



UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**Prof. Patrocinante:**

Sergio Sobarzo G, Ph.D.

# Sistema de localización Indoor mediante FTM en sistema ESP32-S2

**Carlos Arnoldo Castillo Candia**

Informe de Memoria de Título para optar al Título de  
Ingeniero Civil en Telecomunicaciones

Concepción, Chile.

17 de marzo de 2022

# Resumen

En la presente memoria de título se desarrolla un sistema de localización en interiores o localización indoor. En ésta, se hará incapie tanto las antiguas formas de detección que existen como a una nueva forma de detección, en la cual se debe de implementar el concepto de tiempo de vuelo en la medición de datos. Para la implementación de este sistema, se realizaron diferentes pruebas con el fin de determinar si esta nueva forma de medición es más precisa al momento de la detección que las formas ya existentes de localización en interiores, logrando resultados bastante prometedores en el ámbito de la localización. Además se implemento un sistema de bajo consumo para el funcionamiento del sistema, a través de dispositivos ESP, permitiendo así verificar que la localización indoor puede ser implementada de manera más económica en diferentes actividades a futuro.

# Abreviaciones

- FTM: Fine Time Measurement.
- RTT: Round-trip time.
- ToF: Time Of Flight.
- GPS: Global Positioning System.
- STA: Station.
- AP: Access Point.
- BLE: Bluetooth Low Energy.
- RFID: Radio Frecuency Identification.

# Índice General

<b>Resumen</b>	<b>I</b>
<b>Abreviaciones</b>	<b>I</b>
<b>Índice de Figuras</b>	<b>IV</b>
<b>Índice de Tablas</b>	<b>V</b>
<b>Agradecimientos</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del Problema . . . . .	2
1.2. Estado del Arte . . . . .	2
1.2.1. Trabajos previos . . . . .	3
1.3. Discusión . . . . .	5
1.4. Hipótesis de Trabajo . . . . .	6
1.5. Objetivos . . . . .	6
1.5.1. Objetivo General . . . . .	6
1.5.2. Objetivos Específicos . . . . .	6
1.6. Alcances y Limitaciones . . . . .	7
1.7. Metodología . . . . .	8
<b>2. Fundamentos teóricos</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Marco Teórico . . . . .	9

2.2.1. Localización en exteriores . . . . .	9
2.2.2. Detección de interiores . . . . .	10
2.2.3. Protocolo 802.11 mc . . . . .	12
2.2.4. Técnica de medición . . . . .	13
<b>3. Desarrollo</b>	<b>16</b>
3.1. Introducción . . . . .	16
3.1.1. Equipamiento de trabajo . . . . .	16
3.2. Código de ejemplo . . . . .	17
3.2.1. Funcionamiento del código . . . . .	18
3.3. Automatización del código . . . . .	20
<b>4. Resultados</b>	<b>21</b>
4.1. Introducción . . . . .	21
4.2. Lugar de medición . . . . .	21
4.3. Ejecución del código . . . . .	24
<b>5. Conclusiones</b>	<b>31</b>
5.1. Sumario . . . . .	31
5.2. Conclusiones . . . . .	31
5.3. Trabajo Futuro . . . . .	32
<b>Bibliografía</b>	<b>34</b>
<b>Anexo</b>	<b>62</b>

# Índice de Figuras

2.1. Concepto básico del WiFi RTT. . . . .	12
2.2. Esquema de trilateración. . . . .	14
3.1. ESPs2-32. . . . .	17
3.2. Antena externa. . . . .	17
3.3. Pantalla inicial del ejemplo de FTM. . . . .	18
3.4. Parte 1 del Ejemplo en funcionamiento. . . . .	19
3.5. Parte 2 del Ejemplo en funcionamiento. . . . .	19
3.6. Partes del sistema AP. . . . .	20
4.1. Imagen referencial del lugar de medición. . . . .	22
4.2. Imagen referencial del lugar de medición. . . . .	23
4.3. Detección de los AP. . . . .	24
4.4. Imagen ejemplo del código de trilateración. . . . .	27
4.5. Gráfica de las posiciones del STA utilizando el código. . . . .	27
4.6. Gráfica de las posiciones obtenidas por el código vs los valores reales. . . . .	29

# Índice de Tablas

4.1. Distancia entre el STA con cada AP. . . . .	25
4.2. Posición del STA a través de la técnica de trilateración. . . . .	26
4.3. Tabla de datos para la posición real. . . . .	28
4.4. Comparación código vs distancia real. . . . .	30

# Agradecimientos

Me gustaría dar las gracias a todas las personas que han estado durante este largo camino que fue la universidad, de verdad fue bastante largo, en el cual se pasaron momentos muy buenos y otros bien malos, pero bueno, así es la vida del universitario.

Me gustaría primero agradecer a las personas mas importantes para mi los cuales son mi familia y mi pareja, sin ellos ni siquiera hubiera podido llegar a este punto ya que lo que no tengo en confianza en mi mismo ellos la tienen en mi y eso fue mas que suficiente para poder llegar acá. Darle un agradecimientos es lo mínimo que puedo hacer a esta gente que amo tanto, mis viejos, Juan y Angélica, siempre han estado para mi en momentos buenos de mi carrera como en momentos en donde lo único que quería era dejar todo y no querer seguir, esto que he logrado hasta hora es mas de ellos que mio. Mi hermano Angel fue de los pocos que entendía lo bueno y lo malo de la universidad, tenerlo al lado para hablar y conversar fue algo que me salvo en mas de una ocasión en estos años. Mi hermanita Victoria fue algo que nadie esperaba que apareciera, pero fue capaz de darme la alegría suficiente de seguir adelante y tratar de ser un ejemplo a seguir para ella. Mis Abuelos, Berta y Carlos, son gente de campo que quizás no sepa de temas de la universidad, pero aun asi no dudaron nunca que lo lograría y eso es lo que vale al final del día. Finalmente, pero no menos importante, quiero darle las gracias a la persona que ha estado conmigo desde antes que entrara a la u, que ha estado tanto en las buenas como en los momentos de mierda, que cuando mas la necesitaba no dudaba en viajar para poder estar a mi lado y que en estos momentos sigue estando al lado mio sin importar las situaciones, gracias por todo Jossy, no sabes lo mucho que te amo y lo importante que has sido para mi en mi vida.



Si bien la gente antes mencionada es la mas importante para mi en mi vida, la gente que conocí durante estos años de universidad fueron lo mejor que tuvo la universidad, desde los cabros que por diferentes cosas ya no estan en la u, como el Anibal, Limache, El cachu, la pollo, Koala, Mati (BIO), Marcelo, coke, Mati (ELE). Con los que aun estamos terminando todo este largo camino: el Abel, Camilo, Paula, Juanito, Martin, Diego, Nacho y todos los telecom. No me queda mas que darle las gracias a todos por haberme soportado todos estos años, que si bien con algunos no fuimos tan grandes amigos, me alegro mucho el haberlos conocido y de pasar muy buenos momentos junto a ellos. Espero que terminando todo esto podamos hacer su recorriendo conce otra vez y pegarnos sus pingponetes en el tm como lo hacíamos antes

Finalmente darle las gracias a Felipe. Luciano y al profe Sobarzo por su ayuda en diferentes ámbitos dentro de la memoria.

Muchas gracias por todo a todos.

Carlos Castillo

# Capítulo 1

## Introducción

A lo largo de los años, la tecnología ha estado en una constante evolución con el fin de brindar a los usuarios las herramientas necesarias para solucionar los problemas que se presentan a lo largo de su vida. Uno de los grandes problemas es la ubicación, ya que si bien hay mapas que dan indicaciones detalladas de hacia dónde quiere ir el usuario, la explicación no siempre es clara debido a diferentes factores. Debido a esto una de las mejoras que tuvo la tecnología en este ámbito fue la de poder determinar la posición de un objeto en el exterior con una gran exactitud, nos referimos al sistema de posicionamiento global. Este nos permitió avanzar en el ámbito de la localización de una gran manera, ya que en la actualidad, es una de las herramientas más indispensables por los usuarios al momento de guiarse por algún lugar, lo que se debe a la facilidad de su uso, y al poder usarse en cualquier dispositivo electrónico moderno.

Si bien el excelente trabajo del GPS en el posicionamiento en exteriores es indiscutible, éste posee un gran defecto en interiores debido a que los satélites con los que funciona no son capaces de obtener buenas mediciones dentro de un espacio cerrado. Es en este punto en donde entran en juego nuevas tecnologías, capaces de solucionar los problemas presentes en el ámbito de la detección.

En la actualidad hay varias formas de poder obtener la posición de un usuario dentro de un espacio cerrado, pero hace algunos años apareció una nueva tecnología con la cual se espera una

localización del usuario mucho mas precisa y de fácil uso. Esta nueva tecnología corresponde al protocolo 802.11mc, el cual nos permitirá obtener una estimación de la distancia del usuario a través de un intercambio de mensajes. Es a través de este nuevo protocolo que se da inicio a este proyecto, en el cual se brindará el conocimiento necesario para poder implementarlo en un sistema de posicionamiento.

## 1.1. Definición del Problema

El problema que se abordará dentro de esta memoria de titulo está enfocado en la implementación de un método que permita mejorar la técnica de posicionamiento de objetos en interiores denominada localización indoor. En la actualidad hay diferentes formas y técnicas que permiten la detección en interiores de manera eficiente, pero aun así, éstas no están exentas de problemas al momento de la estimación de una correspondiente posición, esto se debe a diferentes factores al momento de implementar, como por ejemplo: Altos costos de implementación, problemas de detección por intensidad, dependencia de aplicaciones externas, entre otras. Es debido a esto que la finalidad de esta memoria es implementar una nueva tecnología de detección para ver si esta es capaz de mejorar los resultados obtenidos por las tecnologías ya existentes de manera mucho más precisa.

## 1.2. Estado del Arte

En la presente sección se dispondrá al lector de la información recopilada para abarcar de una mejor manera todo lo relacionado con el tema de localización indoor mediante FTM, en el cual, se presentara información relevante con respecto a diferentes temas que permitirán obtener un conocimiento general del tema que se quiere abarcar.

### 1.2.1. Trabajos previos

#### Protocolo 802.11mc

- Betzi Mabel Paccha Colala, "Caracterización de la métrica RTT para la implementación de algoritmos de localización de smartphones en entornos indoor", trabajo de fin de grado, Ingeniera en Telemática, tecnologías de la Información y las Comunicaciones, Escuela Técnica Superior de Ingeniería de Telecomunicación[1].

El presente informe se pretende implementar un sistema de localización de dispositivos IoT en interiores empleando como tecnología para el posicionamiento redes Wi-Fi. El sistema implementado emplea la combinación de dos técnicas de localización. Una de estas técnicas se basa en el estándar IEEE802.11mc, donde emplea la métrica RTT como estimador de la distancia a la que se encuentra un dispositivo móvil con respecto a un AP compatible con dicha tecnología. Y como segunda técnica se emplea el sistema monopulso, con el que se pretende estimar la dirección de la llegada de la señal que emita el dispositivo IoT.

- Leor Banin, Uri Schatzberg, Yuval Amizur – Intel Corp, Next Generation Indoor Positioning System Based on WiFi Time of Flight, septiembre 2013 [5].

En este artículo se presenta el protocolo WiFi Time of Flight (ToF), un nuevo protocolo de medición de rango basado en el tiempo de alta precisión que proporciona información de posicionamiento de alta precisión. El concepto básico de WiFi ToF es determinar distancias midiendo los tiempos de viaje, de manera similar al GPS. Sin embargo, WiFi ToF se enfrenta a algunos de los principales desafíos que plantea WiFi a diferencia del GPS: la falta de una señal de hora sincronizada y la propagación por trayectos múltiples.

#### Técnicas para la obtención de medidas

- Berthold K. P. Horn, "Observation Model for Indoor Positioning", Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA, 20 de julio 2020[6].

En el presente informe se presentan algunas técnicas que pueden interactuar con el protocolo 802.11mc, para poder determinar de mejor medida la posición de un usuario utilizando posiciones conocidas de los puntos de acceso. El problema de la detección se vuelve mas

complejo en un espacio de interior donde las señales atraviesan paredes, muebles, suelos, entre otros obstáculos produciendo que se complique la detección de un individuo. El método de Bayesian Grid Update se ha mostrado particularmente prometedor para mejorar el proceso de detección, este método requiere un "modelo de observación" que da la probabilidad condicional de observar una distancia informada dada una distancia real conocida.

- Berthold K. P. Horn, "Doubling the Accuracy of Indoor Positioning: Frequency Diversity", Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA, 9 de marzo 2020 [7].

En este trabajo se abordaran temas correspondientes a la determinación de la posición en interior basada en la medición de tiempo fino (FTM) del tiempo de ida y vuelta (RTT) de una señal entre un iniciador y un respondedor. Sin embargo, se debe de considerar que la precisión que se alcanza actualmente con FTM limita el rango de posibles aplicaciones, debido a que posee una desviación estándar de 1 a 2 m (en condiciones óptimas). Es por esto, que este artículo presenta diferentes métodos para la obtención de medidas de una forma más eficiente.

- Markus Bullmann (1), Toni Fetzner (1), Frank Ebner (1), Markus Ebner (1), Frank Deinzer (1) and Marcin Grzegorzek (2), "Comparison of 2.4 GHz WiFi FTM- and RSSI-Based Indoor Positioning Methods in Realistic Scenarios", (1) Faculty of Computer Science and Business Information Systems, University of Applied Sciences Würzburg-Schweinfurt, 97070 Würzburg, Germany; (2) Institute of Medical Informatics, University of Lübeck, 23547 Lübeck, Germany, 12 de agosto del 2020 [8].

En el presente trabajo, se evaluó FTM en el Banda de 2,4 GHz con un ancho de banda de 20 MHz para el concepto de posicionamiento en interiores, pero de forma realistas. Para este propósito, se implemento un método de estimación de mínimos cuadrados, un enfoque de posicionamiento probabilístico y una implementación de filtro de partículas simplista. Cada método se evalúa utilizando FTM y RSSI por separado para mostrar la diferencia de las técnicas.

## Competencia

- Daniel Barba , Ana Moretón-Fernández , Héctor del Campo-Pando ,Sergio García-Villanueva Juan March y Diego R. Llanos,“ Uso de Bluetooth de Baja Energía en Aplicaciones de Localización Indoor" [2].

En el presente documento se mencionan diferentes formas que se ocupan en la actualidad para el proceso de detección de interiores. Estas técnicas tratan de resolver problemas en diferentes campos y actividades, como la sanidad, prevención de riesgos o diferentes actividades comerciales. En estos lugares, donde la localización mediante GPS no está disponible, es necesario ofrecer una nueva solución que permita resolver el problema ofreciendo una precisión adecuada y manteniendo la viabilidad económica del proyecto. Algunas de estas técnicas son, por ejemplo: Beacons, RFID, Ultrasonido, entre otras.

## Implementación

- Guia de programacion para ESP32-S2, espressif, Espressif Systems (Shanghai) Co., Ltd, 2016 - 2021 [9].

En esta guia se entregan las herramientas necesarias para la instalación de los ESP32-S2, los cuales se utilizaran para ejecutar el correspondiente ejemplo de prueba de FTM.

## 1.3. Discusión

De la información obtenida de los artículos antes mencionados podemos considerar que si bien ya existen diferentes alternativas con las cuales poder detectar la localización de interiores de manera precisa, uno de sus principales problemas al momento de realizar el proceso de detección es que éstas se basan en mediciones de intensidad para obtener resultados, dicha forma de medición no es la mas óptima de utilizar, debido a que es muy susceptible a errores. Es en este punto en donde el protocolo FTM sobresale sobre los demás, ya que permitirá obtener sus datos a través del tiempo de vuelo, lo que facilitaría y entregaría mejores resultados en el ámbito de la detección.

## 1.4. Hipótesis de Trabajo

La implementación del código en función al protocolo FTM permite una mejora importante en el ámbito de la localización indoor en comparación a las ya existentes técnicas de detección. Esta hipótesis se basa principalmente en que la implementación del sistema es relativamente sencilla y de bajo costo, por lo que sería posible de implementar en cualquier lugar entregando resultados óptimos sin grandes esfuerzos.

## 1.5. Objetivos

### 1.5.1. Objetivo General

El objetivo general de esta memoria es la implementación de un código en función del protocolo FTM para así obtener un sistema de localización indoor de bajo costo, ya que se emplearán dispositivos de bajo consumo como sistema de detección, y de gran eficiencia, debido a que empleará un sistema de obtención de datos que no se ve tan afectado a interferencias y obstrucciones, como los sistemas de detección actuales basados en intensidad.

### 1.5.2. Objetivos Específicos

- Objetivo 1

Investigar las herramientas con las que se llevará a cabo el trabajo.

- Objetivo 2

Determinar la técnica de detección que mejor se implemente en un espacio cerrado.

- Objetivo 3

Desarrollar un software capaz de detectar la distancia entre un STA y los diferentes AP que se encontrarán en un espacio cerrado.

## 1.6. Alcances y Limitaciones

Los alcances para este proyecto son:

- Se trabajará con un sistema de detección basado en 3 ESP y 3 antenas para la detección de la distancia.
- Se empleará un código capaz de entregar la posición de un STA en un espacio 2D.

Las limitaciones para este proyecto son:

- La implementación del proyecto se llevará a cabo en una casa, para ser más exactos sólo en el primer piso de ésta, esto se debe a que por temas de la contingencia el proceso de medición de datos, el cual se implementaría en un laboratorio, no se pudo realizar.
- El lugar de medición contara con una superficie de 4x6 mts.
- Por motivos de pandemia, no se pudieron obtener los 4 dispositivos ESP para la realización del proyecto, por lo que se adaptó la obtención de medidas.
- El código viene implementado para realizar comunicación con sólo un AP y no con los tres al mismo tiempo, por lo que si bien se puede optimizar para que el STA reconozca los diferentes AP, el proceso de medición se debe de realizar de manera offline en la cual el usuario deberá de conectarse a cada AP de manera manual.
- Por diferentes motivos los ESP no detectaban de buena manera las diferentes mediciones realizadas, por lo que se debieron implementar antenas externas para un mejor funcionamiento.



## 1.7. Metodología

La metodología que se empleará en este proyecto comenzará con la obtención de los conocimientos necesarios para el entendimiento del protocolo y de la trilateración que posteriormente se empleará en un programa. Luego de esto, se llevará a cabo la implementación del ejemplo de FTM para así ir viendo las capacidades y falencias que tiene al principio el sistema. Posteriormente, se desarrollará un sistema capaz de detectar de manera clara el movimiento de un individuo dentro de un espacio cerrado. Cabe destacar que para la realización de este procedimiento se deberán de considerar diferentes parámetros además de considerar el uso de más equipos ESP32-S2 para lograr mejores resultados.

# Capítulo 2

## Fundamentos teóricos

### 2.1. Introducción

En el presente tema, se ve todo lo relacionado con la teoría que abarcara el proyecto, esto quiere decir que se expondrá la teoría vista para el entendimiento del protocolo, algunas de las diferentes formas de detección tanto en interiores como en exteriores que existen y el concepto de trilateración que se implementará para la obtención de un mejor resultado en nuestro sistema.

### 2.2. Marco Teórico

#### 2.2.1. Localización en exteriores

Antes de profundizar en el tema principal de la memoria, es importante conocer el concepto de la localización en exteriores y las tecnologías que se implementan en ésta, principalmente para poder entender el concepto general del sistema de localización y como este posee similitudes con el sistema de localización en interiores[11].

La localización en exterior es posible de implementar gracias al GPS, el cual tiene como objetivo el determinar la posición de un punto cualquiera dentro de un espacio de coordenadas. Dicha posición; es posible de determinar gracias al cálculo de las distancias de por lo menos tres satélites, de los cuales; se posee la ubicación exacta de éstos. La forma en la que se obtiene la

distancia entre el punto de estudio y el satélite, proviene de la multiplicación entre el tiempo de vuelo de la señal emitida y la velocidad de propagación de ésta. Si bien, esta forma de localización es indispensable en la actualidad gracias a sus versatilidad de funciones que esta posee, también tiene defectos que poco a poco se deben ir mejorando. Uno de sus principales problemas es el concepto de localización en espacios cerrados, por lo que se volvió indispensable el poder implementar dicho sistema de localización en interiores, en la cual la premisa de su funcionamiento es la misma: un sistema que cuente con varios equipos de detección en el interior de un espacio (que cumplan la misma función que los satélites en el exterior) y posteriormente aplicar el concepto de tiempo de vuelo para así determinar la posición del usuario.

### 2.2.2. Detección de interiores

En este punto se hablará de otras tecnologías que actualmente se usan para la obtención de la posición de las personas [2] [3].

- Beacons o BLE (Bluetooth Low Energy)

Como su nombre lo indica los Beacons son dispositivos de baja energía que generan una señal de bluetooth de baja energía, con la cual son capaces de conectarse con otros dispositivos móviles que se encuentren dentro de su rango de funcionamiento. La forma mas común de implementación de éstos es colocar una cierta cantidad en diferentes lugares dentro del espacio a trabajar, posteriormente, los Beacons emiten cada cierto tiempo un pequeño paquete de datos los cuales son recibidos por el receptor el cual se encarga de, en función de la intensidad de señal recibida, calcular la distancia que hay entre estos para así obtener su posicionamiento.

Ventajas

- Los dispositivos receptores pueden tener cualquier sistema operativo para su funcionamiento.
- Las necesidades de cómputo pueden trasladarse a los dispositivos receptores, quienes únicamente tienen que preocuparse de su ubicación según los paquetes que reciben.

### Desventajas

- Se necesita de cierta aplicación para que exista un funcionamiento pleno del servicio.
- El usuario al no estar relativamente cerca del Beacons no se puede saber con claridad su ubicación, ya que al funcionar con la intensidad de la señal, la estimación no sería muy certera.

#### ■ RFID (Radio Frequency Identification)

Este sistema se basa en un conjunto de tecnologías capaces de leer etiquetas o tags a distancia a través de un lector, no necesariamente tienen que tener una línea de vista directa con el lector, permitiendo así el traspaso de información entre estos.

Dentro de los RFID podemos encontrar 2 tipos: Los del tipo pasivo, que son capaces de entregar la información gracias a la energía que entrega el lector, por lo cual las hace más económicas al no poseer una fuente de energía propia, pero a su vez la vuelve más ineficientes ya que su rango de detección es menor. Por otro lado se encuentran las del tipo activo, que a diferencia del anterior tipo, poseen una fuente de alimentación propia y su rango de detección es mucho mayor, pero como desventaja su tiempo de vida es más limitado, además que de su costo es más elevado.

#### ■ Ultrasonido.

Existen varias formas de determinar la localización de una persona a través de ultrasonidos. Por ejemplo el sistema BAT, este sistema se basa en que el usuario u objeto a determinar su posición posea un dispositivo capaz de emitir un ultrasonido al recibir una señal, la cual será recepcionada por una red de receptores, ubicados en el lugar de estudio, que determinarán la ubicación del punto de emisión.

### Ventajas

- Su mayor ventaja radica en el bajo costo que tiene para implementarse.

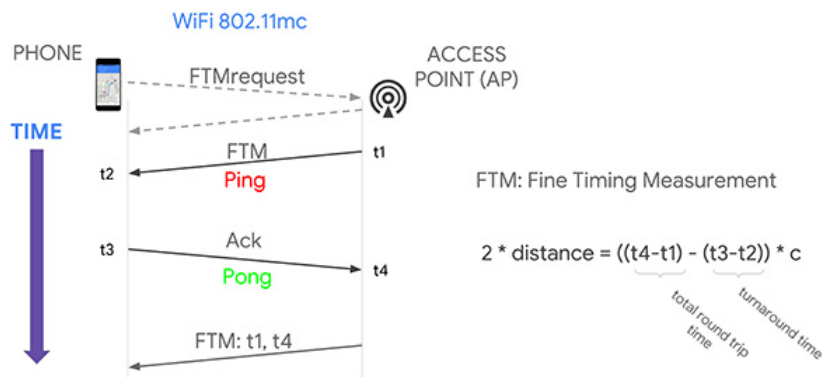
Desventajas

- La señal solo funciona dentro de un espacio específico.
- La refracción afecta severamente su desempeño.

### 2.2.3. Protocolo 802.11 mc

Para que este proyecto empiece a tener forma, debemos de hablar primero de la herramienta que permitirá poner en marcha todo el proceso de localización, esta herramienta no es más que el protocolo 802.11mc o también denominada protocolo WiFi Fine Time Measurement (FTM). Este protocolo fue publicado en el año 2016 y permite que un dispositivo WLAN mida con precisión el tiempo de viaje de ida y de vuelta a otro dispositivo (RTT) [1][4][5].

La forma con que se obtiene el RTT se puede ver en la figura 2.1. En esta se nos muestra que para obtener el RTT se debe considerar un initiator y un responder. Según el estándar el initiator es una estación STA que inicia el proceso en el cual se envía una solicitud FTM a un AP correspondiente. Posteriormente, el AP producirá un protocolo ping-pong en donde el ping enviado al STA corresponderá al FTM y el pong que es enviado devuelta corresponderá a un reconocimiento de este paquete.



**Fig. 2.1:** Concepto básico del WiFi RTT.

En la figura 2.1 podemos observar que al momento de generarse el procesos, aparecen en el sistema 4 nuevos elementos (T1, T2, T3 y T4) los cuales corresponderán a los diferentes tiempos

para que se lleve a cabo el proceso. T1 corresponderá a la hora de salida de mensaje por parte de la AP, T2 corresponderá a el tiempo de llegada del mensaje proveniente del AP, T3 corresponde a la hora de salida del mensaje recibido por parte del STA y finalmente el T4 corresponderá a la hora de llegada del mensaje enviado del STA al AP. Una vez que se determinan los diferentes tiempos antes mencionados, el sistema puede utilizar las formulas necesarias para obtener el RTT, TOF y la distancia, las cuales estarán dadas por:

$$RTT = (t_4 - t_1) - (t_3 - t_2) \quad (2.1)$$

$$ToF = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (2.2)$$

$$distancia = \left( \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \right) \cdot c \quad (2.3)$$

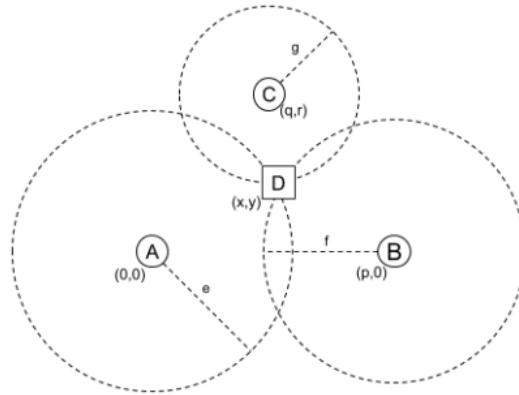
Si bien en la teoría este proceso es correcto, no basta con que se realice una vez para obtener los resultados óptimos, debido a esto se recomienda repetir el proceso varias veces, para que así se pueda obtener una obtención de datos mucho mas precisa al respecto. En la actualidad, el protocolo FTM se encuentra presente en Android 9 (API nivel 28) y Android 10 (API nivel 29), permitiendo que su uso sea aún más fácil para los usuarios de móviles.

#### 2.2.4. Técnica de medición

- Trilateración

De forma resumida, la trilateración consiste en la medición de una determinada posición en función de posiciones ya establecidas. Para nuestro caso, las posiciones establecidas corresponderían a los AP, para los cuales se asume un rango de detección en forma circular, en el cual se puede estimar la ubicación del STA al momento que este se logre conectar al dispositivo. Cabe recalcar, que al momento de la detección se deben de considerar posibles error de estimación, ya que, cabe la posibilidad que el STA se encuentre en una posición en donde los rangos de los AP se intercepten produciendo inconsistencias en las medidas, dicho problema se puede resolver a través del método de mínimos cuadrados, el cual se encarga de minimizar el error cuadrático entre la distancia medida y la distancia real en un punto dado.

Como se puede observar en la figura 2.2, para lograr el proceso de trilateración es necesario realizar una serie de estimaciones, entre las cuales se encuentra la ubicación de los tres AP (para este caso se consideraran como puntos A, B y C), además se debe de considerar la distancia entre el STA (considerado como D) y los AP (dicha distancia entre estos se consideraran como e, f y g) [10].



**Fig. 2.2:** Esquema de trilateración.

Para poder realizar de forma sencilla este proceso, se considerará que uno de los AP estará ubicado en el origen (0,0) y los otros dos AP estarán ubicados en relación a este. Para este caso se consideraran como puntos de referencias el AP B (p,0) y el AP C (q,r). Considerando que ya poseemos la ubicación de los AP, podemos considerar la ecuación de un círculo para así poder determinar la ubicación del STA.

$$e^2 = x^2 + y^2 \quad (2.4)$$

$$f^2 = (x - p)^2 + y^2 \quad (2.5)$$

$$g^2 = (x - q)^2 + (y - r)^2 \quad (2.6)$$

Finalmente, estas ecuaciones permitirán la obtención de la posición del STA la cual corresponderá a (x,y), para lograr dicha posición bastan con reemplazar las ecuaciones entre

ellas para posteriormente despejar las incógnitas. De este modo, se tiene que las incógnitas  $x$  e  $y$  están dadas por las siguientes ecuaciones.

$$x = \frac{e^2 - f^2 + p^2}{2p} \quad (2.7)$$

$$y = \frac{e^2 - g^2 + q^2 + r^2}{2r} - \frac{q}{r}x \quad (2.8)$$



# Capítulo 3

## Desarrollo

### 3.1. Introducción

En este capítulo se presentará todo el desarrollo previo a la obtención de los resultados, en el que se verán diferentes temas como son el equipamiento con el cual se trabajo, el código de ejemplo que se utilizó como base para el proyecto, el funcionamiento del código y finalmente, la mejora del código para su posterior utilización.

#### 3.1.1. Equipamiento de trabajo

Antes de empezar con la realización del sistema, se indicara los elementos que estarán presentes para la realización de este:

- ESP32-S2

En el apartado físico del proyecto se priorizo un equipamiento simple para cumplir con la función de ser tanto un STA como un AP. Además, este debía ser de bajo costo y alimentarse de manera simple para su utilización en cualquier espacio cerrado. Dicho dispositivo capaz de cumplir con las características mencionadas corresponde al chip ESP32-S2, este dispositivo es un SoC de un solo núcleo, de bajo consumo, de reducido tamaño, posee WI-FI y permite trabajar con FTM, lo cual lo vuelve en un equipo indispensable para éste proyecto.

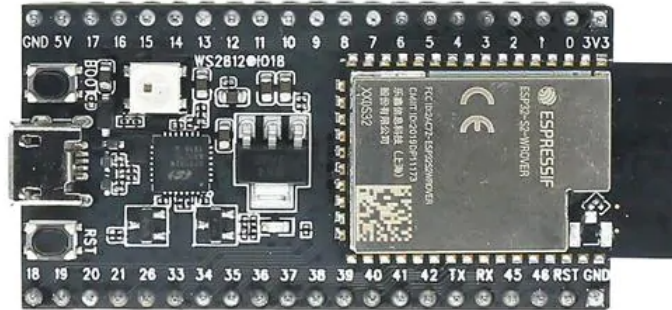


Fig. 3.1: ESPs2-32.

- Antena externa

Si bien el ESP funciona de gran manera para la obtención de distancia de objetos en interiores, se priorizó el uso de este en conjunto con una antena externa para la obtención de valores. Dicha antena se conectaba de manera directa al ESP.



Fig. 3.2: Antena externa.

## 3.2. Código de ejemplo

Una vez comentado lo correspondiente al protocolo FTM, se debe de analizar un código de ejemplo correspondiente a este. Dicho código permitirá realizar el proceso de obtención de los

diferentes tiempos (T1, T2, T3 y T4) además de indicar la distancia aproximada en la cual se encuentra el STA. Dicho ejemplo permitirá entender la implementación del protocolo FTM en el ESP de manera simple, para posteriormente implementar el sistema completo de detección de interiores sin ningún contratiempo. Se debe de considerar que para que este código funcione seguiremos los pasos mencionados en la página de Espressif, empresa encargada en la creación de los ESP [9].

### 3.2.1. Funcionamiento del código

Posterior a la instalación de las herramientas básicas para el reconocimiento del ejemplo por parte de los ESP32-S2 e implementación de diferentes comandos para la calibración de los ESP, se nos presentará la siguiente ventana.

```
=====
                          Steps to test FTM
=====
| 1. Use 'help' to gain overview of commands
| 2. Use 'scan' command to search for external AP's
|                               OR
| 2. Start SoftAP on another device using 'ap' command
| 3. Start FTM with command 'ftm -I -s <SSID>'
=====
```

Fig. 3.3: Pantalla inicial del ejemplo de FTM.

Ahora que ya estamos en la ventana del ejemplo, debemos de configurar los 2 ESP para que se produzca la comunicación entre ellos, considerando el primer ESP como un STA y el segundo ESP como un AP, se deberán de implementar los siguientes comandos para la obtención de los resultados.

- `help`: Muestra la implementación de los comandos disponibles.
- `ap < ssid >< password >`: Permite la utilización de un ESP32-S2 como un Ap asignándole un correspondiente ssid y contraseña
- `scan < ssid >`: Busca APs con el ssid correspondiente e imprime si los encuentra.
- `ftm -I -s < ssid >`: Comienza un procedimiento FTM con los detalles especificados y con el AP del ssid correspondiente.

Una vez implementado el código, se nos muestra la siguiente imagen en consola.

```
I (2647185) ftm_station: sta start to scan
I (2649285) ftm_station: [FTM][rssi=-85][FTM Responder]
I (2649285) ftm_station: [Carlos][rssi=-95]
I (2649285) ftm_station: sta scan done
```

Fig. 3.4: Parte 1 del Ejemplo en funcionamiento.

Aquí podemos observar que nuestro primer ESP, correspondiente a STA, es capaz de reconocer a nuestro Ap asignado (FTM), correspondiente a nuestro segundo ESP. Una vez reconocidos podemos inicializar el código.

```
I (2664025) ftm_station: Requesting FTM session with Frm Count - 32, Burst Period - 200mSec (0: No Preference)
I (2664025) ftm_station: FTM Report:
W (2664025) wifi:Starting FTM session in 0.200 Sec
I (2664035) ftm_station:
I (2664055) ftm_station:
I (2664065) ftm_station:
I (2664075) ftm_station:
I (2664085) ftm_station:
I (2664095) ftm_station:
I (2664115) ftm_station:
I (2664125) ftm_station:
I (2664135) ftm_station:
I (2664145) ftm_station:
I (2664155) ftm_station:
I (2664165) ftm_station:
I (2664185) ftm_station:
I (2664185) ftm_station:
I (2664195) ftm_station:
I (2664205) ftm_station:
I (2664215) ftm_station:
I (2664225) ftm_station:
I (2664245) ftm_station:
I (2664255) ftm_station:
I (2664265) ftm_station:
I (2664275) ftm_station:
I (2664285) ftm_station:
I (2664305) ftm_station:
I (2664315) ftm_station:
I (2664325) ftm_station:
I (2664335) ftm_station:
I (2664345) ftm_station:
I (2664365) ftm_station: Estimated RTT - 3 nSec, Estimated Distance - 0.52 meters
ftm> W (2665645) wifi:FTM session ends with 27 valid readings out of 31, Avg raw RTT: 10.532 nSec, Avg RSSI: -85
```

Diag	RTT	T1	T2	T3	T4	RSSI
5	23438	1737349564062	883045631250	883150031250	1737453987500	-86
6	25000	1769563564062	915259642187	915364031250	1769667978125	-87
7	23437	1801776564062	947472676562	947577031250	1801880942187	-88
8	0	1833980564062	979676679687	979781031250	1834084906250	-86
9	10937	1937511564062	1083207767187	1083312031250	1937615839062	-85
10	10937	1969716576562	1115412795312	1115517031250	1969820823437	-87
11	15625	2002019564062	1147715812500	1147820031250	2002123798437	-86
12	29688	2034169564062	1179865846875	1179970031250	2034273778125	-87
13	7813	2137826564062	1283522915625	1283627031250	2137930687500	-87
14	20312	2170049564062	1315745939062	1315850031250	2170153676562	-87
15	6250	2202271564062	1347967964062	1348072031250	2202375637500	-86
16	10937	2234466564062	1380162985937	1380267031250	2234570620312	-86
17	20312	2338015564062	1483712067187	1483816031250	2338119548437	-87
18	10937	2370247564062	1515944082812	1516048031250	2370351523437	-87
19	20313	2402451576562	1548148128125	1548252031250	2402555500000	-87
20	21875	2434673564062	1580370129687	1580474031250	243477487500	-87
21	21875	2538241576562	1683938235937	1684042031250	2538345393750	-87
22	25000	2570445564062	1716142235937	1716246031250	2570549384375	-87
23	15625	2602713564062	1748410256250	1748514031250	2602817354687	-87
24	12500	2634934564062	1780631282812	1780735031250	2635038325000	-87
25	14062	2738510576562	1884207373437	1884311031250	2738614248437	-86
26	0	2770686576562	1916383389062	1916487031250	2770790212500	-86
27	10938	2802998564062	1948695400000	1948799031250	2803102206250	-87
28	23437	2835156564062	1980853435937	1980957031250	2835260182812	-87
29	0	2938704564062	2084401498437	2084505031250	2938808093750	-87
30	1563	2970871576562	2116568534375	2116672031250	2970975075000	-87
31	17188	3003174564062	2148871550000	2148976031250	3003279062500	-86

Fig. 3.5: Parte 2 del Ejemplo en funcionamiento.

De este podemos observar que en ventana de comando se registraron los diferentes valores para nuestros diferentes T, además de el tiempo estimado del RTT y la distancia estimada en la cual se encuentra el STA.

Con la implementación correcta entre un STA y un AP, se procederá a realizar un sistema capaz de detectar la ubicación de un objetivo dentro de un espacio cerrado. Considerando que este código ejemplo esta en función de 2 equipos, se debe obtener una forma que permita la

obtención de datos para 4 equipos, en el cual uno corresponde al equipo STA y los otros tres corresponderán a los equipos AP. Para este sistema se considerará que el equipo que estará conectado directo al computador será el STA y los equipos que necesiten de una alimentación externa para su funcionamiento corresponderán a los AP. Debido a esto es necesario realizar modificaciones en el código para que los AP sean capaces de realizar la comunicación FTM sin la necesidad de estar conectados directamente a un computador.

### 3.3. Automatización del código

Ahora se debe de solucionar una de las limitaciones principales que posee el código de ejemplo en el cual se basa dicho trabajo y es que los dispositivos que se emplean como AP deben de estar conectados directamente a un computador para su funcionamiento como AP capaces de realizar la comunicación FTM. Para solucionar dicho inconveniente, se realizó una optimización en el código permitiendo que los dispositivos AP al conectarse directamente a corriente, a través de un cargador, puedan realizar la comunicación FTM. Como se observa en la figura 3.6 el sistema estará compuesto por el ESP que actúa como AP, la antena y la fuente de alimentación simple.



**Fig. 3.6:** Partes del sistema AP.

# Capítulo 4

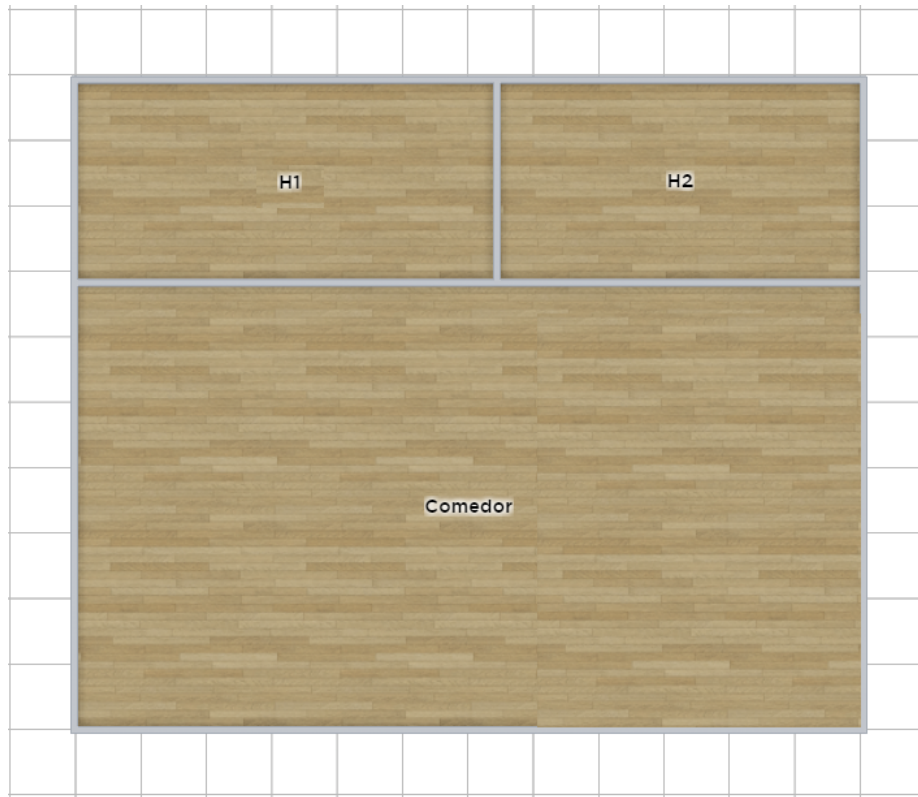
## Resultados

### 4.1. Introducción

Dentro del siguiente capítulo se expondrán los diferentes resultados obtenidos posterior a la ejecución del código ejemplo mostrado anteriormente, hablando en detalle de diferentes puntos que permitirán la obtención de las mediciones realizadas entre el STA y los AP. Posterior a esto, se analizarán los resultados obtenidos para así poder determinar los puntos fuertes y débiles que tiene el sistema y ver cuales serán las posibles soluciones que se deberían poder implantar a futuro para obtener mejores resultados

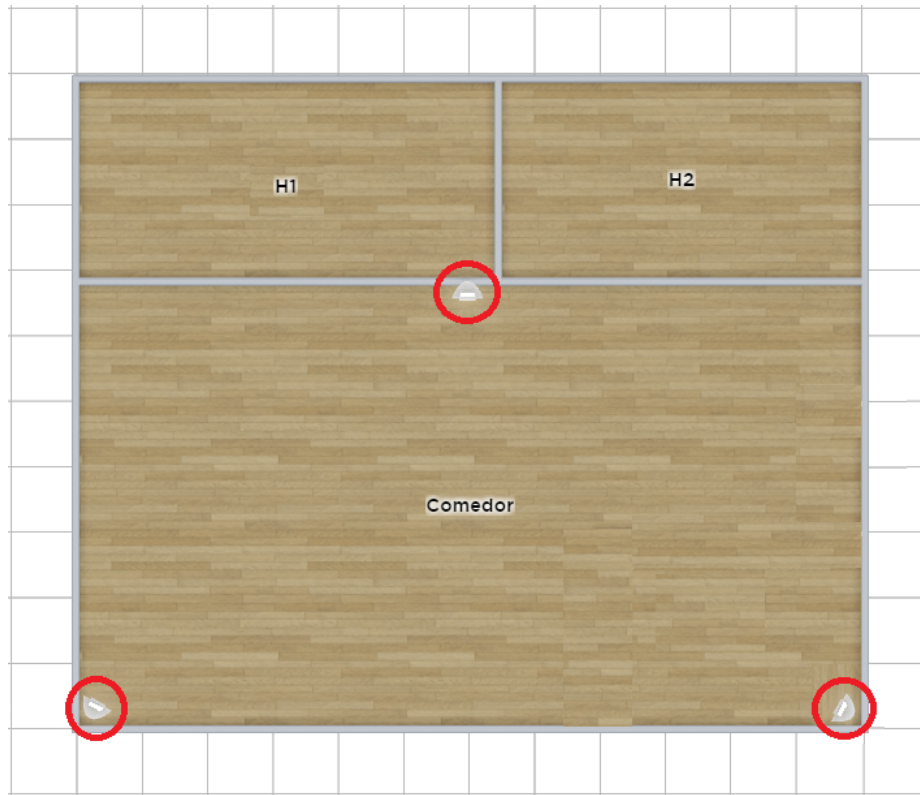
### 4.2. Lugar de medición

Podemos observar en la figura [4.1](#) una representación minimalista del entorno de trabajo, en la que se observa la distribución de las salas que esta tiene al momento de realizar la correspondiente medición. Dicho lugar contará con tres espacios claramente definidos, un comedor y dos habitaciones, además cabe destacar que ésta no dispondrá de un ambiente ideal para el momento de la medición, ya que al igual que en cualquier lugar, este espacio contará con diferentes objetos que darán un ambiente realista al momento de realizar la medición.



**Fig. 4.1:** Imagen referencial del lugar de medición.

Una vez visto el lugar en el cual se trabajará, se procede a posicionar los ESP que actuarán como AP en el lugar de medición.



**Fig. 4.2:** Imagen referencial del lugar de medición.

Como se puede observar en la figura 4.2, la ubicación de los AP se dispuso de tal manera que al momento de determinar la ubicación del STA dentro del espacio del comedor esta sea lo más cercana posible a la realidad. Además, se consideró dicha ubicación debido a que el espacio del comedor era el que posee mayor superficie en comparación a las otras habitaciones. Si bien es verdad que la mayoría de las mediciones se realizarán en el comedor, tanto la habitación 1 como la habitación 2 también se emplearán como espacios de medición.



### 4.3. Ejecución del código

Una vez optimizado el código he implementado los AP remotamente, se realizó el proceso de medición de las distancia entre el STA y los diferentes AP. Cabe destacar que una información relevante con la implementación del código, como se menciono con anterioridad, en el sistema en general se implementaron antenas capaces de mejorar la obtención de datos por lo que al momento de implementar el código y ejecutar el comando SCAN se observó un incremento considerable respecto a la cantidad de AP que fue capaz de reconocer el STA, observando la figura 4.3 se observa que las primeras AP detectadas corresponderán a los AP capaces de trabajar con el protocolo FTM, posteriormente se observa diferentes AP de los alrededores, los cuales, son incapaces de realizar la comunicación FTM por lo que se omitirán en el proceso de medición.

```
I (3785) ftm_station: sta start to scan
I (5895) ftm_station: [AP1][rssi=-44][FTM Responder]
I (5895) ftm_station: [AP2][rssi=-48][FTM Responder]
I (5895) ftm_station: [Carlos][rssi=-53]
I (5905) ftm_station: [Aly][rssi=-73]
I (5905) ftm_station: [Romi][rssi=-73]
I (5915) ftm_station: [POLETT][rssi=-82]
I (5915) ftm_station: [Home][rssi=-82]
I (5915) ftm_station: [La vida es bella][rssi=-83]
I (5925) ftm_station: [Eugenia][rssi=-83]
I (5925) ftm_station: [Andrea][rssi=-86]
I (5935) ftm_station: [JOTAESE][rssi=-87]
I (5935) ftm_station: sta scan done
```

Fig. 4.3: Detección de los AP.

Como se mencionó en el tema de las limitaciones, no fue posible contar con cuatro ESP para realizar el proceso de medición. Esto se puede observar en la figura 4.3 donde la consola solo muestra dos AP y no tres como debería ser para poder implementar correctamente el proceso de trilateración, por lo que una vez realizadas las mediciones para el AP1 y AP2, se cambio la posición del AP2 a la posición del AP3. Lo antes mencionado solo pudo llevarse a cabo debido a que el código utilizado solo permitía la comunicación de forma offline, por lo que se eligieron con antelación los diferentes puntos en los cuales se realizarían las mediciones entre el STA y los AP.

Una vez listo el sistema tanto en equipamiento como en código, se realizó las diferentes mediciones con respecto a todos los AP, en total se consideraron quince diferentes posiciones

entregando los siguientes resultados.

Posición	AP 1	AP 2	AP 3
1	0.5	3.7	5.7
2	2.02	3.85	5.01
3	2.9	3.9	3.7
4	1.9	1.65	6.6
5	1.7	2.8	5.6
6	5.28	4.68	2.4
7	1.17	3.90	5.7
8	4.6	4.55	7.14
9	2.8	1.17	4.54
10	4.01	1.82	5.45
11	4.70	2.73	4.9
12	2.60	3.25	4.3
13	2.86	3.77	3.9
14	2.21	2.99	5.9
15	2.76	1.6	1.2

**Tabla 4.1:** Distancia entre el STA con cada AP.

Una vez ya tabulado los diferentes valores obtenidos entre el STA y los diferentes AP se procede a obtener la posición exacta en la que se encuentra nuestro STA en función a los valores correspondientes de la tabla 4.1 implementando el proceso de trilateración. Como se observó en el capítulo 2: avance del tema, se conoce que las ecuaciones 2.7 y 2.8 permitirán la obtención del posicionamiento del STA en un plano 2D, por lo que ahora se deben de asignar diferentes valores a las incógnitas presentes en las ecuaciones.

Como se indicó con anterioridad, los valores para e, f y g corresponderán a las distancia para cada AP con respecto al STA o de forma más simple, corresponderán a los valores expresados en la tabla 4.1. Para el caso de los valores de p, q y r, éstos tendrán relación con la posición que

se indique para los AP, por lo que para simplificar un poco las mediciones, se considerara que sus valores serán  $p = 4$ ,  $q = 2$  y  $r = 6$ . Considerando que el primer AP este ubicado en el origen del sistema  $AP1 = (0,0)$ , los otros dos AP estarán en la posición  $AP1 = (4,0)$  y  $AP1 = (2,6)$ .

Reemplazando los diferentes valores antes mencionados en las ecuaciones 2.7 y 2.8, se tiene la siguiente tabla.

Posición	STA
1	(0.339988,0.539171)
2	(0.65525,1.385417)
3	(1.12563,2.57171)
4	(0.7832,0.821067)
5	(1.38125,0.51375)
6	(2.70245,4.213317)
7	(2.1,3.22)
8	(2.05719,0.1693)
9	(2.80889,1.325)
10	(3.514,0.9132)
11	(3.84,1.413)
12	(1.525,1.848)
13	(1.4584,3.63219)
14	(1.493,0.3184)
15	(2.6422,4.526)

**Tabla 4.2:** Posición del STA a través de la técnica de trilateración.

Para la ejecución de la técnica de trilateración, se implementó un pequeño código el cual de manera simple nos indica la posición del STA.

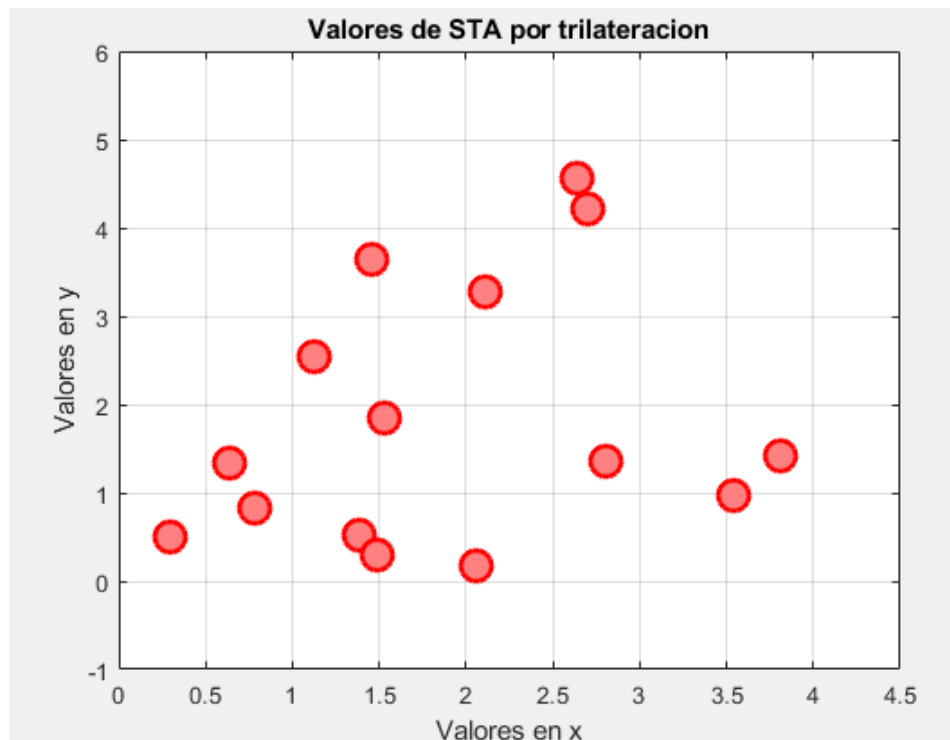
```

La posición entre el AP1 y STA es:2.6
La posición entre el AP2 y STA es:3.25
La posición entre el AP3 y STA es:4.3
La posición final del STA es:(1.52469,1.8476)

```

**Fig. 4.4:** Imagen ejemplo del código de trilateración.

Dicha posición final que se obtiene del código de trilateración se implementó en un gráfico para así observar con mayor claridad los diferentes valores obtenidos.



**Fig. 4.5:** Gráfica de las posiciones del STA utilizando el código.

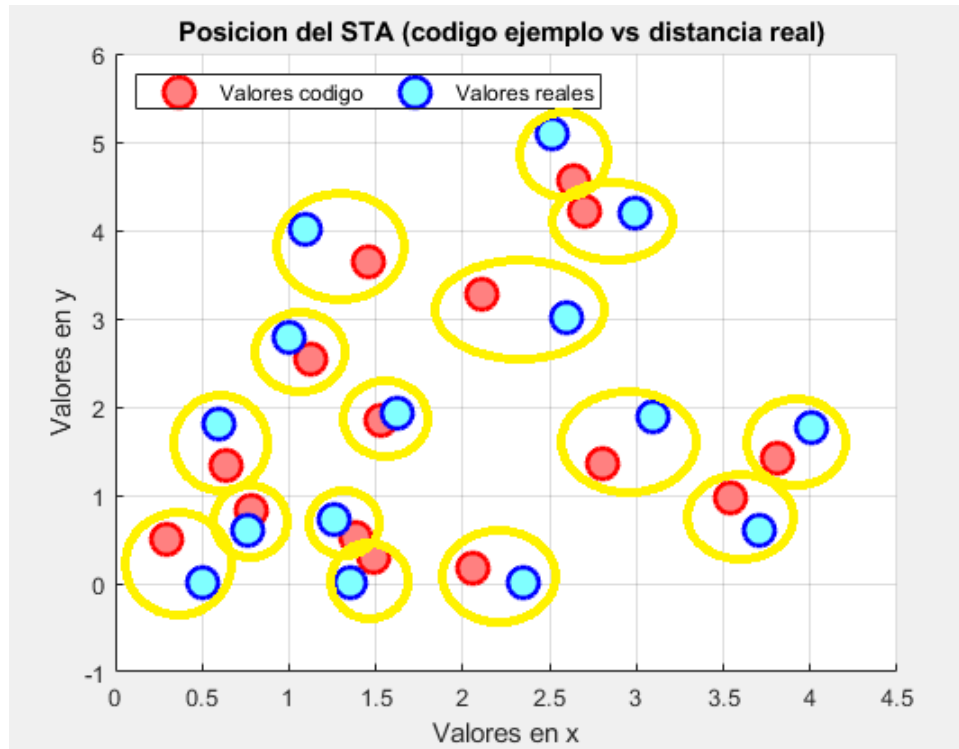
Considerando que los AP se encuentran en los puntos  $(0,0)$ ,  $(4,0)$  y  $(2,6)$ , podemos observar que los valores obtenidos se encuentran dentro del área acorde estimada, la cual se mencionó con anterioridad que correspondería al comedor debido a que este posee la mayor área dentro del espacio de trabajo. Además podemos observar que algunos valores se encuentran muy cercanos entre éstos, lo que se debe a que al no poseer un lugar ideal para trabajar y tanto los AP como el STA no se encuentran en posiciones exactas, se producen diferentes errores dentro del proceso de medición.

Ahora que ya se posee la ubicación del STA para cada punto de estudio en función del sistema FTM, se realizó una nueva medición de posicionamiento considerando los valores reales de las posiciones que se emplearon en el estudio, dichas mediciones no se realizaron con el código ejemplo para así poder verificar si la utilización del protocolo FTM permite la obtención de buenos resultados

Posición	AP 1	AP 2	AP 3	STA
1	0.5	3.5	6.18	(0.5,0.0048)
2	1.35	2.65	6.04	(1.35,0.004925)
3	2.35	1.65	6	(2.35,0.0102083)
4	2.34	3.90	5.7	(0.767038,0.600054)
5	1.71	2.97	5.41	(1.2629,0.717033)
6	3.95	0.61	5.72	(3.7038,0.605742)
7	1.9	3.85	4.43	(0.598437,1.79928)
8	2.06	3.5	4.28	(0.9992,2.7987)
9	2.84	3.33	4.65	(1.62209,1.93824)
10	4.39	1.8	4.54	(4.00401,1.77947)
11	3.5	2.01	4.39	(3.0955,1.88421)
12	4.39	4.57	2.01	(1.09984,4.0032)
13	4.77	4.24	2.09	(2.59691,3.09976)
14	5.16	4.32	2.06	(2.9954,4.20003)
15	5.19	5.12	1.75	(2.51271,5.10023)

**Tabla 4.3:** Tabla de datos para la posición real.

Posteriormente, se realiza el gráfico de los valores determinados por el código vs los valores reales



**Fig. 4.6:** Gráfica de las posiciones obtenidas por el código vs los valores reales.

Podemos observar del gráfico 4.6 que los valores correspondientes a la posición real estarán representados por círculos celestes, en los cuales se puede apreciar que se encuentran en su totalidad dentro del área que abarcan los AP. Por otro lado, los valores obtenidos por el código empleado en esta memoria estarán representados por el color rojo. Como se puede apreciar y considerando que el sistema de medición no era un sistema ideal, los valores obtenidos a través del protocolo FTM son muy similares a los valores reales de los diferentes puntos de estudio, lo que indica el buen uso que este podrá tener ha futuro en el ámbito de la detección.

Finalmente, se tabularon los resultados obtenidos por el STA correspondiente al código y al STA en función a la distancia real de la medición. Es posible percibir que estos resultados son muy parecidos entre ellos, corroborado por la columna error, la cual muestra la diferencia de la posición para el eje x como para el eje y de cada posición las cuales muestran un valor bastante pequeño de error, indicando lo efectivo que funciona el sistema implementado.

Posición	STA (código )	STA (distancia real)	Error
1	(0.339988,0.539171)	(0.5,0.0048)	0,55
2	(0.65525,1.385417)	(0.598437,1.79928)	0,42
3	(0.7832,0.821067)	(0.767038,0.600054)	0,22
4	(1.12563,2.57171)	(0.9992,2.7987)	0,26
5	(1.38125,0.51375)	(1.2629,0.717033)	0,23
6	(1.493,0.3184)	(1.35,0.004925)	0,33
7	(2.05719,0.1693)	(2.35,0.00102083)	0,34
8	(2.80889,1.325)	(3.0955,1.88421)	0,62
9	(3.514,0.9632)	(3.7038,0.605742)	0,40
10	(3.84,1.413)	(4.00401,1.77947)	0,41
11	(1.525,1.848)	(1.62209,1.93824)	0,14
12	(1.4529,3.6476)	(1.0984,4.0032)	0,54
13	(2.1,3.22)	(2.59691,3.09976)	0,51
14	(2.70245,4.213317)	(2.9954,4.20003)	0,20
15	(2.6422,4.526)	(2.51271,5.10023)	0,59

**Tabla 4.4:** Comparación código vs distancia real.

# Capítulo 5

## Conclusiones

### 5.1. Sumario

En esta memoria se quiso hacer incapié en un nuevo sistema de detección de interiores, para el cual, se implementó un sistema simple y económico, el cual, fue capaz de implementar de gran manera la obtención de datos correspondientes al posicionamiento de un STA. Si bien queda mucho por mejorar al respecto, es claro ver que esta primera ida del concepto funciona bien.

### 5.2. Conclusiones

A modo de conclusión del presente informe se observó un nuevo tema dentro de las telecomunicaciones que podría, a futuro, convertirse en uno de los temas más importantes para la localización en interiores, debido a las diferentes posibilidades que ésta posee y de las diferentes formas en las cuales se puede seguir mejorando dicho concepto para así poder emplearse en muchas aplicaciones actuales y futuras, como lo son las aplicaciones IoT, realidad virtual, entre otras. Si bien en el presente informe se abarcaron temas básicos con respecto al ámbito de FTM, se lograron resultados bastantes prometedores con relación al concepto de detección de interiores, logrando obtener valores dentro de un intervalo bastante cercano a los valores reales que se tenían de la posición.



### 5.3. Trabajo Futuro

Se espera a futuro poder mejorar diferentes factores que fueron relevantes al momento de la obtención de datos, los cuales con un mayor conocimiento por parte del alumno y con mayor información al respecto por parte de esta nueva tecnología, podrían lograr que esta tecnología pueda posicionarse como una de las mejores técnica para el proceso de detección.

1. El poder tener una mayor cantidad de ESP permitirá determinar de mejor manera la posición del STA, esto se debe a que al haber mas sensores para la obtención de datos, estos tendrán menos errores al momento de realizar la medición.
2. Implementar un sistema en el cual el individuo vea en tiempo real el movimiento del STA y este sea detectado con precisión.
3. Mejorar el código base para que este pueda realizar la comunicación instantánea al reconocer un AP.

# Bibliografía

- [1] Betzi Mabel Paccha Colala, "Caracterización de la métrica RTT para la implementación de algoritmos de localización de smartphones en entornos indoor", Trabajo de fin de grado, ingeniera en Telemática, 2020, Tecnologías de la Información y las Comunicaciones, Escuela Técnica Superior de Ingeniería de Telecomunicación. <https://repositorio.upct.es/handle/10317/8853>
  
- [2] Daniel Barba, Ana Moretón-Fernández, Héctor del Campo-Pando, Sergio García-Villanueva, Juan March, Diego R. Llanos, "Uso de Bluetooth de Baja Energía en Aplicaciones de Localización Indoor ", <https://www.infor.uva.es/~diego/docs/JP2017-XtremeLoc.pdf>
  
- [3] ATRIA Innovation, "Sistemas de localización en interiores", <https://www.atriainnovation.com/sistemas-localizacion-interiores/>
  
- [4] GPS World, "Cómo lograr una precisión de 1 metro en Android", <https://www.gpsworld.com/how-to-achieve-1-meter-accuracy-in-android/>
  
- [5] Leor Banin, Uri Schatzberg, Yuval Amizur – Intel Corp, "Next Generation Indoor Positioning System Based on WiFi Time of Flight", septiembre 2013 [https://www.researchgate.net/publication/273443111\\_Next\\_Generation\\_Indoor\\_Positioning\\_System\\_Based\\_on\\_WiFi\\_Time\\_of\\_Flight](https://www.researchgate.net/publication/273443111_Next_Generation_Indoor_Positioning_System_Based_on_WiFi_Time_of_Flight)
  
- [6] Berthold K. P. Horn, "Observation Model for Indoor Positioning", Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA, 20 de julio 2020 [https://people.csail.mit.edu/bkph/articles/Observation\\_Model.pdf](https://people.csail.mit.edu/bkph/articles/Observation_Model.pdf)

- [7] Berthold K. P. Horn ,“ Doubling the Accuracy of Indoor Positioning:Frequency Diversity", Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA, 9 de marzo 2020. <https://www.mdpi.com/1424-8220/20/5/1489>
- [8] Markus Bullmann (1),Toni Fetzner (1), Frank Ebner (1), Markus Ebner (1), Frank Deinzer (1) and Marcin Grzegorzec (2) , “Comparison of 2.4 GHz WiFi FTM- and RSSI-Based Indoor Positioning Methods in Realistic Scenarios", (1)Faculty of Computer Science and Business Information Systems, University of Applied Sciences Würzburg-Schweinfurt, 97070 Würzburg, Germany;(2) Institute of Medical Informatics, University of Lübeck, 23547 Lübeck, Germany, 12 de agosto del 2020. <https://www.mdpi.com/1424-8220/20/16/4515>
- [9] Guia de programación para ESP32-S2, espressif, Espressif Systems (Shanghai) Co., Ltd, 2016 - 2021 <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/get-started/index.html>
- [10] RSSI-Based Indoor Localization With the Internet of Things, espressif, SEBASTIAN SADOWSKI, (Student Member, IEEE), PETROS SPACHOS , (Senior Member, IEEE),School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada,June 20, 2018. <https://ieeexplore.ieee.org/document/8371230>
- [11] SISTEMA DE POSICIONAMIENTO GLOBAL (GPS): DESCRIPCIÓN, ANÁLISIS DE ERRORES, APLICACIONES Y FUTURO, Dpto. de Tecnología Electrónica E.T.S. Ingenieros de Telecomunicación. Universidad de Málaga [http://www.oocities.org/es/foro\\_gps/infografia/gps5.pdf](http://www.oocities.org/es/foro_gps/infografia/gps5.pdf)

# Anexo

Código ejemplo Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% codigo ejemplo
```

```
% coordenadas de origen del AP A
```

```
a1 = 0;
```

```
a2 = 0;
```

```
% coordenadas de origen del AP B
```

```
p = 4;
```

```
b1= 0;
```

```
% coordenadas de origen del AP C
```

```
q = 2;
```

```
r = 6;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Distancia entre los AP y el STA (radio de la circunferencia)
```

```
e = 2.6; % AP1
```

```
f = 3.25; % AP2
```

```
g = 4.3; % AP3
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Ecuacion para determinar x e y
```

```
x = ((e^2) - (f^2) + (p^2))/double(2.*p);
```

```
y = ((e^2-g^2+q^2+r^2)/(2.*r))-((q/r).*x);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% visualizacion de la distancia
```

```
posicionap1 = 'La posicion entre el AP1 y STA es:%g \n';
```

```
fprintf(posicionap1,e);
```

```
posicionap2 = 'La posicion entre el AP2 y STA es:%g \n';
```

```
fprintf(posicionap2,f);
```

```
posicionap3 = 'La posicion entre el AP3 y STA es:%g \n';
```

```
fprintf(posicionap3,g);
```

```
posicionfinal = 'La posicion final del STA es:(%g,%g) \n';
```

```
fprintf(posicionfinal,x,y);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Grafico de la posicion del STA
```

```
Q = [0.3,0.639988,0.7832,1.124584,1.38125,1.493,1.52469,1.45525,2.05719,2.10763,
2.6422,2.70245,2.80889,3.545,3.814];
```

```
W = [0.5,1.339171,0.821067,2.53219,0.51375,0.30184,1.8476,3.6485417,0.1693,3.27171,
4.5526,4.213317,1.3625,0.96857,1.4132];
```

```
Q1 = [0.5,0.598437,0.767038,0.9992,1.2629,1.35,1.62209,1.0984,2.35,2.59691,2.9954,
2.51271,3.0955,3.7038,4.00401];
```

```
W1 = [0.0048,1.79928,0.600054,2.79,0.717,0.004,1.938,4.003,0.0010,3,4.2,5.1,1.88,
0.605,1.77];
```

```
figure(1)
```

```
plot(Q,W,'o', 'LineWidth',2,'MarkerSize',13,'MarkerEdgeColor','r','MarkerFaceColor',
[1,0.5,0.5])
```

```
axis([0,4.5,-1,6]);
```

```
xlabel('Valores en x')
```

```
ylabel('Valores en y')
```

```
title('Valores de STA por trilateracion')
```

```
grid on;
```

```
figure(2)
```

```
hold on;
```

```
plot(Q,W,'o', 'LineWidth',2,'MarkerSize',13,'MarkerEdgeColor','r','MarkerFaceColor',
[1,0.5,0.5])
```

```
plot(Q1,W1,'o', 'LineWidth',2,'MarkerSize',13,'MarkerEdgeColor','b','MarkerFaceColor',
[0.5,1,1])
```

```
legend({'Valores codigo','Valores reales'},'Location','northwest',
'Orientation','horizontal')
hold off;
axis([0,4.5,-1,6]);
xlabel('Valores en x')
ylabel('Valores en y')
title('Posicion del STA (codigo ejemplo vs distancia real) ')
grid on;
```

Código ejemplo FTM

```
/* Wi-Fi FTM Example
```

This example code is in the Public Domain (or CC0 licensed, at your option.)

Unless required by applicable law or agreed to in writing, this software is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

```
*/
```

```
#include <errno.h>
#include <string.h>
#include <stdio.h>
#include <string.h>
#include "nvs_flash.h"
#include "cmd_system.h"
```

```
#include "argtable3/argtable3.h"
#include "freertos/FreeRTOS.h"
#include "freertos/event_groups.h"
#include "esp_event.h"
#include "esp_log.h"
#include "esp_err.h"
#include "esp_wifi.h"
#include "esp_console.h"
```

```
typedef struct {
    struct arg_str *ssid;
    struct arg_str *password;
    struct arg_end *end;
} wifi_args_t;
```

```
typedef struct {
    struct arg_str *ssid;
    struct arg_end *end;
} wifi_scan_arg_t;
```

```
typedef struct {
    struct arg_lit *mode;
    struct arg_int *frm_count;
    struct arg_int *burst_period;
    struct arg_str *ssid;
    struct arg_end *end;
```



```
} wifi_ftm_args_t;

static wifi_args_t sta_args;
static wifi_args_t ap_args;
static wifi_scan_arg_t scan_args;
static wifi_ftm_args_t ftm_args;

static bool s_reconnect = true;
static const char *TAG_STA = "ftm_station";
static const char *TAG_AP = "ftm_ap";

static EventGroupHandle_t wifi_event_group;
const int CONNECTED_BIT = BIT0;
const int DISCONNECTED_BIT = BIT1;

static EventGroupHandle_t ftm_event_group;
const int FTM_REPORT_BIT = BIT0;
const int FTM_FAILURE_BIT = BIT1;
wifi_ftm_report_entry_t *g_ftm_report;
uint8_t g_ftm_report_num_entries;
static uint32_t g_rtt_est, g_dist_est;

const int g_report_lvl =
#ifdef CONFIG_ESP_FTM_REPORT_SHOW_DIAG
    BIT0 |
#endif
#ifdef CONFIG_ESP_FTM_REPORT_SHOW_RTT
    BIT1 |
#endif
#ifdef CONFIG_ESP_FTM_REPORT_SHOW_T1T2T3T4
```

```
        BIT2 |
#endif
#ifdef CONFIG_ESP_FTM_REPORT_SHOW_RSSI
        BIT3 |
#endif
0;

uint16_t g_scan_ap_num;
wifi_ap_record_t *g_ap_list_buffer;

static void wifi_connected_handler(void *arg, esp_event_base_t event_base,
                                   int32_t event_id, void *event_data)
{
    wifi_event_sta_connected_t *event = (wifi_event_sta_connected_t *)event_data;

    ESP_LOGI(TAG_STA, "Connected to %s (BSSID: "MACSTR", Channel: %d)", event->ssid,
              MAC2STR(event->bssid), event->channel);

    xEventGroupClearBits(wifi_event_group, DISCONNECTED_BIT);
    xEventGroupSetBits(wifi_event_group, CONNECTED_BIT);
}

static void disconnect_handler(void *arg, esp_event_base_t event_base,
                               int32_t event_id, void *event_data)
{
    if (s_reconnect) {
        ESP_LOGI(TAG_STA, "sta disconnect, s_reconnect...");
        esp_wifi_connect();
    } else {
        ESP_LOGI(TAG_STA, "sta disconnect");
    }
}
```

```
    }
    xEventGroupClearBits(wifi_event_group, CONNECTED_BIT);
    xEventGroupSetBits(wifi_event_group, DISCONNECTED_BIT);
}

static void ftm_report_handler(void *arg, esp_event_base_t event_base,
                               int32_t event_id, void *event_data)
{
    wifi_event_ftm_report_t *event = (wifi_event_ftm_report_t *) event_data;

    if (event->status == FTM_STATUS_SUCCESS) {
        g_rtt_est = event->rtt_est;
        g_dist_est = event->dist_est;
        g_ftm_report = event->ftm_report_data;
        g_ftm_report_num_entries = event->ftm_report_num_entries;
        xEventGroupSetBits(ftm_event_group, FTM_REPORT_BIT);
    } else {
        ESP_LOGI(TAG_STA, "FTM procedure with Peer("MACSTR") failed! (Status - %d)",
                 MAC2STR(event->peer_mac), event->status);
        xEventGroupSetBits(ftm_event_group, FTM_FAILURE_BIT);
    }
}

static void ftm_process_report(void)
{
    int i;
    char *log = malloc(200);

    if (!g_report_lvl)
        return;
}
```

```

if (!log) {
    ESP_LOGE(TAG_STA, "Failed to alloc buffer for FTM report");
    return;
}

bzero(log, 200);
sprintf(log, "%s%s%s%s", g_report_lvl & BIT0 ? " Diag |":"", g_report_lvl & BIT1 ? "
        g_report_lvl & BIT2 ? "          T1          |          T2          |          T3
        g_report_lvl & BIT3 ? "  RSSI  |":"");
ESP_LOGI(TAG_STA, "FTM Report:");
ESP_LOGI(TAG_STA, "|%s", log);
for (i = 0; i < g_ftm_report_num_entries; i++) {
    char *log_ptr = log;

    bzero(log, 200);
    if (g_report_lvl & BIT0) {
        log_ptr += sprintf(log_ptr, "%6d|", g_ftm_report[i].dlog_token);
    }
    if (g_report_lvl & BIT1) {
        log_ptr += sprintf(log_ptr, "%7u |", g_ftm_report[i].rtt);
    }
    if (g_report_lvl & BIT2) {
        log_ptr += sprintf(log_ptr, "%14llu |%14llu |%14llu |%14llu |", g_ftm_re
                                g_ftm_report[i].t2, g_ftm_report[i].t3, g_ftm_re
    }
    if (g_report_lvl & BIT3) {
        log_ptr += sprintf(log_ptr, "%6d |", g_ftm_report[i].rssi);
    }
    ESP_LOGI(TAG_STA, "|%s", log);
}

```

```
    }
    free(log);
}

void initialise_wifi(void)
{
    esp_log_level_set("wifi", ESP_LOG_WARN);
    static bool initialized = false;

    if (initialized) {
        return;
    }

    ESP_ERROR_CHECK(esp_netif_init());
    wifi_event_group = xEventGroupCreate();
    ftm_event_group = xEventGroupCreate();
    ESP_ERROR_CHECK( esp_event_loop_create_default() );
    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
        WIFI_EVENT_STA_CONNECTED,
        &wifi_connected_handler,
        NULL,
        NULL));
    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
        WIFI_EVENT_STA_DISCONNECTED,
        &disconnect_handler,
        NULL,
        NULL));
    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
```

```
        WIFI_EVENT_FTM_REPORT,
        &ftm_report_handler,
        NULL,
        NULL));
    ESP_ERROR_CHECK(esp_wifi_set_storage(WIFI_STORAGE_RAM) );
    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_NULL) );
    ESP_ERROR_CHECK(esp_wifi_start() );
    initialized = true;
}

static bool wifi_cmd_sta_join(const char *ssid, const char *pass)
{
    int bits = xEventGroupWaitBits(wifi_event_group, CONNECTED_BIT, 0, 1, 0);

    wifi_config_t wifi_config = { 0 };

    strncpy((char *) wifi_config.sta.ssid, ssid, sizeof(wifi_config.sta.ssid));
    if (pass) {
        strncpy((char *) wifi_config.sta.password, pass, sizeof(wifi_config.sta.password) );
    }

    if (bits & CONNECTED_BIT) {
        s_reconnect = false;
        xEventGroupClearBits(wifi_event_group, CONNECTED_BIT);
        ESP_ERROR_CHECK( esp_wifi_disconnect() );
        xEventGroupWaitBits(wifi_event_group, DISCONNECTED_BIT, 0, 1, portTICK_RATE_MS);
    }

    s_reconnect = true;
    ESP_ERROR_CHECK( esp_wifi_set_mode(WIFI_MODE_STA) );
}
```

```
ESP_ERROR_CHECK( esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config) );
ESP_ERROR_CHECK( esp_wifi_connect() );

xEventGroupWaitBits(wifi_event_group, CONNECTED_BIT, 0, 1, 5000 / portTICK_RATE_MS);

return true;
}

static int wifi_cmd_sta(int argc, char **argv)
{
    int nerrors = arg_parse(argc, argv, (void **) &sta_args);

    if (nerrors != 0) {
        arg_print_errors(stderr, sta_args.end, argv[0]);
        return 1;
    }

    ESP_LOGI(TAG_STA, "sta connecting to '%s'", sta_args.ssid->sval[0]);
    wifi_cmd_sta_join(sta_args.ssid->sval[0], sta_args.password->sval[0]);
    return 0;
}

static bool wifi_perform_scan(const char *ssid, bool internal)
{
    wifi_scan_config_t scan_config = { 0 };
    scan_config.ssid = (uint8_t *) ssid;
    uint8_t i;

    ESP_ERROR_CHECK( esp_wifi_set_mode(WIFI_MODE_STA) );
    ESP_ERROR_CHECK( esp_wifi_scan_start(&scan_config, true) );
}
```

```
esp_wifi_scan_get_ap_num(&g_scan_ap_num);
if (g_scan_ap_num == 0) {
    ESP_LOGI(TAG_STA, "No matching AP found");
    return false;
}

if (g_ap_list_buffer) {
    free(g_ap_list_buffer);
}

g_ap_list_buffer = malloc(g_scan_ap_num * sizeof(wifi_ap_record_t));
if (g_ap_list_buffer == NULL) {
    ESP_LOGE(TAG_STA, "Failed to malloc buffer to print scan results");
    return false;
}

if (esp_wifi_scan_get_ap_records(&g_scan_ap_num, (wifi_ap_record_t *)g_ap_list_buffer)
    if (!internal) {
        for (i = 0; i < g_scan_ap_num; i++) {
            ESP_LOGI(TAG_STA, "[%s][rssi=%d]""%s", g_ap_list_buffer[i].ssid, g_ap_list_buffer[i].rssi,
                g_ap_list_buffer[i].ftm_responder ? "[FTM Responder]" : "");
        }
    }
}

ESP_LOGI(TAG_STA, "sta scan done");

return true;
}
```



```
static int wifi_cmd_scan(int argc, char **argv)
{
    int nerrors = arg_parse(argc, argv, (void **) &scan_args);

    if (nerrors != 0) {
        arg_print_errors(stderr, scan_args.end, argv[0]);
        return 1;
    }

    ESP_LOGI(TAG_STA, "sta start to scan");
    if ( scan_args.ssid->count == 1 ) {
        wifi_perform_scan(scan_args.ssid->sval[0], false);
    } else {
        wifi_perform_scan(NULL, false);
    }
    return 0;
}

static bool wifi_cmd_ap_set(const char* ssid, const char* pass)
{
    wifi_config_t wifi_config = {
        .ap = {
            .ssid = "",
            .ssid_len = 0,
            .max_connection = 4,
            .password = "",
            .authmode = WIFI_AUTH_WPA2_PSK,
            .ftm_responder = true
        },
    };
};
```

```
s_reconnect = false;
strncpy((char*) wifi_config.ap.ssid, ssid, sizeof(wifi_config.ap.ssid));
if (pass) {
    if (strlen(pass) != 0 && strlen(pass) < 8) {
        s_reconnect = true;
        ESP_LOGE(TAG_AP, "password less than 8");
        return false;
    }
    strncpy((char*) wifi_config.ap.password, pass, sizeof(wifi_config.ap.password));
}

if (strlen(pass) == 0) {
    wifi_config.ap.authmode = WIFI_AUTH_OPEN;
}

ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_AP));
ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_AP, &wifi_config));
return true;
}

static int wifi_cmd_ap(int argc, char** argv)
{
    int nerrors = arg_parse(argc, argv, (void**) &ap_args);

    if (nerrors != 0) {
        arg_print_errors(stderr, ap_args.end, argv[0]);
        return 1;
    }
}
```

```
wifi_cmd_ap_set(ap_args.ssid->sval[0], ap_args.password->sval[0]);
ESP_LOGI(TAG_AP, "Starting SoftAP with FTM Responder support, SSID - %s, Password -
return 0;
}

static int wifi_cmd_query(int argc, char **argv)
{
    wifi_config_t cfg;
    wifi_mode_t mode;

    esp_wifi_get_mode(&mode);
    if (WIFI_MODE_AP == mode) {
        esp_wifi_get_config(WIFI_IF_AP, &cfg);
        ESP_LOGI(TAG_AP, "AP mode, %s %s", cfg.ap.ssid, cfg.ap.password);
    } else if (WIFI_MODE_STA == mode) {
        int bits = xEventGroupWaitBits(wifi_event_group, CONNECTED_BIT, 0, 1, 0);
        if (bits & CONNECTED_BIT) {
            esp_wifi_get_config(WIFI_IF_STA, &cfg);
            ESP_LOGI(TAG_STA, "sta mode, connected %s", cfg.ap.ssid);
        } else {
            ESP_LOGI(TAG_STA, "sta mode, disconnected");
        }
    } else {
        ESP_LOGI(TAG_STA, "NULL mode");
        return 0;
    }

    return 0;
}
```

```
wifi_ap_record_t *find_ftm_responder_ap(const char *ssid)
{
    bool retry_scan = false;
    uint8_t i;

    if (!ssid)
        return NULL;

retry:
    if (!g_ap_list_buffer || (g_scan_ap_num == 0)) {
        ESP_LOGI(TAG_STA, "Scanning for %s", ssid);
        if (false == wifi_perform_scan(ssid, true)) {
            return NULL;
        }
    }

    for (i = 0; i < g_scan_ap_num; i++) {
        if (strcmp((const char *)g_ap_list_buffer[i].ssid, ssid) == 0)
            return &g_ap_list_buffer[i];
    }

    if (!retry_scan) {
        retry_scan = true;
        if (g_ap_list_buffer) {
            free(g_ap_list_buffer);
            g_ap_list_buffer = NULL;
        }
        goto retry;
    }
}
```

```
    ESP_LOGI(TAG_STA, "No matching AP found");

    return NULL;
}

static int wifi_cmd_ftm(int argc, char **argv)
{
    int nerrors = arg_parse(argc, argv, (void **) &ftm_args);
    wifi_ap_record_t *ap_record;

    wifi_ftm_initiator_cfg_t ftmi_cfg = {
        .frm_count = 32,
        .burst_period = 2,
    };

    if (nerrors != 0) {
        arg_print_errors(stderr, ftm_args.end, argv[0]);
        return 0;
    }

    if (ftm_args.mode->count == 0) {
        goto ftm_start;
    }

    if (ftm_args.ssid->count == 1) {
        ap_record = find_ftm_responder_ap(ftm_args.ssid->sval[0]);
        if (ap_record) {
            memcpy(ftmi_cfg.resp_mac, ap_record->bssid, 6);
            ftmi_cfg.channel = ap_record->primary;
        } else {
```

```
        return 0;
    }
}

if (ftm_args.frm_count->count != 0) {
    uint8_t count = ftm_args.frm_count->ival[0];
    if (count != 0 && count != 16 &&
        count != 24 && count != 32 && count != 64) {
        ESP_LOGE(TAG_STA, "Invalid Frame Count! Valid options are 0/16/24/32/64");
        return 0;
    }
    ftmi_cfg.frm_count = count;
}

if (ftm_args.burst_period->count != 0) {
    if (ftm_args.burst_period->ival[0] >= 2 &&
        ftm_args.burst_period->ival[0] < 256) {
        ftmi_cfg.burst_period = ftm_args.burst_period->ival[0];
    } else {
        ESP_LOGE(TAG_STA, "Invalid Burst Period! Valid range is 2-255");
        return 0;
    }
}
}
```

ftm\_start:

```
ESP_LOGI(TAG_STA, "Requesting FTM session with Frm Count - %d, Burst Period - %dmSec
    ftmi_cfg.frm_count, ftmi_cfg.burst_period*100);

if (ESP_OK != esp_wifi_ftm_initiate_session(&ftmi_cfg)) {
    ESP_LOGE(TAG_STA, "Failed to start FTM session");
    return 0;
}

EventBits_t bits = xEventGroupWaitBits(ftm_event_group, FTM_REPORT_BIT | FTM_FAILURE
    pdFALSE, pdFALSE, portMAX_DELAY);

/* Processing data from FTM session */
if (bits & FTM_REPORT_BIT) {
    ftm_process_report();
    free(g_ftm_report);
    g_ftm_report = NULL;
    g_ftm_report_num_entries = 0;
    ESP_LOGI(TAG_STA, "Estimated RTT - %d nSec, Estimated Distance - %d.%02d meters"
        g_rtt_est, g_dist_est / 100, g_dist_est % 100);

    xEventGroupClearBits(ftm_event_group, FTM_REPORT_BIT);
} else {
    /* Failure case */
}

return 0;
}

void register_wifi(void)
{
    sta_args.ssid = arg_str1(NULL, NULL, "<ssid>", "SSID of AP");
```

```
sta_args.password = arg_str0(NULL, NULL, "<pass>", "password of AP");
sta_args.end = arg_end(2);
```

```
const esp_console_cmd_t sta_cmd = {
    .command = "sta",
    .help = "WiFi is station mode, join specified soft-AP",
    .hint = NULL,
    .func = &wifi_cmd_sta,
    .argtable = &sta_args
};
```

```
ESP_ERROR_CHECK( esp_console_cmd_register(&sta_cmd) );
```

```
ap_args.ssid = arg_str1(NULL, NULL, "<ssid>", "SSID of AP");
ap_args.password = arg_str0(NULL, NULL, "<pass>", "password of AP");
ap_args.end = arg_end(2);
```

```
const esp_console_cmd_t ap_cmd = {
    .command = "ap",
    .help = "AP mode, configure ssid and password",
    .hint = NULL,
    .func = &wifi_cmd_ap,
    .argtable = &ap_args
};
```

```
ESP_ERROR_CHECK( esp_console_cmd_register(&ap_cmd) );
```

```
scan_args.ssid = arg_str0(NULL, NULL, "<ssid>", "SSID of AP want to be scanned");
scan_args.end = arg_end(1);
```



```
const esp_console_cmd_t scan_cmd = {
    .command = "scan",
    .help = "WiFi is station mode, start scan ap",
    .hint = NULL,
    .func = &wifi_cmd_scan,
    .argtable = &scan_args
};

ESP_ERROR_CHECK( esp_console_cmd_register(&scan_cmd) );

const esp_console_cmd_t query_cmd = {
    .command = "query",
    .help = "query WiFi info",
    .hint = NULL,
    .func = &wifi_cmd_query,
};

ESP_ERROR_CHECK( esp_console_cmd_register(&query_cmd) );

ftm_args.mode = arg_lit1("I", "ftm_initiator", "FTM Initiator mode");
ftm_args.ssid = arg_str0("s", "ssid", "SSID", "SSID of AP");
ftm_args.frm_count = arg_int0("c", "frm_count", "<0/16/24/32/64>", "FTM frames to be
ftm_args.burst_period = arg_int0("p", "burst_period", "<2-255 (x 100 mSec)>", "Perio
ftm_args.end = arg_end(1);

const esp_console_cmd_t ftm_cmd = {
    .command = "ftm",
    .help = "FTM command",
    .hint = NULL,
    .func = &wifi_cmd_ftm,
    .argtable = &ftm_args
```

```
};

ESP_ERROR_CHECK( esp_console_cmd_register(&ftm_cmd) );
}

void app_main(void)
{
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK( ret );

    initialise_wifi();

    esp_console_repl_t *repl = NULL;
    esp_console_repl_config_t repl_config = ESP_CONSOLE_REPL_CONFIG_DEFAULT();
    esp_console_dev_uart_config_t uart_config = ESP_CONSOLE_DEV_UART_CONFIG_DEFAULT();
    repl_config.prompt = "ftm>";
    // init console REPL environment
    ESP_ERROR_CHECK(esp_console_new_repl_uart(&uart_config, &repl_config, &repl));
    /* Register commands */
    register_system();
    register_wifi();

    printf("\n =====\n");
    printf(" |                               Steps to test FTM                               |\n");
    printf(" |                                                                                   |\n");
    printf(" | 1. Use 'help' to gain overview of commands                                     |\n");
}
```

```
printf(" | 2. Use 'scan' command to search for external AP's      |\n");
printf(" |                                     OR                    |\n");
printf(" | 2. Start SoftAP on another device using 'ap' command  |\n");
printf(" | 3. Start FTM with command 'ftm -I -s <SSID>'          |\n");
printf(" |                                                         |\n");
printf(" =====\n\n");

// start console REPL
ESP_ERROR_CHECK(esp_console_start_repl(repl));}
```

Código automatizado FTM Luciano Ignacio Radriagan Figueroa

```
#include <errno.h>
#include <string.h>
#include <stdio.h>
#include <string.h>
#include "nvs_flash.h"
#include "cmd_system.h"
#include "argtable3/argtable3.h"
#include "freertos/FreeRTOS.h"
#include "freertos/event_groups.h"
#include "esp_event.h"
#include "esp_log.h"
#include "esp_err.h"
#include "esp_wifi.h"
#include "esp_console.h"

static bool s_reconnect = true;
static const char *TAG_AP = "ftm_ap";

static EventGroupHandle_t wifi_event_group;
const int CONNECTED_BIT = BIT0;
const int DISCONNECTED_BIT = BIT1;

static EventGroupHandle_t ftm_event_group;
const int FTM_REPORT_BIT = BIT0;
const int FTM_FAILURE_BIT = BIT1;
wifi_ftm_report_entry_t *g_ftm_report;
uint8_t g_ftm_report_num_entries;
```

```
uint16_t g_scan_ap_num;
wifi_ap_record_t *g_ap_list_buffer;

void initialise_wifi(void)
{
    esp_log_level_set("wifi", ESP_LOG_WARN);
    static bool initialized = false;

    if (initialized) {
        return;
    }

    ESP_ERROR_CHECK(esp_netif_init());
    wifi_event_group = xEventGroupCreate();
    ftm_event_group = xEventGroupCreate();
    ESP_ERROR_CHECK( esp_event_loop_create_default() );
    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
    ESP_ERROR_CHECK(esp_wifi_set_storage(WIFI_STORAGE_RAM) );
    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_NULL) );
    ESP_ERROR_CHECK(esp_wifi_start() );
    initialized = true;
}

static bool wifi_cmd_ap_set(const char* ssid, const char* pass)
{
    ESP_LOGI(TAG_AP, "Starting SoftAP with FTM Responder support, SSID - %s, Password - %s",

    wifi_config_t wifi_config = {
        .ap = {
```

```
.ssid = "",
.ssid_len = 0,
.max_connection = 4,
.password = "",
.authmode = WIFI_AUTH_WPA2_PSK,
.ftm_responder = true
},
};

s_reconnect = false;
strncpy((char*) wifi_config.ap.ssid, ssid, sizeof(wifi_config.ap.ssid));
if (pass) {
    if (strlen(pass) != 0 && strlen(pass) < 8) {
        s_reconnect = true;
        ESP_LOGE(TAG_AP, "password less than 8");
        return false;
    }
    strncpy((char*) wifi_config.ap.password, pass, sizeof(wifi_config.ap.password));
}

if (strlen(pass) == 0) {
    wifi_config.ap.authmode = WIFI_AUTH_OPEN;
}

ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_AP));
ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_AP, &wifi_config));
return true;
}

void app_main(void)
```

```
{
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
        ESP_ERROR_CHECK(nvs_flash_erase());
        ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK( ret );

    initialise_wifi();

    esp_console_repl_t *repl = NULL;
    esp_console_repl_config_t repl_config = ESP_CONSOLE_REPL_CONFIG_DEFAULT();
    esp_console_dev_uart_config_t uart_config = ESP_CONSOLE_DEV_UART_CONFIG_DEFAULT();
    repl_config.prompt = "ftm>";

    // init console REPL environment
    ESP_ERROR_CHECK(esp_console_new_repl_uart(&uart_config, &repl_config, &repl));

    wifi_cmd_ap_set("AP", "apapapap");

    // start console REPL
    ESP_ERROR_CHECK(esp_console_start_repl(repl));
}
```