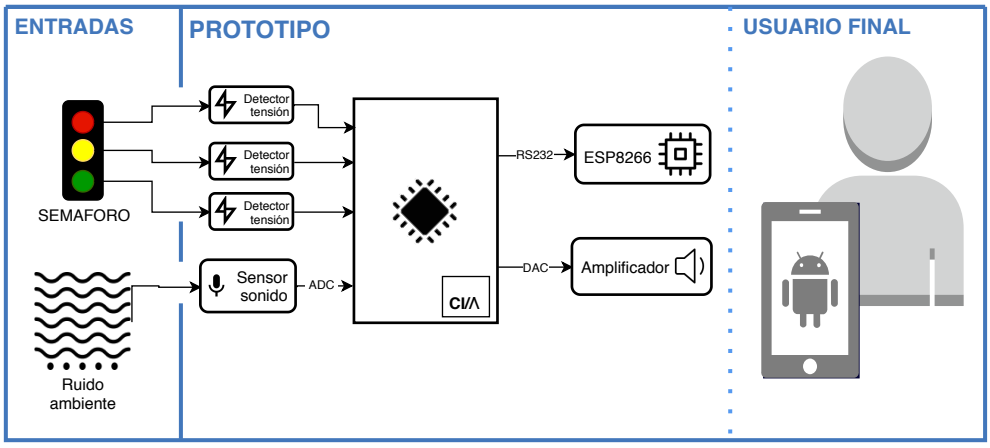

FIGURA 3.1: Diagrama general del hardware.

De esta manera se lista los módulos con su respectiva interfaces utilizada:

- Detectores de tension: Estos módulos se conectan por los puertos GPIO de la placa.
- Sensor de sonido: Se conecta por el puerto analógico digital de la placa.

- Amplificador (Salida al parlante): Utiliza el puerto de conversión digital-analógico.
- ESP8266: utilizar el otro puerto RS232 que posee la EDU-CIAA.

3.2. Firmware

3.2.1. Arquitectura

A este proyecto se le dió el nombre de Soniforo, de la conjunción da las palabras sonido y semáforo.

Se optó por un modelo de capas jerárquico para organizar el código en los distintos niveles de abstracción.

En la figura 3.2 se puede observar como se planteó este trabajo.

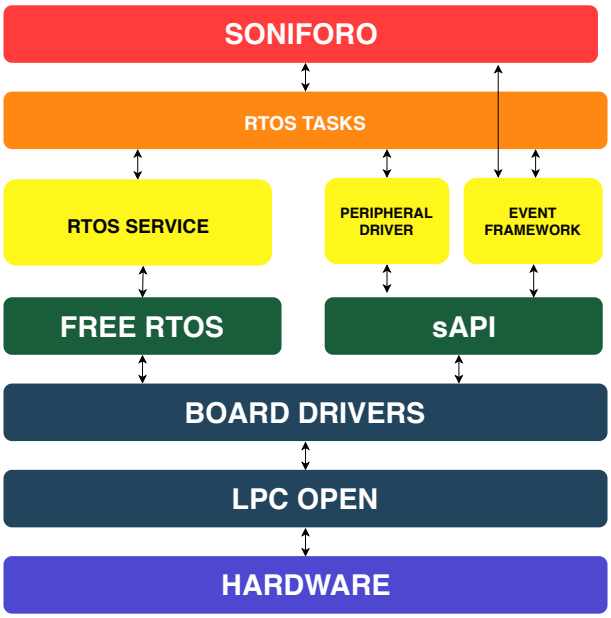

FIGURA 3.2: Estructura de capas para el *firmware*.

En cuanto a los niveles de abstracción, la primera capa constituye el hardware de la plataforma EDU-CIAA. La segunda capa, *Hardware Abstraction Layer* (HAL), permite desacoplar las capas superiores del hardware. Aquí se encuentran los drivers del fabricante del microcontrolador, en el bloque funcional LPC Open. La capa Hardware Independent Layer (HIL) incluye los módulos del RTOS (no está asociada a ningún hardware en particular) y sAPI que permite el manejo de los periféricos de la placa con mayor facilidad. Sobre esta base se observa una capa orientada a servicio que procesar la información del sistema, estas son:

- RTOS Service: Se encarga de la creación de tareas, temporizadores, semáforos binarios.
- Peripheral Driver: Se encarga de la configuración de los periféricos, como ser las interrupciones.

- Envent Framework: esta capa se encarga de la definición y creación de todos los eventos del sistema.

Sobre estas capas se ubican todas las tareas que corren y son propias del sistema, a esta se la llama RTOS Task. Finalmente, la última capa contiene la aplicación Soniforo.

3.2.2. Diseño del firmware

Para modelar la solución se utilizo una maquina de estado finito, también conocidas como MEF, esta se puede definir como un modelo computacional que realiza cómputos en forma automática sobre una entrada para producir una salida.

Este modelo está conformado por un alfabeto, un conjunto de estados finito, una función de transición, un estado inicial y un conjunto de estados finales.

De esta manera se observa en la siguiente figura 3.3 la maquina de estado finito perteneciente al sistema, por simplicidad algunos estados fueron eliminados para no dificultar el entendimiento (?).

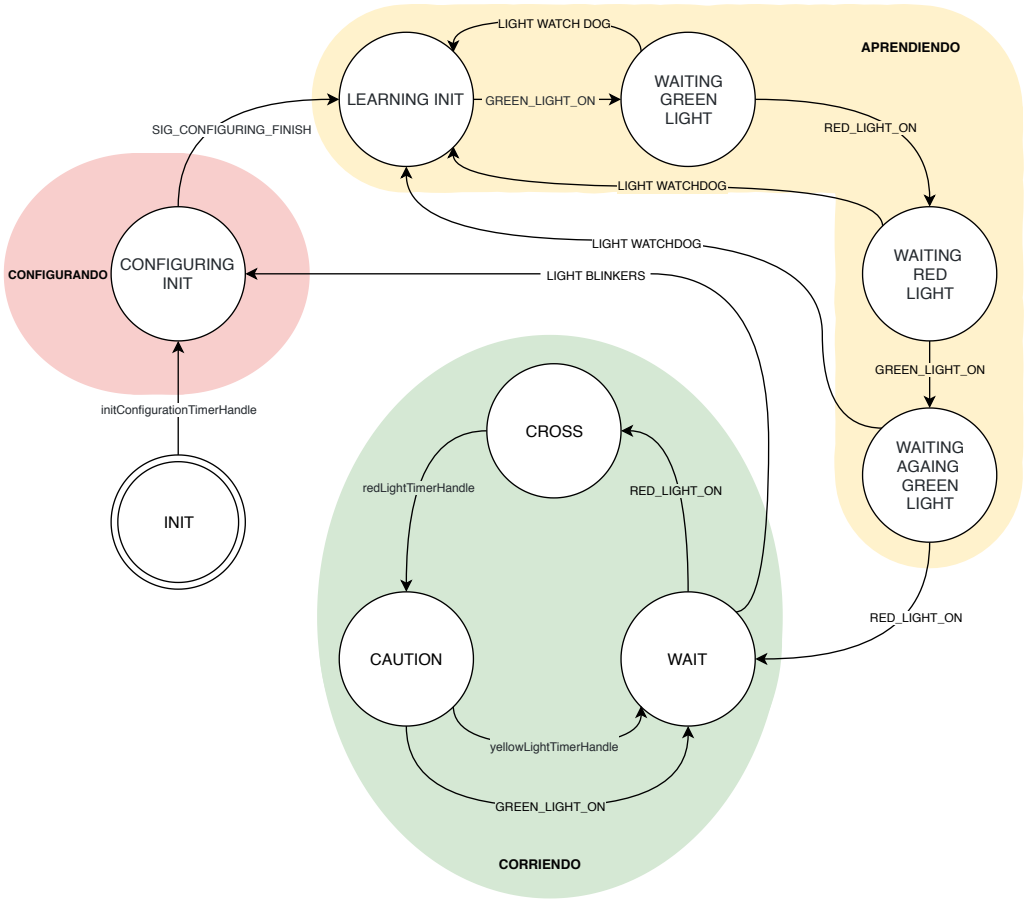

FIGURA 3.3: Maquina de estado finito del sistema.

De una manera general el sistema puede encontrarse en 3 estados:

- Configuracion: Donde se inicializan todos los dispositivos de advertencia. Se denomina dispositivo de advertencia a todos los módulos presentes en el

sistema que envía mensajes de manera tal que la persona no vidente puede conocer el estado del semáforo.

- Aprendizaje: El sistema usa el prendido y apagado de las luces del semáforo y con ello aprende su funcionamiento tomando los tiempos de las luces.
- Corriendo: Una vez finalizados los dos estados anteriores correctamente, el sistema comienza a funcionar de manera autónoma, siempre y cuando no detecte anomalías en los cambios de las luces del semáforo.

Configuración

Al energizarse el sistema o luego de un reinicio, el este inicia en el estado INIT donde comienza un temporizador (initConfigurationTimerHandle) que al momento de expirar envía un mensaje para iniciar el sistema, en este momento el sistema pasa a estado CONFIGUING_INIT este estado consiste en configurar todos los dispositivos de advertencia conectados a la placa principal. En el caso particular para la ESP8266 dicha configuración se la hace a traves de comandos AT, lo que se quiere lograr con la esto es que el modulo Wi-Fi levante un punto de acceso inalámbrico, de forma tal que, cualquier teléfono inteligente pueda conectarse a la red inalámbrica propiamente dicha. Tambien una vez configurado correctamente el punto de acceso, se abre una conexion UDP en el puerto 4096 de manera tal que se pueda enviar mensajes broadcast por dicho puerto. (todo: mejorar esta parte) A continuación se enumeran los comando que debe enviar la EDU-CIAA a traves del puerto RS232 hacia la ESP8266 para poder configurar lo nombrado anteriormente de manera correcta:

1. AT
2. AT+RST
3. AT+CWMODE=2
4. AT+CWSAP=“Soniforo_CIAA”,“”,8,0
5. AT+CWDHCP=0,1
6. AT+CWDHCP=1,1
7. AT+CIPSTART=3,“UDP”,“0”,0,4096,2
8. AT+CIPSEND=3,8,“192.168.4.255”,4096
9. AT+CIPMUX=1
10. AT+CIPDINFO=1
11. AT+CWAUTOCONN=0

Para mas detalle de que hace cada comando el lector puede recurrir a la hoja con comandos que provee el fabricante [4].

Cada comando posee una respuesta conocida es por esto que por cada uno enviado se espera la respuesta previamente definida, si esto no sucede después de tres intentos se envía la señal SIG_FAIL_CONFIG de manera que el sistema vuelve al estado INIT.

Aprendizaje

Lo que busca el sistema es tiempo que el peatón pueda cruzar la calle y que tiempo no debe cruzar la calle, esto es los tiempos de las luces tanto roja como verde respectivamente, cuando la luz verde del semaforo vehicular se prende signica que el peaton no puede cruzar la calle ya que se encuentran los vehículos en marcha pero cuando la luz cambia a rojo los vehiculos estan detenidos y el peatón puede cruzar la calle.

Como no sabemos a priori en que momento del ciclo el sistema se va a poner en funcionamiento, ni tampoco conocemos que tiempo puede tardar la configuración de los dispositivos de advertencias, el proceso de aprendizaje se hace a traves del encendido u apagado de las luces roja y verde.

Una vez que los dispositivos de advertencia están configurados de manera correcta el sistema pasa al estado LEARNING_INIT e inmendiatamente se envia al sistema al estado WAITING_GREEN_LIGHT de manera tal que cuando se detecte una luz verde tome el tiempo de inicio, envia el mensjae de poner al sistema en estado WAITING_RED_LIGHT cuando llega la luz roja se detiene el tiempo de la luz verde e inicializa el tiempo para la luz roja mientras espera nueva mente la luz ver, es por esto que se envia una mensaje para pasar al sistema al estado WAITING_AGAIN_GREEN_LIGHT.

Graficamente podemos ejemplificar la secuencia con un ejemplo.

Se puede observar en la figura 3.4 para el caso donde la secuencia de las luces son rojo, rojo-amarillo, verde, amarillo, rojo. Tomando como base que el sistema termino de configurarse correctamente cuando el semáforo estaba en luz roja.

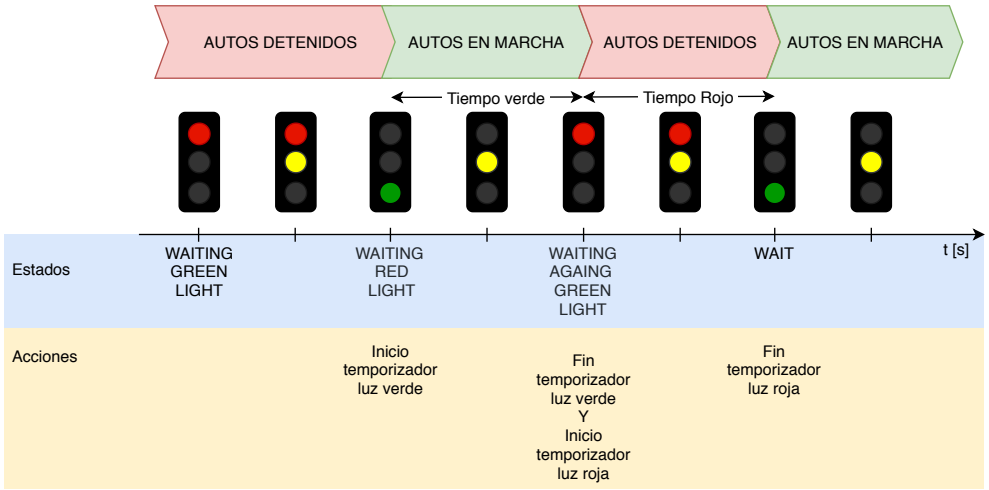

FIGURA 3.4: Maquina de estado finito del sistema.

Si por alguna razon se detecta alguna anomalia, como ser un semaforo fuera de servicio el sistema regresa al estado LEARNING INIT. Esto se logra atravez de un temporizador que debe actualizarce cada 65 segundos, si esto no acurre se ejecuta la accion definida anteriormente.

Corriendo

Una vez finalizado correctamente el aprendizaje de los tiempos, el sistema calcula el tiempo limite para indicar a la persona que puede cruzar la calle tranquilamente, esto se hace a traves de la sigueinte formula 3.1

$$yellowLightTimerHandle = tiempoLuzRojo * 0{,}35 \tag{3.1}$$

De igual manera se hace el calculo para el temporizador de luz roja formula 3.2

$$redLightTimerHandle = tiempoLuzRojo * 0{,}35 \tag{3.2}$$

Estos porcentajes fueron calculados de manera empirica. (decir algo mas) Estos temporizadores son gestionado por el FreeRTOS de manera tal que cuando expiren llamen a funciones predefinidas en el momento de la creacion de estos.

El sistema posee tres temporizadores los cuales son iniciados cuando se detecta una luz encendida.

- redLightTimerHandle: tiempo de la luz roja encendida.
- yellowLightTimerHandle: se mide en base a la luz roja como se menciono anteriormente.
- greenLightTimerHandle: tiempo de la luz verde.

Una vez configurados los temporizadores, se espera una interrupción de la luz roja para desbloquear la tarea de enviar el estado CROSS a la ESP8266 y ademas empieza a correr el temporizador de la luz roja, cuando este expira se bloquea la tarea de enviar el estado de CROSS y se desbloquea la tarea de enviar el estado CAUTION y se incializa el temporizador de la luz amarilla, cabe recordar como se mensiono que los mensajes broadcast UDP a traves del puerto 4096. Una vez que vence este temporizador se bloque la tarea de enviar el estado CROSS y se espera nuevamente por la señal de la luz roja encendida.

3.3. Principio de funcionamiento

El firmware fue diseñado en base a distintos patrones vistos a lo largo de la carrera, principalmente el sistema se basa en el diseño Observar y reaccionar, este patron nos permite reaccionar a distintos eventos y reaccionar de una manera predefinida en conjunto con este patron el sistema implementa Lazos de eventos. Donde las reacciones produce mensajes y segun el estado en el cual este el sistema dispara alguna accion. Donde en nuestro caso particular estas reacciones generan transiciones de estados. En la figura 3.5 se puede observar el flujo de trabajo del sistema, donde los eventos que produce el semaforo se envian a una cola que segun el tipo de evento y el estado del sistema reacciona de una manera definida. Es por eso que se define un tipo de dato especial para enviar el evento hacia una cola.

```
1 typedef struct  {
2   led_name_t Led;
3   led_status_t Status;
```

```
4 } message_t;
```

Los tipos de estados de las luces pueden ser:

- Luz encendida
- Luz Apagada

El flujo por ejemplo si se prendiera la luz roja seria:

1.
2.
3.

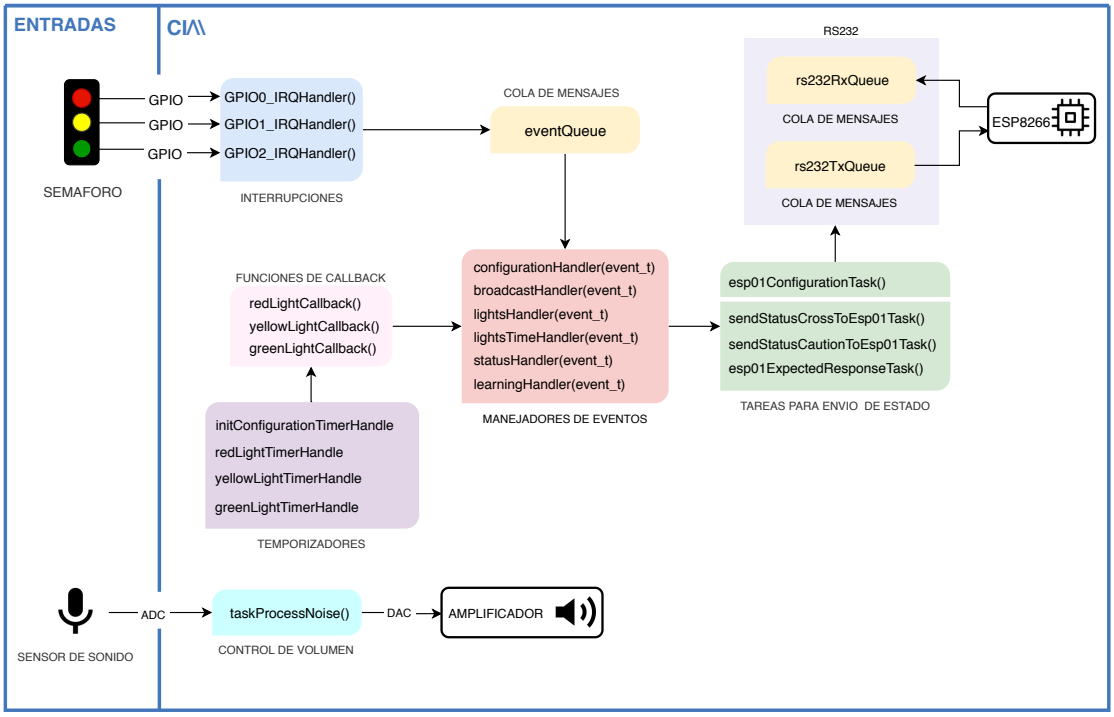

FIGURA 3.5: Diagrama general de funcionamiento.

Bibliografía

[1] Esp01. *Hoja de datos Esp01*. Disponible: 2019-07-31. URL: http://www.microchip.ua/wireless/esp01.pdf.

[2] Real academia española. *Definicion de wifi*. Disponible: 2019-07-13. URL: https://dle.rae.es/?id=c6ehZd8.

[3] Espressif. *Espressif Systems*. Disponible: 2019-07-31. URL: https://www.espressif.com/.

[4] Espressif. *Hoja de comandos ESP8266*. Disponible: 2019-08-07. URL: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf.

[5] Espressif. *Hoja de datos ESP8266*. Disponible: 2019-07-31. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

[6] Indec. *Estudio Nacional sobre el Perfil de las Personas con Discapacidad*. Disponible: 2019-07-13. URL: https://www.indec.gob.ar/ftp/cuadros/poblacion/estudio_discapacidad_12_18.pdf.

[7] Texas Instrument. *Hoja de datos LM386*. Disponible: 2019-07-31. URL: http://www.ti.com/lit/ds/symlink/lm386.pdf.

[8] sAPI. *simpleAPI*. Disponible: 2019-07-20. URL: https://github.com/epernia/sAPI.

[9] Wikipedia. *Semáforo*. Disponible: 2019-07-11. URL: https://es.wikipedia.org/wiki/Sem%C3%A1foro.

[10] Wikipedia. *SoftAP*. Disponible: 2019-07-31. URL: https://es.wikipedia.org/wiki/SoftAP.

[11] Wikipedia. *Wi-Fi Direct*. Disponible: 2019-07-31. URL: https://es.wikipedia.org/wiki/Wi-Fi_Direct.