



UNIVERSIDAD NACIONAL DE LA MATANZA

Departamento de Ingeniería e Investigaciones Tecnológicas

**Ingeniería en Electrónica**

## **Técnicas digitales III**

### **Dispositivo de Localización y Seguimiento**

**Alumno:** Nicolás Agostino

**Profesores:** Lic. Carlos Maidana

Ing. Guillermo Buranits

Ing. Edgardo Alberto Gho

**2023**

# ÍNDICE

<b>1</b>	<b>DESCRIPCIÓN GENERAL .....</b>	<b>1</b>
<b>2</b>	<b>INTRODUCCIÓN TEÓRICA.....</b>	<b>1</b>
2.1	GPS .....	1
2.2	GSM.....	3
2.3	UNIDAD DE MEDICIÓN INERCIAL .....	4
<b>3</b>	<b>DESCRIPCIÓN TÉCNICA DEL PROYECTO .....</b>	<b>5</b>
3.1	HARDWARE.....	5
3.1.1	<i>Diagrama en bloques.....</i>	<i>5</i>
3.1.2	<i>Esquemático .....</i>	<i>6</i>
3.1.3	<i>Circuito.....</i>	<i>9</i>
3.1.4	<i>Fotos .....</i>	<i>9</i>
3.2	SOFTWARE.....	10
3.2.1	<i>Enumeración de rutinas.....</i>	<i>10</i>
3.2.2	<i>Descripción del funcionamiento de cada rutina .....</i>	<i>12</i>
<b>4</b>	<b>MODO DE OPERACIÓN.....</b>	<b>22</b>
4.1	USO DEL DISPOSITIVO.....	22
4.2	ENVÍO DE EVENTOS POR SMS .....	22
4.3	VISUALIZACIÓN DE RECORRIDO REALIZADO .....	24
4.4	DEBUG DEL PROGRAMA.....	28
4.5	CARGA DE BATERÍA .....	29
<b>5</b>	<b>ENSAYOS .....</b>	<b>30</b>
5.1	ACELERÓMETRO / GIRÓSCOPO .....	30
5.2	DATOS DE LOCALIZACIÓN .....	30
5.3	CONSUMO DEL DISPOSITIVO .....	31
<b>6</b>	<b>CONCLUSIONES.....</b>	<b>32</b>
<b>7</b>	<b>PROYECTO FINALIZADO .....</b>	<b>33</b>
<b>8</b>	<b>REFERENCIAS .....</b>	<b>34</b>

# 1 Descripción general

El proyecto consiste en el desarrollo de un localizador GPS con funciones de seguridad y comunicación, diseñado para brindar una solución compacta y portátil de rastreo y protección. Este dispositivo permite a los usuarios conocer su ubicación en tiempo real a través de un sistema de posicionamiento global altamente preciso y enviar esta información mediante SMS. Está especialmente dirigido a personas que salen sin un teléfono celular, como adultos mayores, niños o para el seguimiento de objetos como mochilas o valijas.

El dispositivo incorpora funciones de seguridad clave. Un botón de pánico permite a los usuarios enviar rápidamente un mensaje de alerta a un número de teléfono predefinido en situaciones de emergencia. Además, mediante la detección de movimientos bruscos a través de un acelerómetro, el dispositivo envía automáticamente un mensaje de alerta en caso de detectar actividad anormal, garantizando una respuesta rápida ante posibles peligros.

Adicionalmente, el dispositivo ofrece la capacidad de almacenar datos de recorrido, lo que permite a los usuarios tener un registro detallado del desplazamiento, siendo útil para actividades deportivas o seguimiento de rutas sin poner en riesgo el teléfono celular.

## 2 Introducción teórica

### 2.1 GPS

El GPS (Global Positioning System) es un sistema de radionavegación basado en satélites desarrollado por el Departamento de Defensa de Estados Unidos de América y abierto al uso público en la década de 1980. Este sistema permite a cualquier usuario saber su localización, velocidad y altura, las 24 horas del día, bajo cualquier condición atmosférica y en cualquier punto del planeta. El nombre genérico de esta tecnología es GNSS (Sistema global de navegación por satélite). Además del sistema GPS desarrollado, instalado y utilizado por Estados Unidos, existen también otros sistemas similares como ser el GLONNAS (Federación Rusa), BeiDou (China) y Galileo (Unión Europea).

Para determinar las posiciones, el sistema GPS se sirve de al menos dos satélites y utiliza la trilateración, método matemático para determinar las posiciones relativas de objetos usando la geometría. Para entender cómo funciona el sistema de posicionamiento global, primero se debe saber que el GPS está compuesto por tres componentes: el espacial, el de control y el de usuario. El componente espacial está constituido por una constelación de satélites en órbita terrestre aproximadamente a 20200 km, distribuidos

en 6 planos orbitales. Fue concebido de manera que existan como mínimo 4 satélites visibles por encima del horizonte en cualquier punto de la superficie y en cualquier altura.

El componente de control está constituido por 5 estaciones de rastreo distribuidas a lo largo del globo y una estación de control principal (MCS- Master Control Station). Este componente rastrea los satélites, actualiza sus posiciones orbitales y calibra y sincroniza sus relojes. Otra función importante es la de determinar las órbitas de cada satélite y prever su trayectoria durante las 24 horas siguientes. Esta información es enviada a cada satélite para después ser transmitida por este, informando al receptor local donde es posible encontrar el satélite.

El componente del usuario incluye todos aquellos que usan un receptor GPS para recibir y convertir la señal GPS en posición, velocidad y tiempo. Incluye además todos los elementos necesarios en este proceso, como las antenas y el software de procesamiento.

Los fundamentos básicos del GPS se basan en la determinación de la distancia entre un punto (el receptor), a otros de referencia (los satélites). Conociendo la distancia que separa a un punto de tres puntos de referencia, se puede determinar su posición relativa a esas referencias a través de la intersección de tres circunferencias cuyos radios son las distancias medidas entre el receptor y los satélites. En la realidad, son necesarios como mínimo cuatro satélites para determinar la posición en el globo correctamente. Cada satélite transmite una señal que es recibida por el receptor, y éste, por su parte mide el tiempo que las señales tardan a llegar hasta él. Multiplicando el tiempo medido por la velocidad de la señal (velocidad de la luz), se obtiene la distancia receptor-satélite ( $\text{Distancia} = \text{Velocidad} \times \text{Tiempo}$ ).

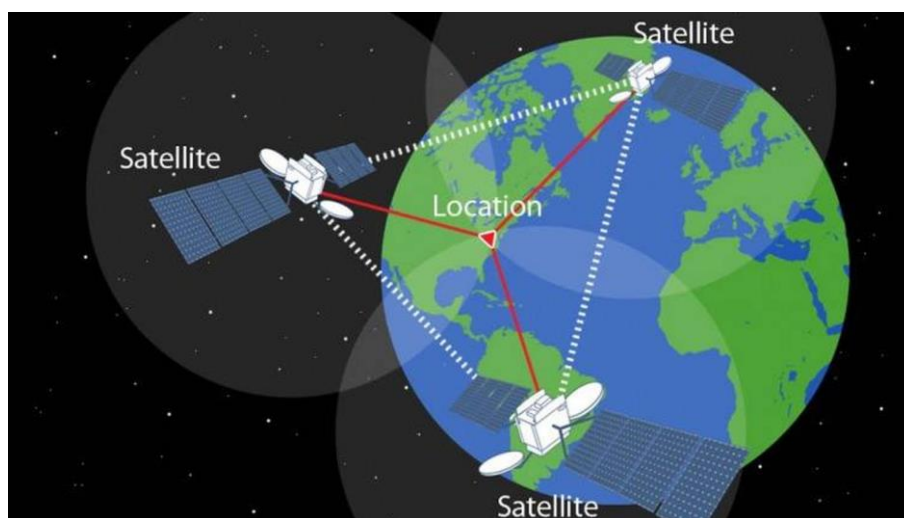


Figura 1 – Método de trilateración

El protocolo del GPS a utilizar soporta la trama NMEA. NMEA 0183 es un protocolo a través del cual los instrumentos de navegación marítima y terrestre pueden comunicarse. Ha sido definido y está controlado por la organización estadounidense National Marine Electronics Association. Esta trama dispone de información como posición, velocidad, dirección de desplazamiento, hora UTC, validez de la señal que se recibe y datos de control.

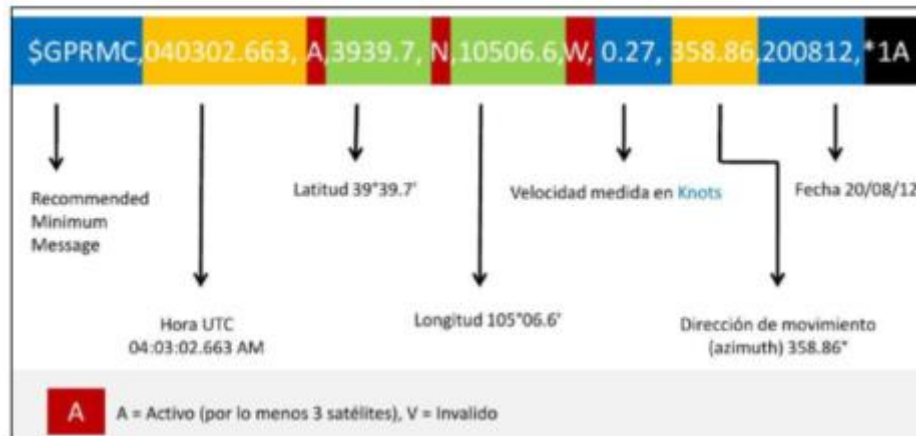


Figura 2 – Trama GPRMC

## 2.2 GSM

El GSM (Global System for Mobile communications), desarrollado a partir de 1982, es el sistema de comunicaciones que más se utiliza en teléfonos celulares. Introducido por primera vez en la década de 1990, este sistema revolucionó la telefonía al permitir la transmisión de voz y datos a través de una red digital. Sus características clave incluyen la conmutación de circuitos para voz y la capacidad de enviar mensajes de texto, sentando las bases para las generaciones posteriores de tecnologías móviles.

Un módulo GSM puede controlarse mediante comandos AT. Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal modem, denominados así por la abreviatura de "Attention".

En un principio, el juego de comandos AT fue desarrollado en 1977 por Dennis Hayes como una interfaz de comunicación con un modem para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, el juego de comandos se fue desarrollando y expandiendo hasta universalizarlo. Hoy en día la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales.

De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales. Este juego de instrucciones puede encontrarse en la documentación técnica de los terminales GSM y permite acciones tales como realizar llamadas de datos o de

voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

## 2.3 Unidad de medición inercial

Una unidad de medición inercial (IMU) es un dispositivo que puede medir e informar la gravedad específica y la velocidad angular de un objeto al que está conectado. Una IMU normalmente consta de giroscopios, proporcionando una medida de velocidad angular, y acelerómetros, que proporcionan una medida de fuerza/aceleración específica.

Una IMU detecta la aceleración de la gravedad terrestre. Gracias a esto, sus lecturas permiten conocer cuál es el ángulo de inclinación respecto al eje X o eje Y. Conociendo que la gravedad es  $9,8 \text{ m/s}^2$ , y observando los valores de los tres ejes del acelerómetro, por trigonometría es posible calcular el ángulo de inclinación del dispositivo. Para ello se utilizan las siguientes fórmulas:

$$\text{AnguloY} = \tan^{-1} \left( \frac{x}{\sqrt{y^2 + z^2}} \right)$$

$$\text{AnguloX} = \tan^{-1} \left( \frac{y}{\sqrt{x^2 + z^2}} \right)$$

Respecto al giróscopo, éste mide la velocidad angular, es decir, la cantidad de grados que gira en un segundo. Conociendo el ángulo inicial de la IMU, se puede sumar el valor que marca el giróscopo para saber el nuevo ángulo a cada momento, obteniendo el ángulo de la siguiente manera:

$$\text{AnguloY} = \text{AnguloYanterior} + x * \Delta t$$

Siendo  $\Delta t$  el tiempo que transcurre cada vez que se calcula esta fórmula.

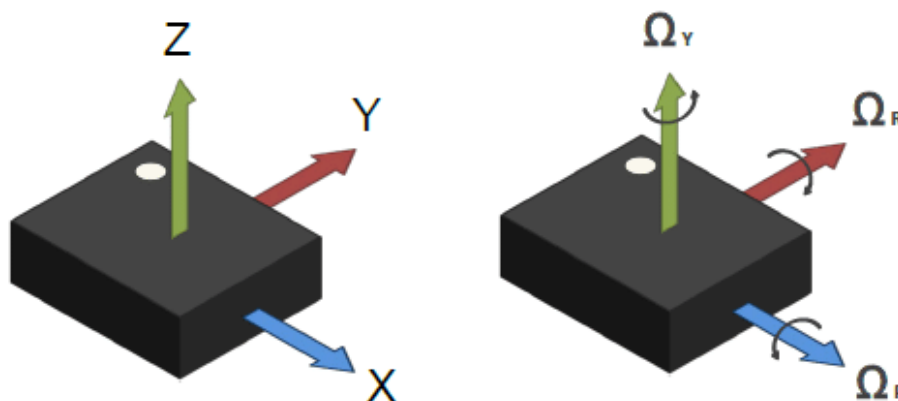


Figura 3 – Ejes de acelerómetro y ángulos de giróscopo

## 3 Descripción técnica del proyecto

A continuación, se pueden observar en detalle las características técnicas del proyecto.

### 3.1 Hardware

Se describe en esta sección el funcionamiento de la electrónica asociada al proyecto.

#### 3.1.1 Diagrama en bloques

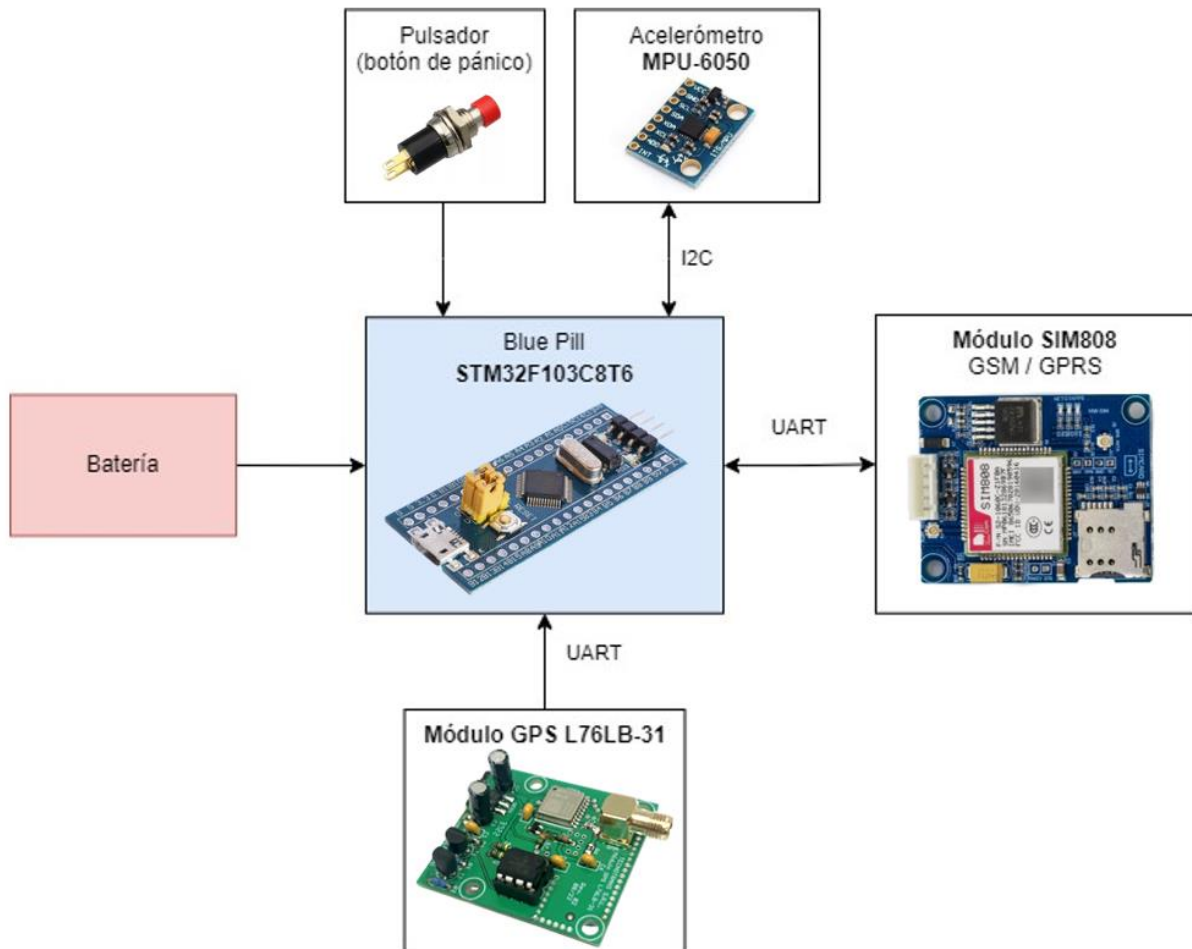


Figura 4 – Diagrama en bloques del circuito

El corazón del dispositivo será el microcontrolador STM32F103C8T6 (Blue Pill), que controlará y coordinará todas las funciones del sistema, incluyendo la interacción con los módulos y los sensores. Se verificará el accionamiento del pulsador de pánico y se medirá el movimiento a través del acelerómetro/giróscopo. El módulo GPS se encargará de proporcionar la ubicación del equipo, mientras que el módulo SIM808 será el encargado de informar eventos a través de mensajes de texto, como así también guardará datos de recorrido. El sistema en su conjunto será alimentado con una batería de litio recargable, lo que garantiza su portabilidad y autonomía.

### 3.1.2 Esquemático

#### Pulsador de Pánico

El pulsador SW1 será presionado en caso de querer generar un evento de pánico. La elección de conectarlo con una resistencia de pull-up de  $10k\ \Omega$  (R1) a 3,3 V se justifica por varias razones. En primer lugar, esta configuración proporciona un valor alto lógico cuando el botón no está presionado, lo cual es fundamental para evitar fluctuaciones no deseadas en el estado de la señal debido a posibles ruidos eléctricos. Además, al utilizar un valor de resistencia de  $10k\ \Omega$ , se logra un equilibrio entre la eficiencia energética y la corriente necesaria para cambiar el estado del pin cuando se presiona el botón.

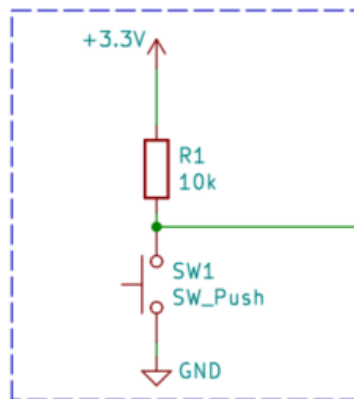


Figura 4 – Pulsador de Pánico en el circuito esquemático

#### Módulo acelerómetro/giróscopo MPU6050

El MPU-6050 es una unidad de medición inercial (IMU) de seis grados de libertad (6DOF) fabricado por Invensense, que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Este módulo (U3) se alimentará con la salida de 3,3 V salientes de la Blue Pill. Sus pines SCL y SDA se conectarán con los pines PB10 (SCL2) y PB11 (SDA2) del microcontrolador, comunicándose por I2C.

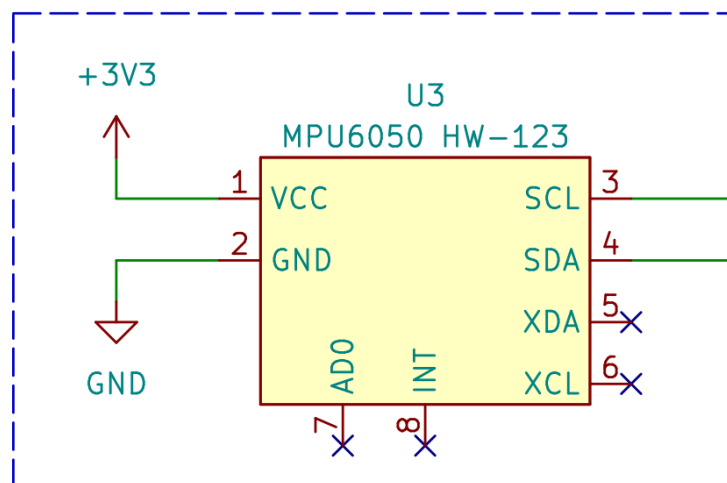


Figura 5 – Módulo acelerómetro/giróscopo MPU6050 en el circuito esquemático



### **Módulo SIM808**

El módulo SIM808 (U2) proporciona la tecnología de comunicación GSM. Requiere para operar de una tarjeta SIM y su correspondiente antena y es controlado por el microcontrolador mediante comandos denominados “AT” por comunicación serie a través de la USART2 a una velocidad de 9600 Baudios. Los comandos AT permitirán enviar y recibir mensajes SMS, establecer la conexión GSM y guardar información en su sistema de archivos. Este módulo se alimentará directamente de la batería, ya que como indica su hoja de datos, se debe alimentar con tensión desde 3,4 V a 4,4 V. Los pines RXD y TXD se conectarán a los pines PA2 (TX2) y PA3 (RX2) de la Blue Pill, respectivamente. Además, se agrega el conector J2, el cual será útil para mandar comandos directamente al módem desde la PC conectando un conversor USB a UART TTL.

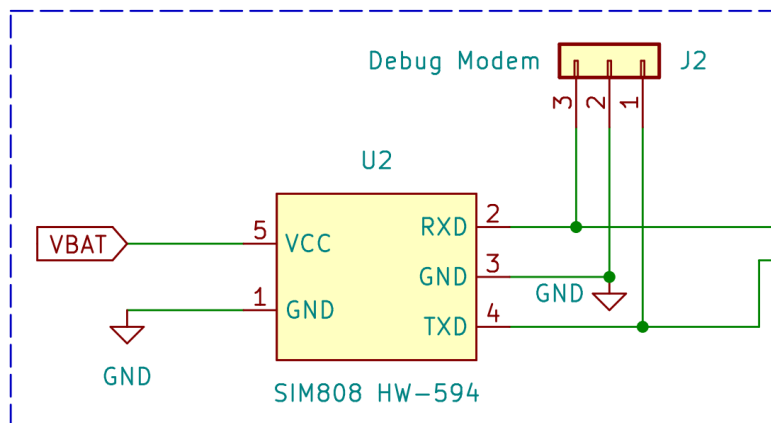


Figura 6 – Módulo SIM808 en el circuito esquemático

### **Módulo GPS L76LB-31**

Este módulo U4 es alimentado por la tensión de la batería y su salida TXD se conecta con el pin PB7 (RX1) de la Blue Pill. Este módulo procesa internamente los datos obtenidos por el satélite y lo comunica sin necesidad de recibir comandos, en el formato de trama GPRMC. La comunicación se establece a una velocidad de 4800 Baudios.

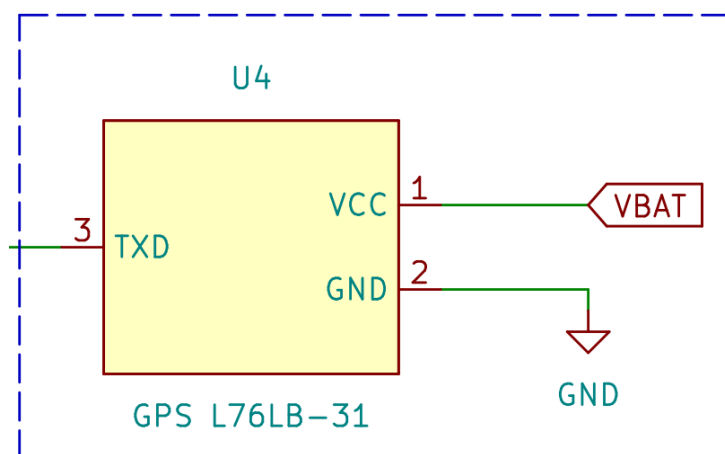


Figura 7 – Módulo GPS L76LB-31 en el circuito esquemático

Es por eso que el pin PB6 (TX1) de la Blue Pill se lo aprovecha en el conector J1 para poder hacer el *debug* del dispositivo.

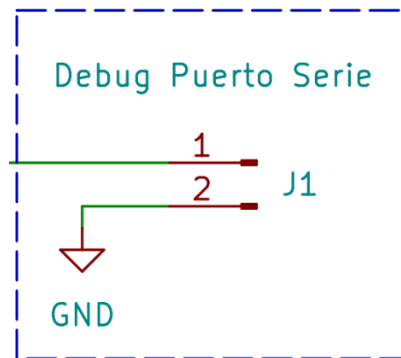


Figura 8 – Puerto serie para debug en el circuito esquemático

### Alimentación y módulo de carga

Para energizar el dispositivo, se utiliza una batería de litio recargable. La misma proporciona una tensión de 3,7 V y una capacidad de 1000 mAh. Se utiliza el módulo TP4050 (U5) con conector USB-C para cargar la batería. Cuando se la quiera cargar se debe colocar un jumper en el conector JP2. En cambio, para alimentar el circuito el jumper debe ser conectar un jumper en el conector JP1. No se utilizaron los pines OUT+ y OUT- del módulo, ya que los mismos fijan una tensión de 3,3 V que no es suficiente para alimentar al módulo SIM808.

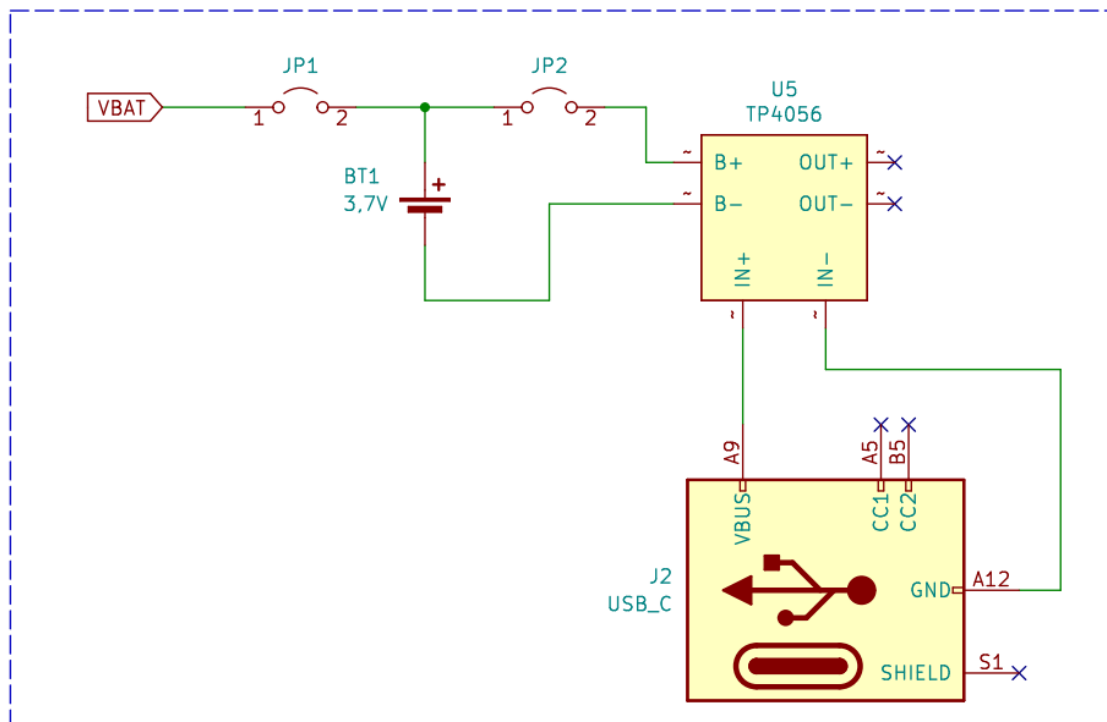


Figura 9 – Alimentación y módulo de carga en el circuito esquemático

### 3.1.3 Circuito

Para montar el circuito, se utilizó una placa experimental de Pertinax de 10 x 10 cm. Se intentó realizar la mayor cantidad de conexiones por debajo para evitar puentes y ser lo más parecido a un posible futuro PCB, el cual no fue realizado hasta el momento por cuestiones de tiempo.

### 3.1.4 Fotos

Se puede observar la vista superior e inferior de la placa finalizada:

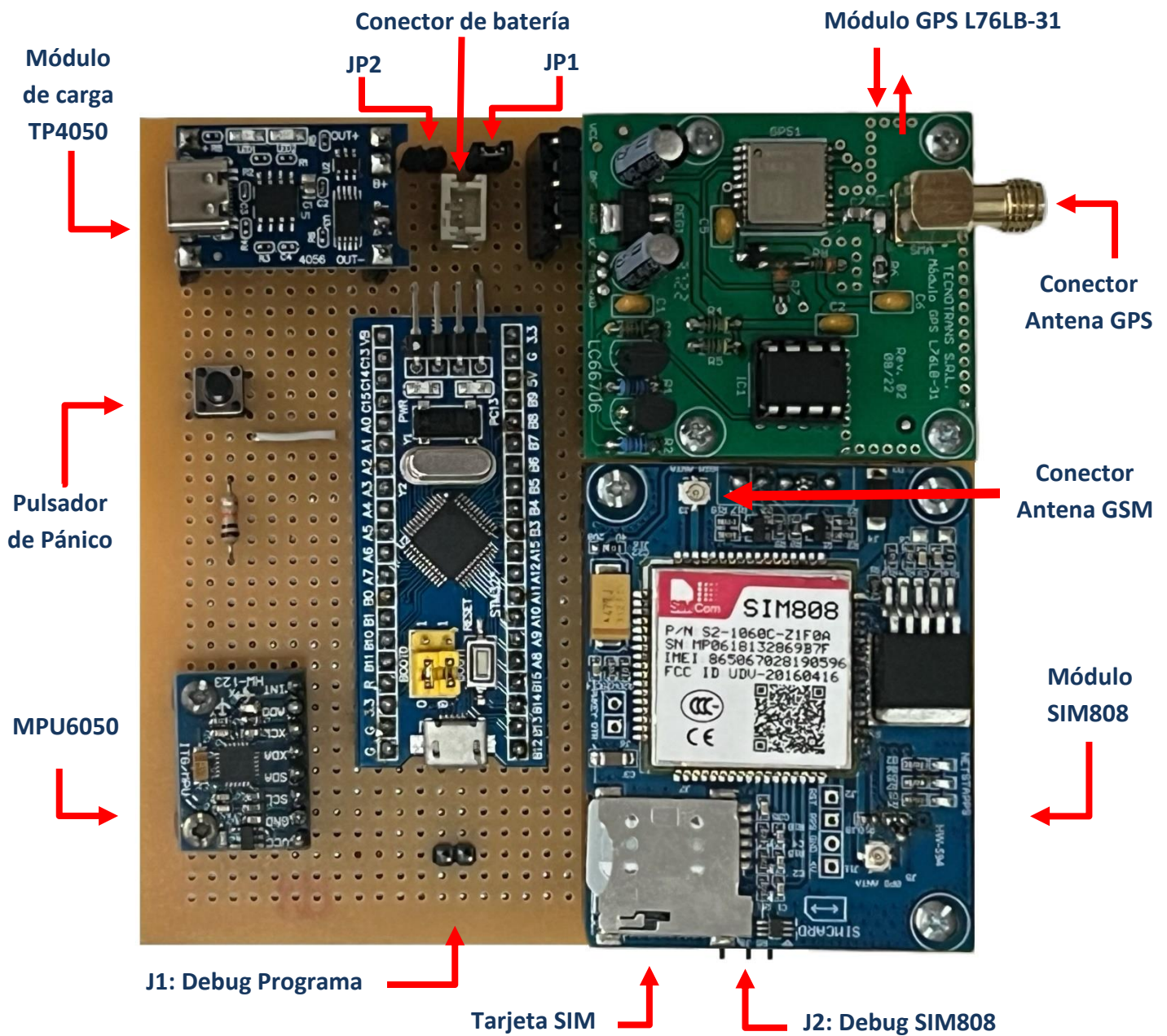


Figura 10 – Vista superior del circuito



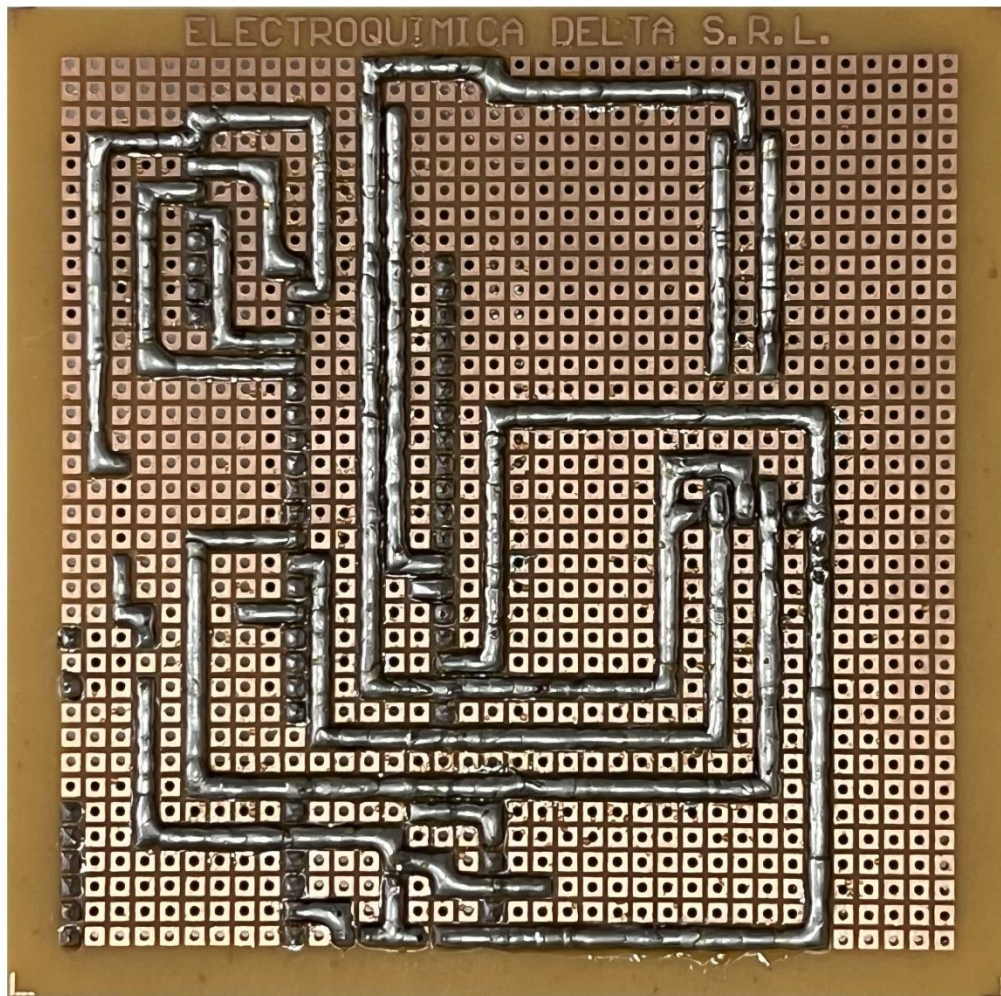


Figura 11 – Vista inferior del circuito

## 3.2 Software

En esta sección, se puede observar cómo fue diagramado el software del dispositivo.

### 3.2.1 Enumeración de rutinas

#### Funciones:

##### GPS.c

- Recepcion\_GPS
- ajustarSegunGMT
- Armar\_Ubi\_Google

**SIM808.c**

- Recepcion\_Modem
- Encolar\_SMS
- Enviar\_SMS
- Cancelar\_SMS
- FS\_Guardar\_Ubicacion

**mpu6050.c**

- mpu6050Init
- mpu6050powerOn
- mpu6050Sampling
- mpu6050GyroScale
- mpu6050AccelScale
- mpu6050Config
- mpu6050GyroRead
- mpu6050AccelRead

**main.c**

- Verificar\_Boton
- send\_uart
- Manejo\_Led

**Rutina principal (Looping)**

- Constantemente:
  - Verificación de botón de pánico y envío de SMS si es pulsado.
- Cada 50 milisegundos:
  - Manejo del led indicador de estado.
  - Verificación del error por time out de los SMS en proceso.
  - Lectura de datos del giróscopo, y envío de SMS si se detecta un movimiento brusco.
- Cada 100 milisegundos:
  - Manejo de comandos de envío de SMS (si es necesario).
- Cada 1 segundo:
  - Debugging por puerto serie.
  - Gestión del temporizador para volver a detectar movimiento brusco si ya se detectó.
- Cada 20 segundos:
  - Se guarda la ubicación en formato de Google Earth en el sistema de archivos del módulo SIM808.

### 3.2.2 Descripción del funcionamiento de cada rutina

#### GPS.c

```
//*****
// Función:      Recepcion_GPS
//*****
```

Esta función se encarga de la recepción de datos del módulo GPS. La misma es llamada dentro de la interrupción por UART1.

El módulo GPS L76LB-31 envía la trama GPRMC (Recommended Minimum Specific GPS/Transit Data) cada un segundo a una velocidad de 4800 Baudios. Se realiza la lógica de recepción para leer la siguiente trama:

**\$GPRMC,hhmmss.ss,a,ddmm.mmmm,n,dddmm.mmmm,w,zz,aa,ddmmaa,dd,v\*CC**

Dato en la trama	Descripción	Interpretación
<b>hhmmss.ss</b>	Hora UTC de la posición	
<i>hh</i>	horas	00 - 24
<i>mm</i>	minutos	00 - 59
<i>ss.ss</i>	segundos	00.000 - 59.99
<b>a</b>	Estado	A = válido
		V = inválido
<b>ddmm.mmmm,n</b>	Latitud	
<i>dd</i>	grados	00 - 90
<i>mm.mmm</i>	minutos	00.000 - 59.999
<i>n</i>	dirección	N = norte, S = sur
<b>dddmm.mmmm,w</b>	Longitud	
<i>ddd</i>	grados	000 - 180
<i>mm.mmm</i>	minutos	00.00 - 59.9999
<i>w</i>	dirección	E = este, W = oeste
<b>z.z</b>	Velocidad media relativa al suelo	0
<b>y.y</b>	Curso (respecto al norte)	0,0 - 359,9
<b>ddmmyy</b>	Fecha UTC de la posición	
<i>dd</i>	día	01 - 31
<i>mm</i>	mes	01 - 12
<i>yy</i>	año	00 - 99
<b>d.d</b>	Variación magnética (grados)	0,0 - 180,0
<b>v</b>	Sentido de variación	E = este
		W = oeste
<b>CC</b>	Checksum	XOR de todos los datos previos

Todos los datos, excepto variación magnética y sentido de variación, al ser recibidos son guardados en una estructura GPS\_aux. Al finalizar el proceso, si el Checksum es válido y el Estado es A (al menos 3 satélites), se da por válido

lo cargado en GPS\_aux y se lo copia en la estructura GPS, que luego es usada en el resto de las funciones del programa.

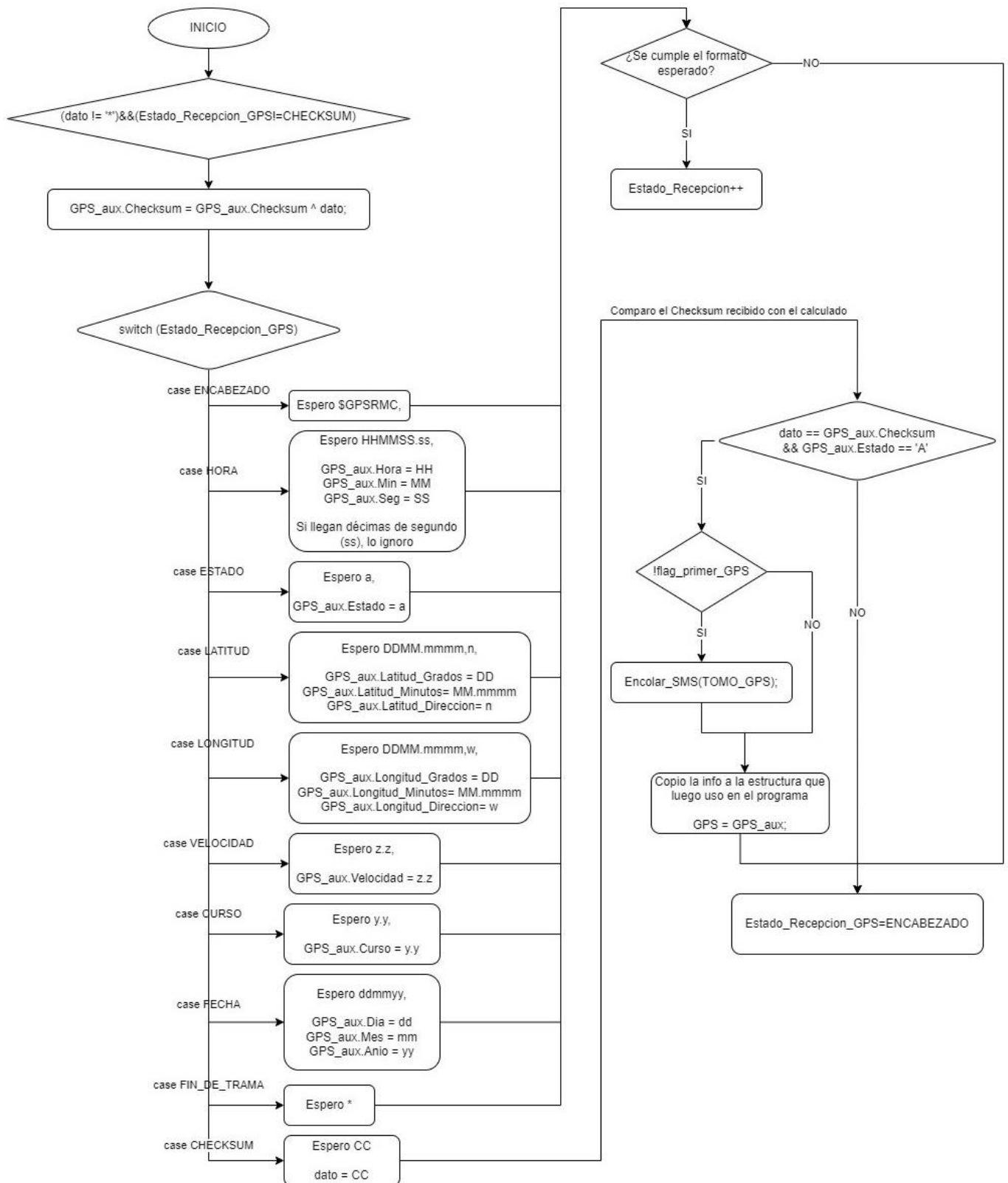


Figura 12 – Diagrama de flujo de la función Recepcion\_GPS

```
//*****  
// Función:      ajustarSegunGMT  
//*****
```

Función para ajustar la fecha y hora recibida del módulo GPS. El GPS envía la fecha y hora en GMT 0. En el programa se fijó la corrección al GMT -3, correspondiente a la zona horaria de Argentina.

```
//*****  
// Función:      Armar_Ubi_Google  
//*****
```

Ubica el punto en el que se encuentra el dispositivo, en Google Maps o Google Earth.

### Google Maps

En el envío de SMS, sin importar la razón por la cual se generó, siempre que se haya validado la posición, se enviará el link para ubicar la posición en Google Maps, junto al texto correspondiente al evento generado.

El link se genera con el siguiente formato:

**<https://www.google.com/maps/@AA.aaaaaa,BB.bbbbbb,20z>**

**AA** corresponde a la parte entera de la Latitud, es decir, a la variable *GPS.Latitud\_Grados*. Se debe colocar con signo negativo en caso de que la variable *GPS.Latitud\_Direccion* sea S (Sur).

**aaaaaa** corresponde a la parte decimal de la Latitud, que se compone realizando:  $(GPS.Latitud\_Minutos/60)*1000000$

**BB** corresponde a la parte entera de la Longitud, es decir, a la variable *GPS.Longitud\_Grados*. Se debe colocar con signo negativo en caso de que la variable *GPS.Longitud\_Direccion* sea W (Oeste).

**bbbbbb** corresponde a la parte decimal de la Longitud, que se compone realizando:  $(GPS.Longitud\_Minutos/60)*1000000$

**20z** es el valor que se decidió fijar para el zoom que se estableció en el mapa. Utilizando valores inferiores a 20 se podría alejar, así como podría acercarse con valores mayores al fijado.

### Google Earth

Cada 20 segundos, se almacena en sistema de archivos del módulo SIM808, la posición en la que se encuentra el dispositivo. Estos datos se podrán posteriormente ingresar en un archivo KML y visualizar en Google Earth el recorrido realizado.

Cada posición se guarda con el siguiente formato:

**[BB.bbbbbb,AA.aaaaaa,0](#)**

(En este caso se ubica primero a la longitud y después a la latitud)



**SIM808.c**

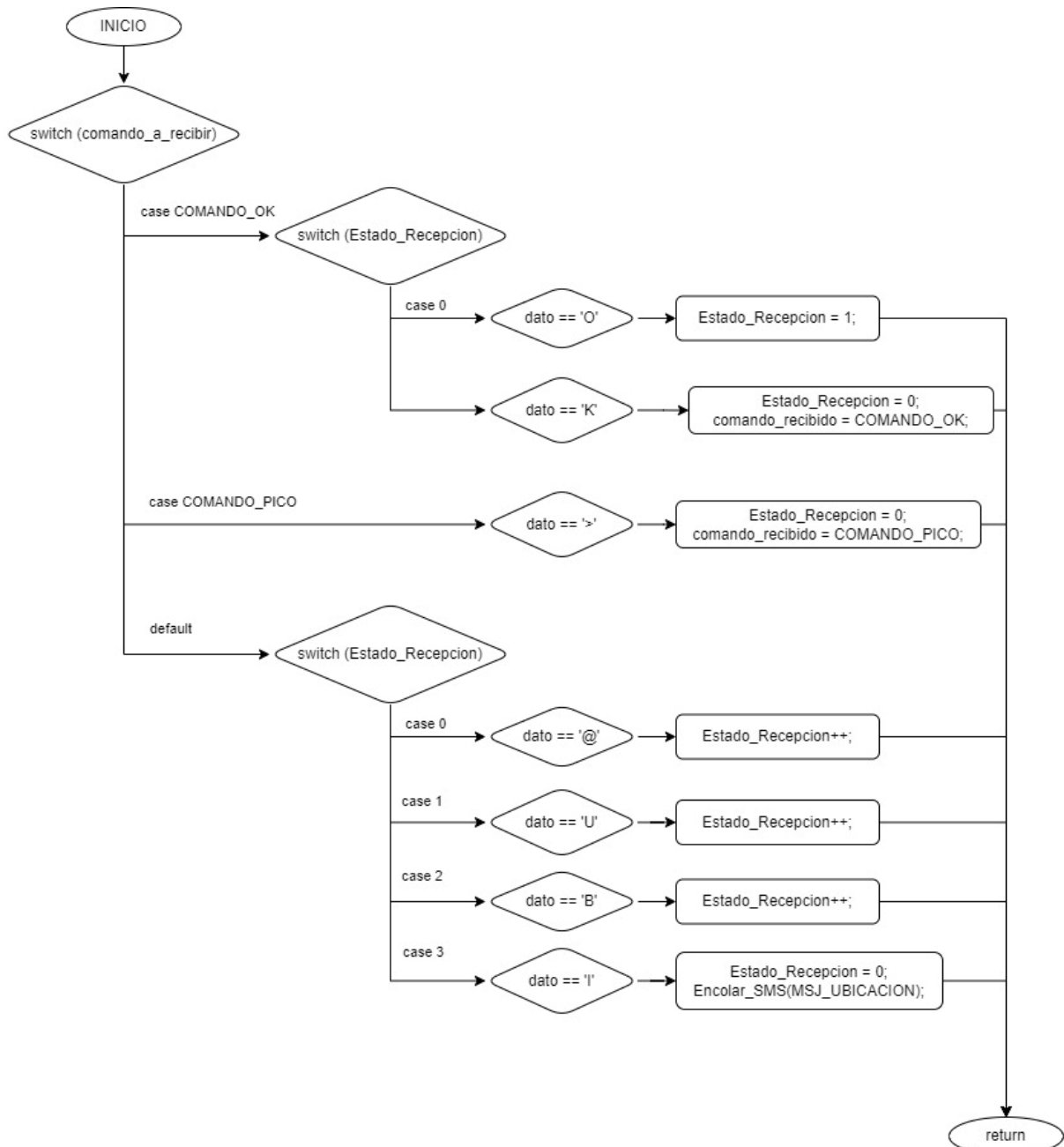
```

//*****
// Función:      Recepcion_Modem
//*****

```

Esta función se encarga de la recepción de datos del módulo SIM808. La misma es llamada dentro de la interrupción por UART2.

El módulo envía caracteres a una velocidad de 9600 Baudios. Se realiza la lógica para recibir las respuestas esperadas a los comandos enviados, como también para recibir el comando @UBI, enviado desde un celular para solicitar la ubicación. De esta forma, se podrían agregar más comandos para diferentes acciones a realizar.



**Figura 13 – Diagrama de flujo de la función Recepcion\_Modem**

```
//*****  
// Función:      Encolar_SMS  
//*****
```

Ingresa en la cola de SMS el próximo mensaje a enviar.

```
//*****  
// Función:      Enviar_SMS  
//*****
```

Envía los comandos al módem para enviar un SMS indicando el evento por el cual se generó junto con el enlace de Google Maps en la última posición válida, con fecha y hora.

La secuencia de comandos es la siguiente, indicando en color rojo los comandos enviados por el microcontrolador, y en verde las respuestas esperadas por parte del módulo.

*SE HABILITAN LAS FUNCIONES (1: Full functionality)*

**AT+CFUN=1**

OK

*SE HABILITA LA LECTURA DE SMS RÁPIDO*

**AT+CNMI=2,2,0,0,0**

OK

*SE HABILITA EL MODO TEXTO*

**AT+CMGF=1**

OK

*SE ADJUNTA A LA RED GPRS*

**AT+CGATT=1**

OK

*SE SETEA EL APN DE CLARO*

**AT+CSTT="igprs.claro.com.ar"**

OK

*SE FIJA EL CENTRO DE SERVICIOS DE CLARO*

**AT+CSCA="+543200000001"**

OK

*SE INDICA A QUE NÚMERO SE DESEA ENVIAR EL MENSAJE*

**AT+CMGS="+XXXXXXXXXXXX"**

> **Texto a enviar**

*(Para finalizar se envía el número 26, equivalente al valor ASCII del CTRL+Z)*

OK

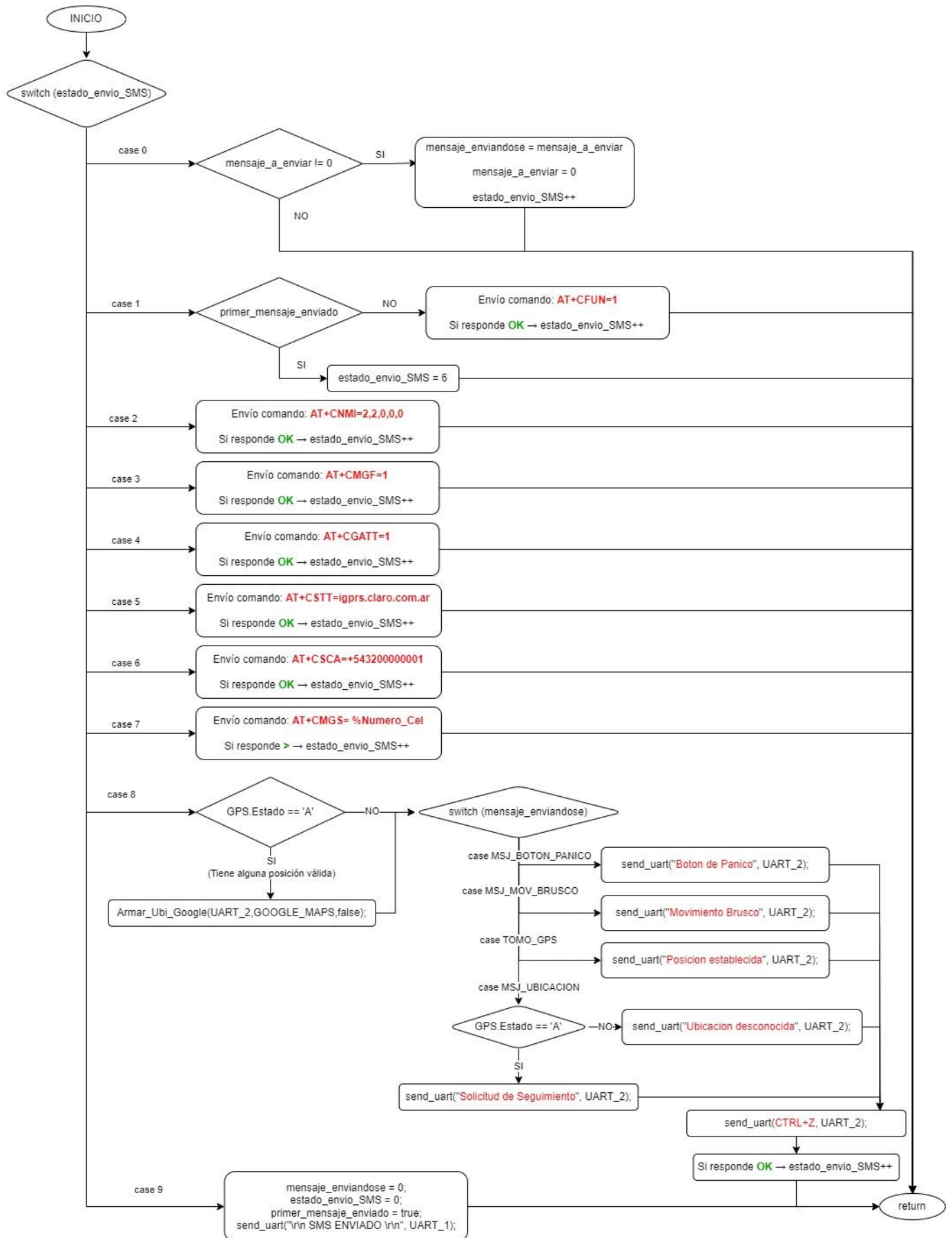


Figura 14 – Diagrama de flujo de la función Enviar\_SMS

Cada envío de comando se trabaja de la siguiente manera:

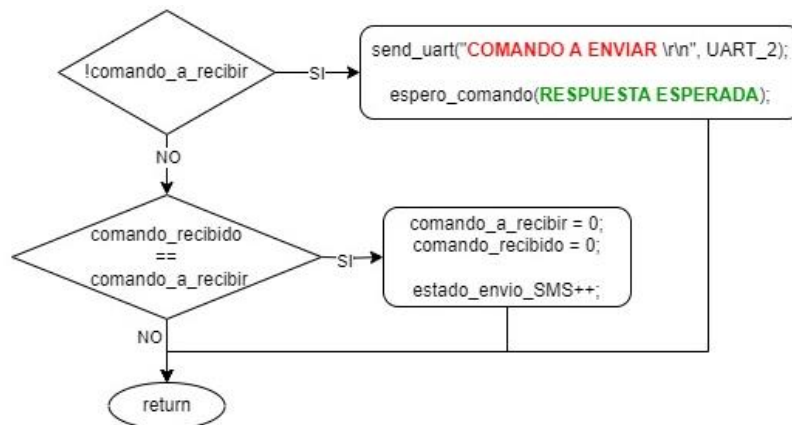


Figura 14 – Diagrama de flujo del envío de comandos al módulo SIM808

```

//*****
// Función:      Cancelar_SMS
//*****

```

Cancela el envío de SMS por Time Out y lo informa por puerto serial.

```

//*****
// Función:      FS_Guardar_Ubicacion
//*****

```

Guarda la Ubicación GPS con formato de Google Earth en el archivo EARTH.TXT dentro del File System del SIM808.

La secuencia de comandos es la siguiente:

```

COMANDO PARA ESCRIBIR EL ARCHIVO
AT+FSWRITE=
<nombre del archivo: EARTH.TXT>,
<modo: 1 (escribe al final del archivo)>,
<tamaño: 30 caracteres>,
<tiempo de espera 1000 segundos>
AT+FSWRITE=EARTH.TXT,1,30,1000
> Texto a guardar

```

(Se envía la ubicación en formato de Google Earth)

OK

Se le da un valor alto de tiempo de espera, porque siempre se envían más caracteres que lo indicado, relleno con espacios.

**mpu6050.c**

```
//*****
// Función:      mpu6050Init
//*****
```

Inicializa el módulo MPU6050, leyendo su identificación para verificar la conexión.

```
//*****
// Función:      mpu6050powerOn
//*****
```

Enciende el MPU6050 configurando el registro de administración de energía.

```
//*****
// Función:      mpu6050Sampling
//*****
```

Configura la tasa de muestreo del MPU6050 para la lectura de datos.

```
//*****
// Función:      mpu6050GyroScale
//*****
```

Configura la escala del giroscopio del MPU6050 según la escala especificada.

```
//*****
// Función:      mpu6050AccelScale
//*****
```

Configura la escala del acelerómetro del MPU6050 según la escala especificada.

```
//*****
// Función:      mpu6050Config
//*****
```

Configura el MPU6050 con parámetros predefinidos si la identificación del dispositivo es la esperada (mpu6050)

```
//*****
// Función:      mpu6050GyroRead
//*****
```

Lee y calcula las velocidades angulares del giroscopio.

Esta es la función más importante, ya que, en el bucle principal, se verifican cada 50 ms los datos obtenidos del giróscopo, comparándolos con un valor fijado empíricamente. Si se sobrepasa este valor, hacia cualquiera de los ejes, se entenderá que ha ocurrido un movimiento brusco.

```
//*****
// Función:      mpu6050AccelRead
//*****
```

Lee y calcula los ejes del acelerómetro.

```
//*****
// Función:      Verificar_Boton
//*****
```

Esta función verifica que el botón se haya pulsado correctamente (lógica anti-rebote). Se sabe que el pulsador tiene un resistor de pull-up. Al ser presionado, se detecta un estado bajo. Una vez que se detecta el estado bajo, para validarlo, se esperan unos 20 segundos (falling time) y se vuelve a chequear que siga en estado bajo, de lo contrario, se tomará como invalido. En caso de estarlo, se esperarán otros 20 segundos (rising time) y se chequeará que vuelva a estar en estado alto. Una vez terminado ese proceso, se considera como válida el pulsado.

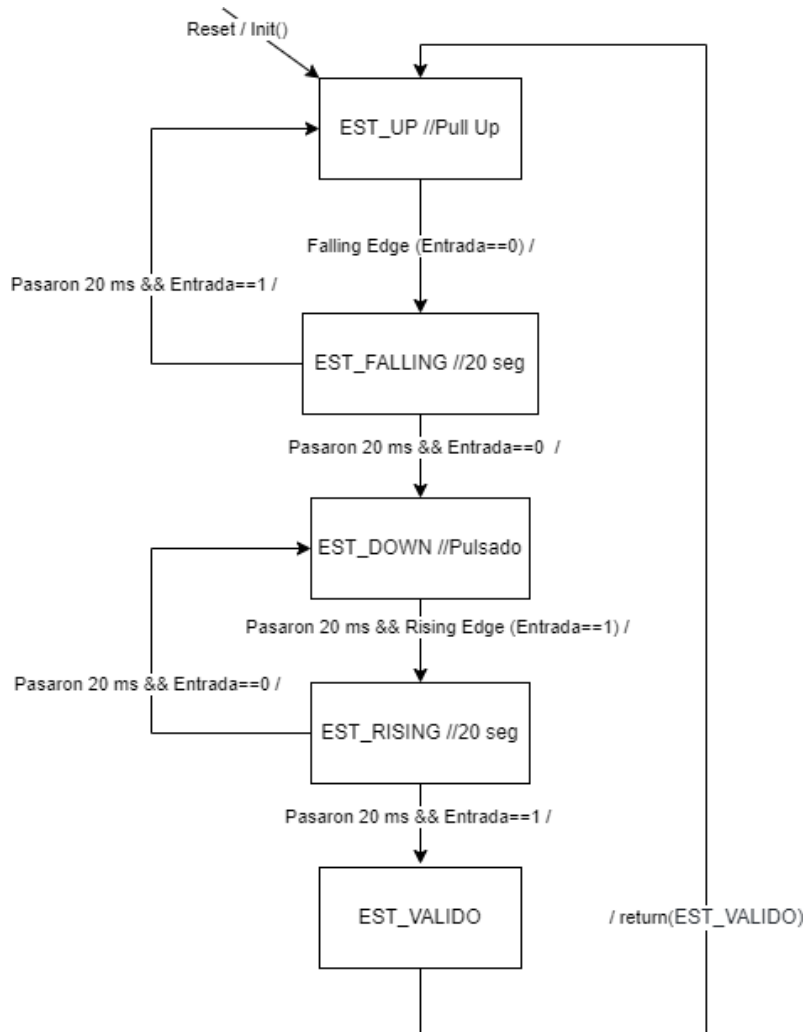


Figura 15 – Diagrama de flujo de la función Verificar\_Boton

```
//*****
// Función:      send_uart
//*****
```

Envía una cadena de caracteres por la UART seleccionada.

- Si se selecciona UART\_1, será para el puerto serie destinado al debugging.
- Si se selecciona UART\_2, será para enviarle comandos al módulo SIM808.

```
//*****
// Función:      Manejo_Led
//*****
```

Se modifica la velocidad en la que titila el led según el estado del dispositivo:

MUY RAPIDO (c/ 100 ms) cuando está enviando un SMS.

RAPIDO (c/ 300 ms) cuando está buscando señal GPS.

LENTO (c/ 1 s) cuando tiene señal GPS y está en funcionamiento normal.

```
//*****
// Rutina principal
//*****
```

Los timers son seteados para levantar un flag cada 50 ms y otro cada 1 segundo. Con estos dos flags se maneja todo el programa dentro del siguiente bucle principal:

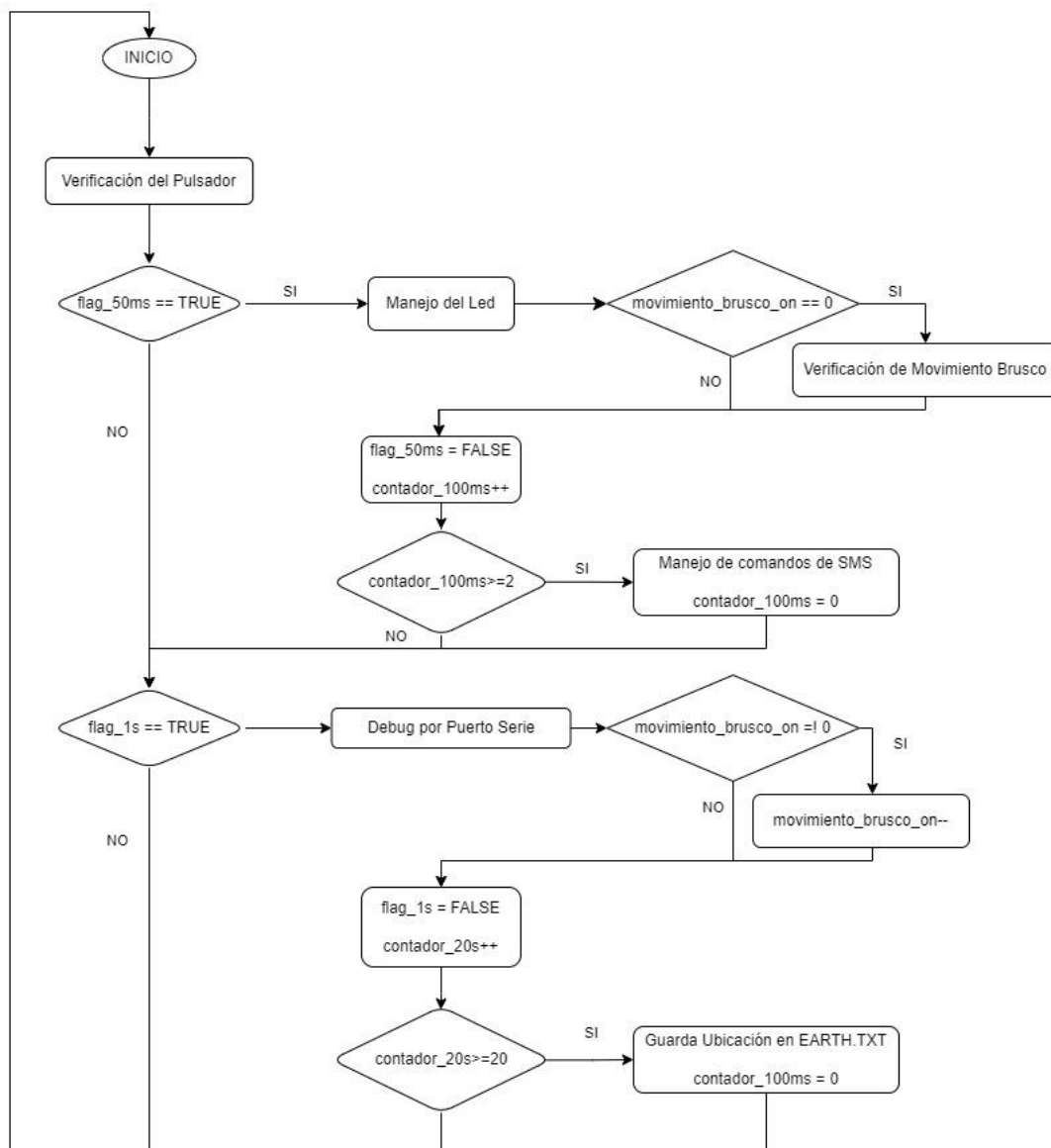


Figura 16 – Diagrama de flujo de la rutina principal

## 4 Modo de operación

### 4.1 Uso del dispositivo

Para encender el dispositivo debe colocar un jumper en el conector JP1 y asegurarse que la batería esté conectada. Vale aclarar que para su uso deben estar debidamente conectadas las antenas de GPS y de GPRS, como también se debe insertar una tarjeta SIM. Este localizador está destinado principalmente al uso de adultos mayores que salen a caminar o hacer las compras. Al salir, se debe encender el dispositivo y colocarlo con buena fijación en un bolso de cintura (riñonera). Esta elección se justifica por su proximidad al centro de masa del cuerpo, favoreciendo una detección de caídas más efectiva al capturar movimientos bruscos en esta área. Además, esta ubicación proporciona comodidad, estabilidad y seguridad, minimizando la interferencia con las actividades cotidianas y promoviendo una experiencia de uso cómoda para el usuario.

### 4.2 Envío de Eventos por SMS

Al utilizar el dispositivo, se pueden generar cuatro tipos de eventos que enviarán distintos mensajes al número predefinido:

1) Primera detección de ubicación

Al encenderlo, el GPS tardará un tiempo hasta obtener buena señal. Una vez que obtenga una ubicación válida enviará un SMS indicando el posicionamiento.

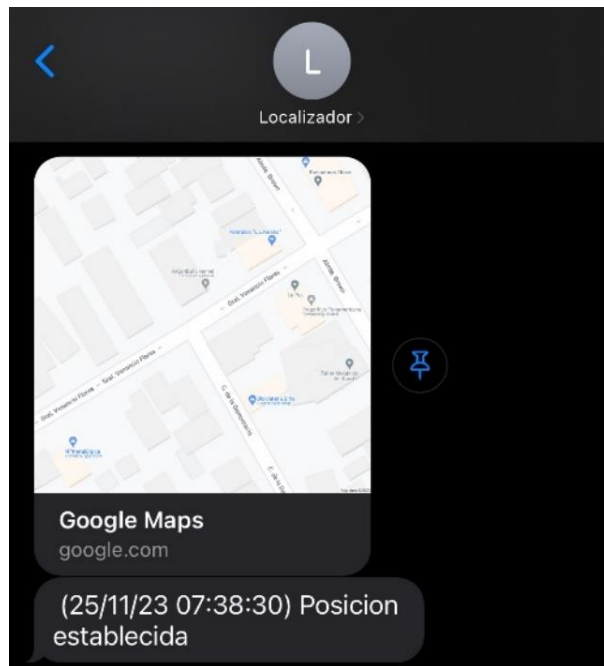


Figura 17 – Envío de SMS de posición establecida



## 2) Botón de Pánico

El usuario que porte el localizador tendrá un botón al alcance, que podrá presionar ante cualquier caso de urgencia, y será avisado por SMS.

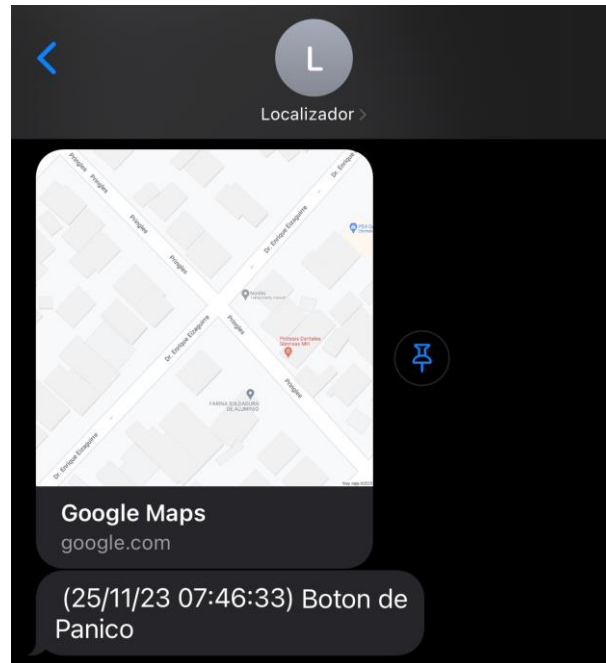


Figura 18 – Envío de SMS de botón de pánico

## 3) Movimiento Brusco

Al detectarse un movimiento brusco se enviará un SMS informándolo, ya que puede deberse a una caída.

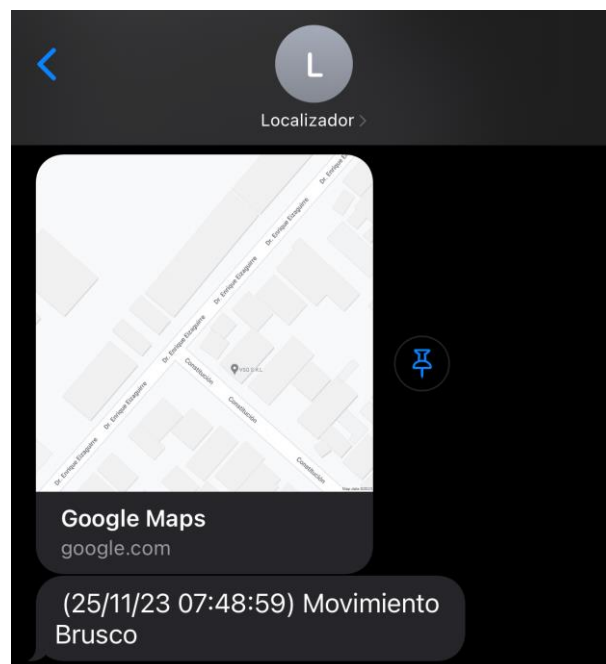


Figura 19 – Envío de SMS de movimiento brusco

#### 4) Solicitud de Seguimiento

Al tener el número telefónico que pertenece al localizador, se le puede enviar un SMS con el texto **@UBI** y ante este comando el dispositivo responderá con la última posición válida.

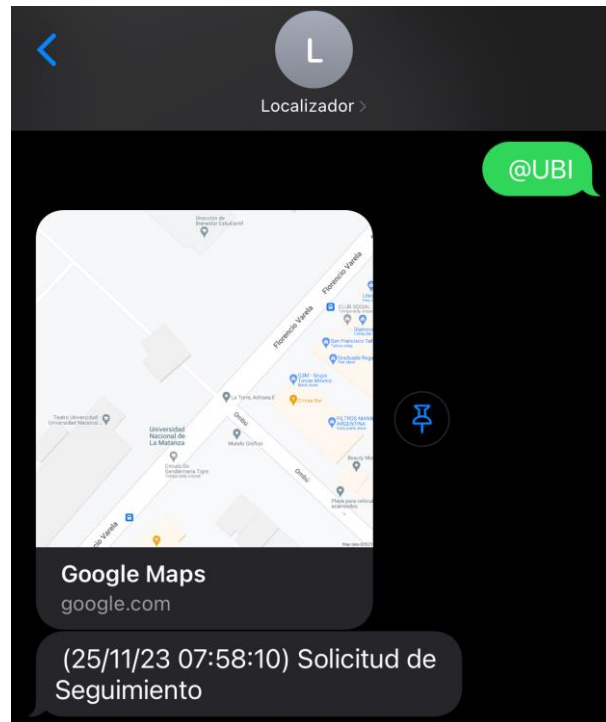


Figura 20 – Envío de SMS de ubicación solicitada por comando

### 4.3 Visualización de recorrido realizado

El dispositivo guarda internamente el recorrido realizado, generando puntos de posicionamiento cada 20 segundos. Para poder visualizarlo se debe hacer lo siguiente:

- 1) Encender el dispositivo, con la batería colocada y el jumper en JP1.

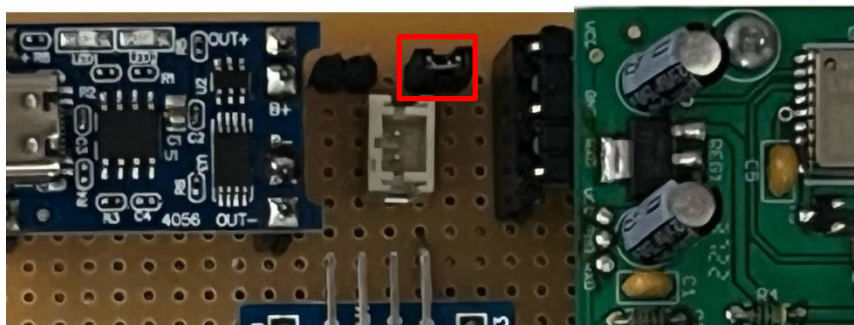


Figura 21 – Ubicación de jumper JP1

- 2) Conectarse a los pines J2 mediante un conversor USB a UART TTL.

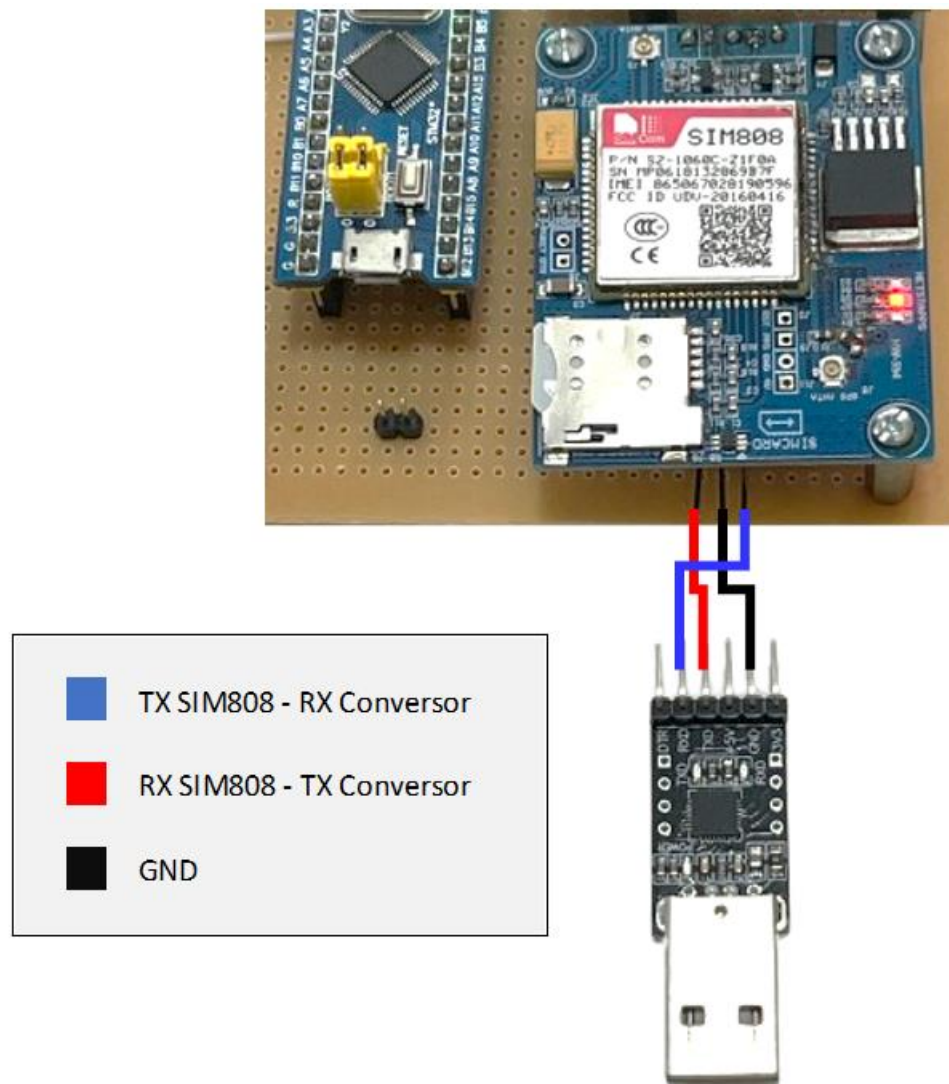


Figura 22 – Conexión para comunicarse con el módulo SIM808

- 3) Abrir una terminal con la siguiente configuración de Puerto Serial:

Speed:	9600	▼
Data:	8 bit	▼
Parity:	none	▼
Stop bits:	1 bit	▼

Figura 23 – Configuración de la terminal para comunicarse con el módulo SIM808

- 4) Enviarle el siguiente comando:

**AT+FSREAD=EARTH.TXT,0,10000,0**

El modem responderá con los puntos de posicionamiento, por ejemplo:

```
-58.5642856, -34.6702692,0
-58.5644672, -34.6701326,0
-58.5644886, -34.6700361,0
-58.5644886, -34.6700361,0
-58.5644783, -34.6700859,0
-58.5644783, -34.6700859,0
-58.5645565, -34.6701137,0
-58.5645786, -34.6701215,0
-58.5645786, -34.6701215,0
-58.5645781, -34.6701479,0
```

- 5) Copiar la respuesta del modem y pegarla en el archivo *SEGUIMIENTO.kml*, guardado en la carpeta del proyecto. Guardar el archivo modificado.



```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Recorrido Localizador</name>
    <Style id="line-000000-1200-nodesc-normal">
      <LineStyle>
        <color>ff0000ff</color>
        <width>4.2</width>
      </LineStyle>
      <BalloonStyle>
        <text><![CDATA[<h3>${name}</h3>]]></text>
      </BalloonStyle>
    </Style>
    <Style id="line-000000-1200-nodesc-highlight">
      <LineStyle>
        <color>ff0000ff</color>
        <width>4.2</width>
      </LineStyle>
      <BalloonStyle>
        <text><![CDATA[<h3>${name}</h3>]]></text>
      </BalloonStyle>
    </Style>
    <StyleMap id="line-000000-1200-nodesc">
      <Pair>
        <key>normal</key>
        <styleUrl>#line-000000-1200-nodesc-normal</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#line-000000-1200-nodesc-highlight</styleUrl>
      </Pair>
    </StyleMap>
    <Placemark>
      <name>Linea 2</name>
      <styleUrl>#line-000000-1200-nodesc</styleUrl>
      <LineString>
        <tessellate>1</tessellate>
        <coordinates>
          //PEGAR PUNTOS DE SEGUIMIENTO AQUI
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Figura 24 – Archivo SEGUIMIENTO.kml

- 6) Ingresar a <https://earth.google.com/>
- Seleccionar +Nuevo e importar el archivo KML guardado.

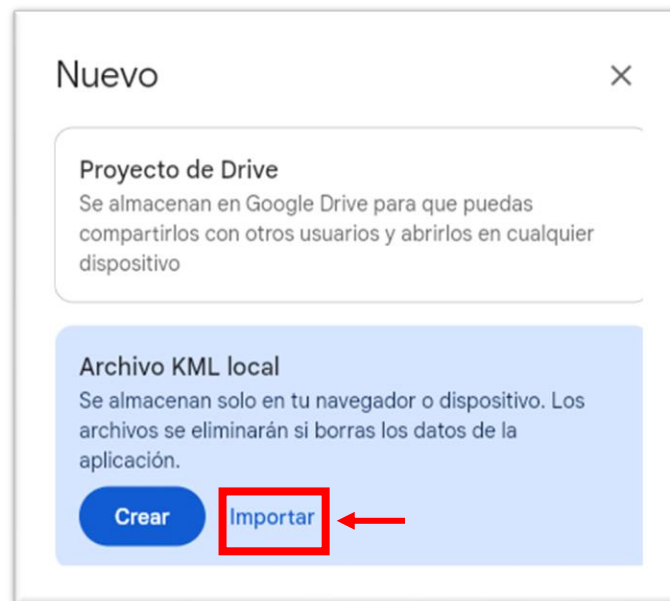


Figura 25 – Importar archivo en Google Earth

- 7) Ya puede visualizar el seguimiento del recorrido:

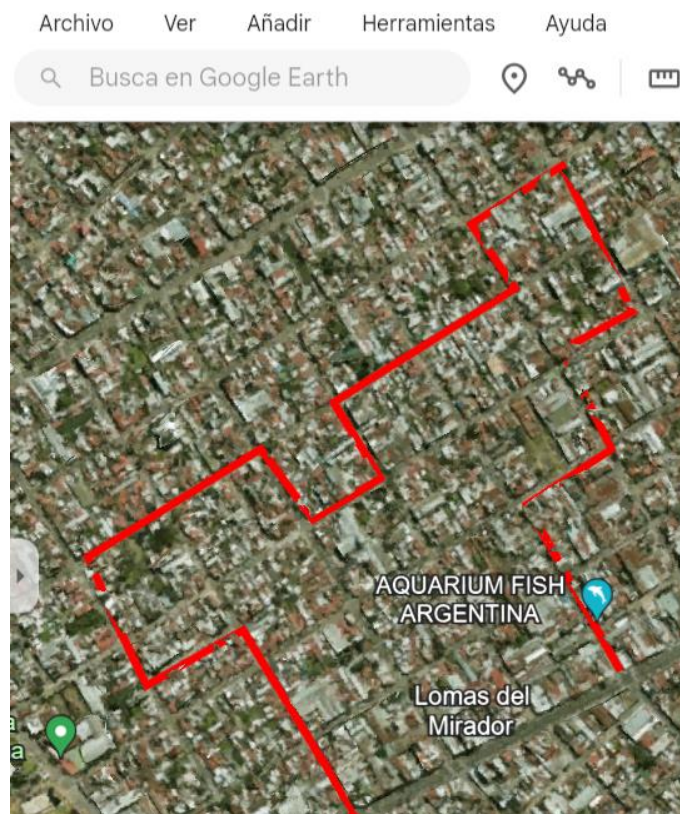


Figura 26 – Visualización de recorrido en Google Earth



- 8) En caso de querer borrar el recorrido almacenado puede enviar los siguientes comandos:

**AT+FSDEL=EARTH.TXT**

**OK**

**AT+FSCREATE=EARTH.TXT**

**OK**

## 4.4 Debug del programa

Si se quiere hacer el debug del programa, se debe usar el conector J1 y comunicarse a la PC mediante un conversor USB a UART TTL, conectado como se ve en la siguiente imagen.

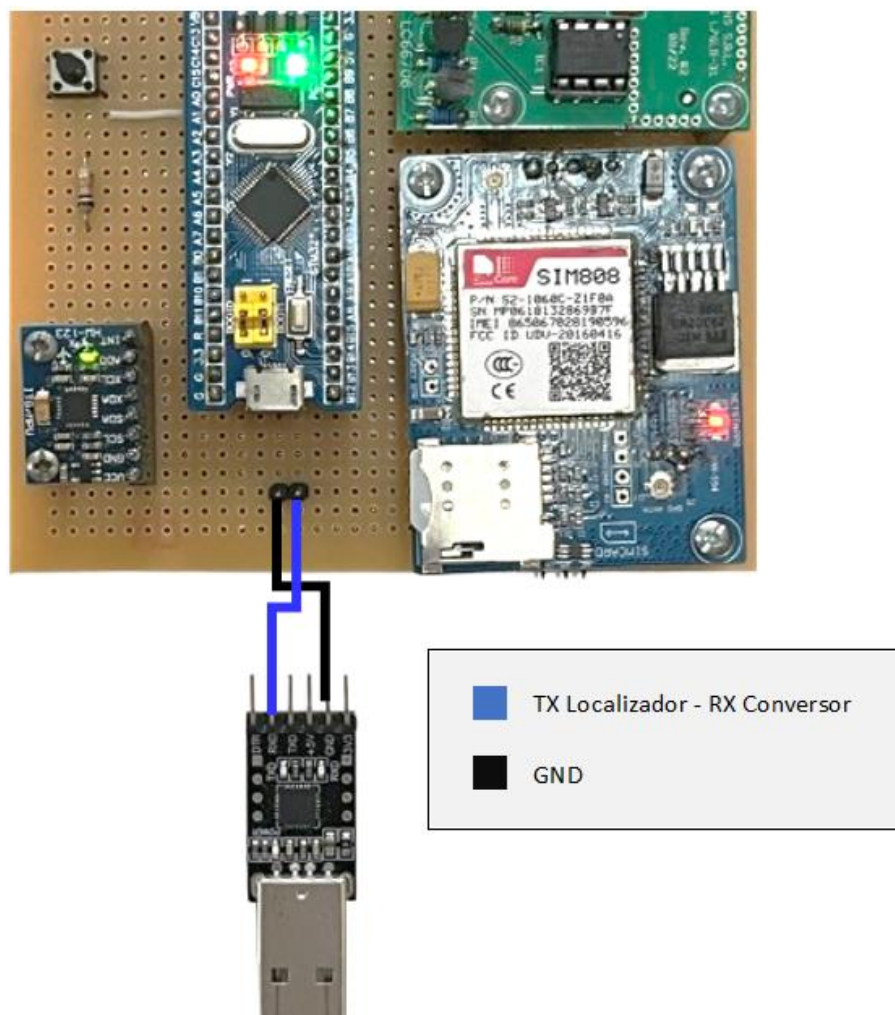


Figura 27 – Conexión para ver el debug del dispositivo

Luego debe abrir una terminal con la siguiente configuración de Puerto Serial:

Speed:	4800	▼
Data:	8 bit	▼
Parity:	none	▼
Stop bits:	1 bit	▼

Figura 28 – Configuración de la terminal para ver el debug del dispositivo

## 4.5 Carga de batería

Para cargar la batería, se debe quitar el jumper del conector JP1 y colocarlo en JP2.

Luego, se debe conectar mediante un cable USB – USB-C, el conector existente en el módulo U5, y por el otro extremo del cable a una fuente de alimentación compatible, como un cargador de pared, una computadora o una batería externa.

Mientras se realiza la carga, el módulo enciende un LED rojo, y cuando la batería llega a plena carga se enciende un LED azul.

Una vez que la batería esté completamente cargada, puede desconectar el cargador y volver a colocar el jumper en JP1 en caso de querer utilizar el dispositivo.

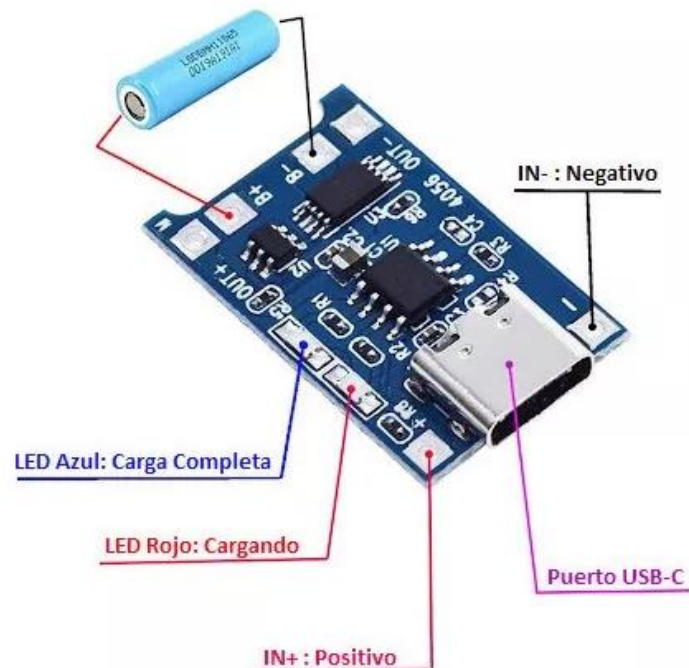


Figura 29 – Módulo de carga TP4056

## 5 Ensayos

### 5.1 Acelerómetro / Giróscopo

Una vez realizado el código para poder interpretar los valores proporcionados por el módulo MPU-6050, se decidió enviar continuamente por puerto serie los valores leídos, para observarlos en la PC. Se procedió a realizar movimientos de la placa, simulando movimientos bruscos que puedan asimilarse al movimiento que implica una caída. De esta forma empírica, se estableció el valor mínimo que se debe recibir en los ejes del giróscopo para interpretar un movimiento brusco.

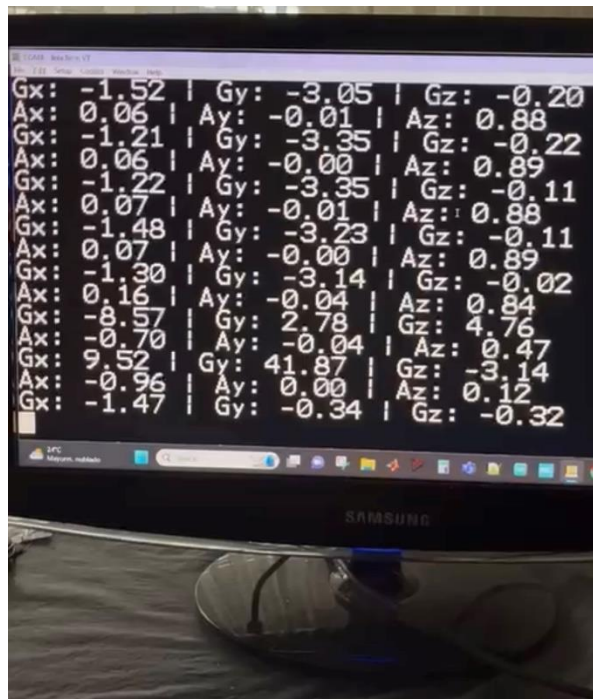


Figura 30 – Debug de MPU-6050 en terminal de PC

### 5.2 Datos de localización

Para probar el envío de los mensajes de texto, se fueron generando los distintos eventos para validar el funcionamiento de todos ellos. Estas pruebas podían realizarse de forma estática. Luego, una vez que el dispositivo contaba con la alimentación por batería, para probar el guardado de la ubicación en el sistema de archivos del módulo SIM808, se realizó un viaje a la UNLaM y, al llegar, se descargó la información. El resultado fue positivo, ya que el viaje se registró completamente. Esta prueba fue realizada en un automóvil, por ende, si usáramos el dispositivo para caminatas, la precisión sería aún mejor, debido a que al reducir la velocidad los distintos puntos que forman el recorrido estarían más cercanos entre sí.

El archivo generado se encuentra con el nombre *Viaje\_a\_UNLAM.kml* en la carpeta del proyecto.



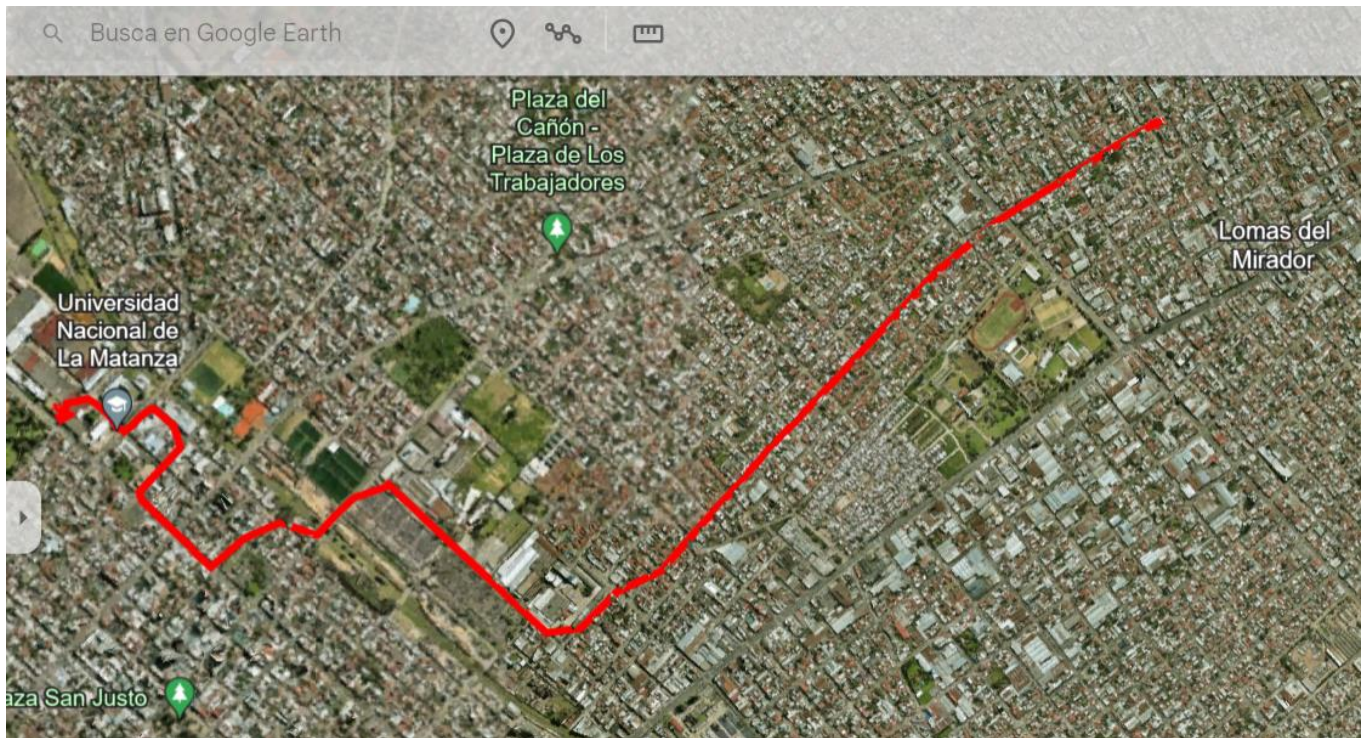


Figura 31 – Recorrido registrado en el viaje a la UNLaM

### 5.3 Consumo del dispositivo

Se midió el consumo de cada módulo por separado, así como también el consumo del dispositivo en su conjunto. Los valores registrados en funcionamiento normal son los siguientes:

CONSUMO NORMAL	
Blue Pill	15 mA
MPU6050	6 mA
GPS	55 mA
SIM808	240 mA
<b>TOTAL</b>	<b>316 mA</b>

Puede notarse que el módulo que más corriente necesita es claramente el SIM808. Así mismo, este módulo tiene picos momentáneos de hasta 2 A cuando realiza una ráfaga de transmisión (*transmitting burst*), generando en esos momentos los picos máximos de consumo del localizador.

## 6 Conclusiones

Se llegó al desarrollar el prototipo de un dispositivo de localización y seguimiento que, en lo personal, satisface las expectativas planteadas a la hora de su diagramación inicial. Sin embargo, quedaron varios aspectos que pueden ser optimizados en futuros desarrollos.

En principio, el tamaño del dispositivo es un tanto incómodo para su uso en seguimiento de personas. Esto se debe principalmente al uso del módulo de GPS externo, en lugar de utilizar el GPS interno del módulo SIM808. Luego de reiteradas pruebas, siguiendo los pasos indicados en las notas de aplicación de SimCom, no se pudo detectar ubicación GPS a pesar de enviar los comandos tal cual como recomienda el fabricante. El correcto uso de esta funcionalidad hubiera sido fundamental para realizar un dispositivo más compacto.

Por otro lado, un punto a favor fue la decisión de no utilizar una tarjeta microSD como se indicaba en el preinforme, sino que en su lugar se utilizó el sistema de archivos del módulo SIM808. Esto ayudó a que el equipo no sea aún más grande, y además reduce el costo del dispositivo final.

Otro objetivo que se planteó en el preinforme y no se pudo realizar por cuestiones de tiempo es el diseño y fabricación de un PCB en KiCAD. Sin embargo, se llegó a un prolijo circuito que podría implementarse a futuro en un PCB sin mucha complicación. Además, por ser un prototipo, el pulsador de pánico quedó posicionado sobre la placa, cuando en un dispositivo real debería ser externo y de mayor tamaño, sobre todo por ser requerido en momentos de urgencia.

Se logró alimentar al dispositivo con batería, sin deber tenerlo enchufado todo el tiempo, lo cual era un paso fundamental para poder realizar pruebas concretas de funcionamiento. Además, para facilitar la carga de la batería, se adicionó el módulo TP4050 en la propia placa. Este módulo no pudo ser aprovechado del todo, ya que su salida de 3,3 V no era suficiente para proveer la tensión mínima requerida por el módulo SIM808 (3,4 V), entonces se debieron colocar los jumpers JP1 y JP2 para independizar el proceso de carga respecto al funcionamiento normal del dispositivo, donde queda alimentado directamente por la batería de 3,7 V.

En cuanto el software, el prototipo cumple gratamente con lo planteado al momento de diagramarlo. Detecta tanto los movimientos bruscos como el accionar del pulsador de pánico y lo informa por SMS, así como también informa la posición en caso de ser solicitada por un comando a través de un SMS. Guarda con precisión los recorridos realizados y permite su posterior visualización en el mapa.

## 7 Proyecto finalizado

Se presenta en la siguiente imagen el localizador finalizado con su batería y sus antenas colocadas.

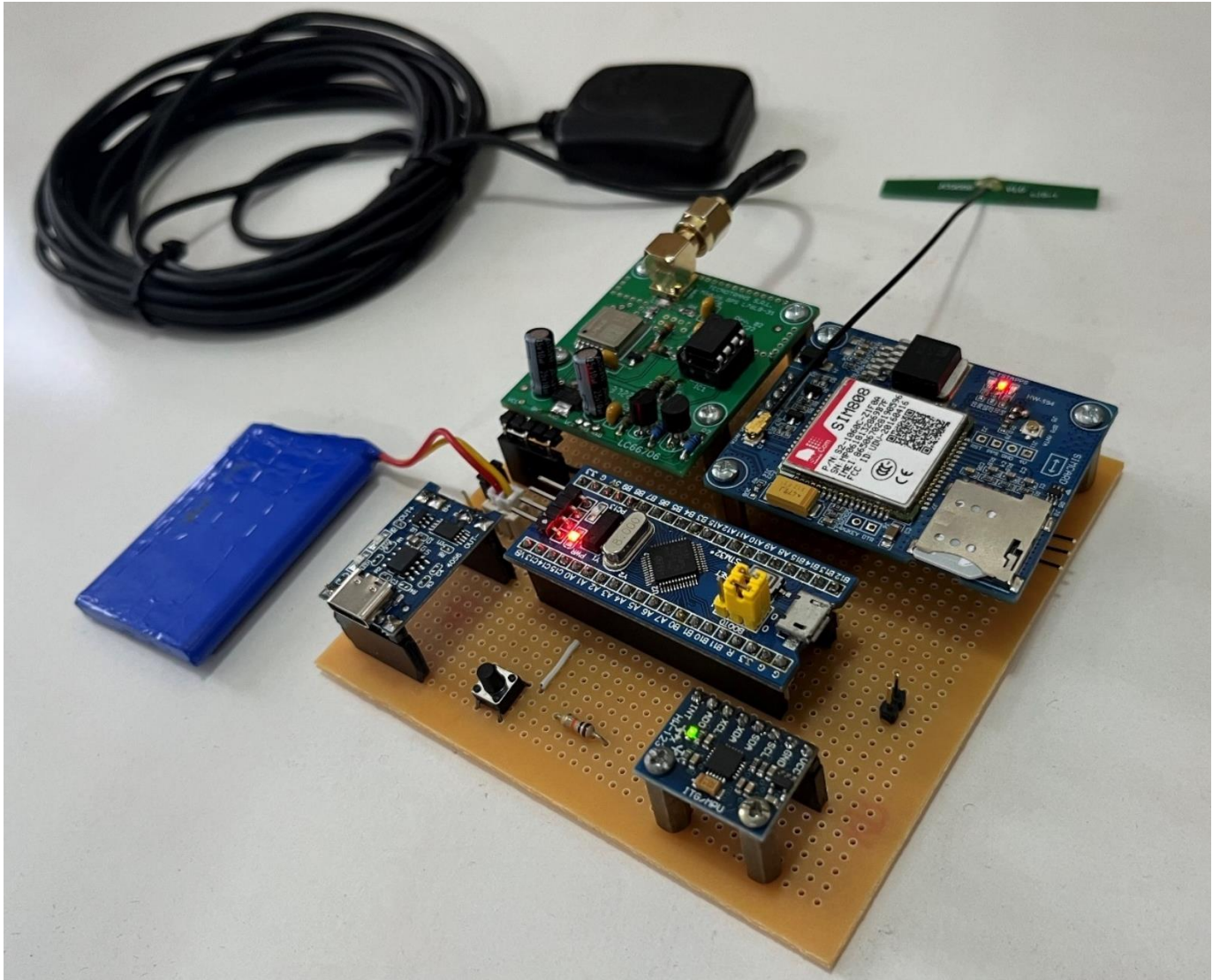


Figura 32 – Proyecto finalizado



## 8 Referencias

1. GPS: The Global Positioning System – <https://www.gps.gov/>.
2. Global Positioning System: The Mathematics of GPS Receivers – Autor: Richard B. Thompson – Mathematics Magazine.
3. RMC - Recommended Minimum Specific GPS/Transit Data – <https://www.ae.utexas.edu/courses/ase389p7/projects/svatek/commands/RMC.html>.
4. Global System for Mobile Communication (GSM) – The International Engineering Consortium.
5. Unidad de medición inercial: estudio de fiabilidad y precisión – Autor: Pablo Martínez Galindo – Universidad de Zaragoza.
6. UM1850 – Description of STM32F1 HAL and low-layer drivers – STMicroelectronics.
7. SIM808\_Hardware\_Design\_V1.00 – SIMCom.
8. SIM800 Series AT Command Manual V1.09 – SIMCom.
9. SIM800 Series\_FS\_Application Note V1.03 – SIMCom.