MANUAL - SENSOR VIPV

David Rodríguez Blanco

Instituto Energía Solar – Febrero/2023

INDICE

[SOFWARE NECESARIO 2](#_Toc132270692)

[STM32 CUBE IDE 2](#_Toc132270693)

[TERA TERM 3](#_Toc132270694)

[DESCARGA Y PUESTA EN MARCHA DEL PROGRAMA 6](#_Toc132270695)

[MODIFICACIONES NECESARIAS EN EL CÓDIGO y THINGSPEAK 7](#_Toc132270696)

[THINGSPEAK 7](#_Toc132270697)

[CÓDIGO 9](#_Toc132270698)

[CONEXIÓN WIFI 9](#_Toc132270699)

[CONEXIÓN THINSPEAK 9](#_Toc132270700)

[HARDWARE 10](#_Toc132270701)

[SOTWARE UPDATE 11](#_Toc132270702)

[MODIFICACIONES EN EL CÓDIGO 11](#_Toc132270703)

[RECUPERACIÓN DEL CÓDIGO INICIAL 12](#_Toc132270704)

[¿QUÉ PASA SI SE PIERDE LA CONEXIÓN? 12](#_Toc132270705)

# SOFWARE NECESARIO

Con el fin de poder modificar, ejecutar y visualizar en tiempo real dicha ejecución, se requiere la instalación de dos programas

## STM32 CUBE IDE

Gracias a este se pueden importar y modificar el código en función de nuestras necesidades, realizar actualizaciones o incorporar nuevas funciones.

A continuación se detallan los pasos necesarios para su instalación en el caso de ser necesario.

1. Dirigirse al siguiente enlace: <https://www.st.com/en/development-tools/stm32cubeide.html>
2. Imagen que contiene Logotipo

   Descripción generada automáticamentePinchar en “Get Software”.
3. Seleccionar “Get latest” en la correspondiente versión, en función de las características del equipo utilizado.

Tabla

Descripción generada automáticamente

1. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteAceptar los términos e iniciar sesión con nuestra cuenta o nombre y correo electrónico en su defecto.

1. Una vez se rellenan todos los campos pinchar en “download”.
2. Dirigirse al correo, donde se recibirá la confirmación de registro y el enlace correspondiente para su descarga.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Finalmente, seguir los pasos del instalador.

## TERA TERM

Este software proporciona la terminal necesaria para visualizar en tiempo real la ejecución del programa. Resulta especialmente útil para depurar el código, detectar errores, corregirlos y verificar el correcto funcionamiento del programa.

Al igual que en el anterior, se detallan a continuación los pasos necesarios para su instalación y configuración:

1. Dirigirse al siguiente enlace: <https://tera-term.softonic.com/>
2. Pinchar en “descargar gratis”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

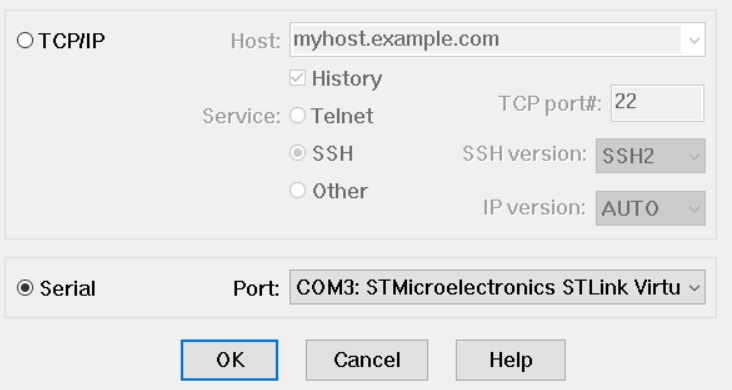
1. Cerrar publicidad y pinchar en “descargar gratis para PC”

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

1. Ejecutar el instalador (.exe) que se ha descargado.

Para la correcta visualización del programa a través de la terminal es necesario modificar ciertos parámetros dentro del programa.

1. Conectar mediante un cable USB la placa al PC. El terminal correspondiente a la placa debe conectarse en el puerto identificado como “USB STLINK”.
2. Una vez está conectado abrir el Tera Term. Debe de aparecer una ventana como la mostrada a continuación:
3. Seleccionar la opción “serial”, indicando además el puerto correspondiente a la placa en el caso de tener varios disponibles. Una vez seleccionado marcar “OK”.
4. Una vez dentro la terminal, en la barra superior, dirigirse a “Setup” -> “Terminal”, estableciendo el “New-line, Receive” como “AUTO”. Pinchar en “OK” para salir.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Nuevamente, desde la opción “Setup” ir a “Serial port”. Aquí, establecer los valores que se observan en la siguiente imagen:

Interfaz de usuario gráfica

Descripción generada automáticamente

Seleccionar “New setting” para aceptar y salir de la pantalla.

Una vez todos estos pasos se han realizado, la terminal estará lista para mostrar los datos cuando comience la ejecución del programa.

# DESCARGA Y PUESTA EN MARCHA DEL PROGRAMA

1. Dirigirse al siguiente enlace de GitHub: <https://github.com/SerPipeVera/TFG-Sensor_IoT_VIPV>
2. Pinchamos en “Code” y seguidamente en “Download ZIP”.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Una vez descargado el ZIP, extraer en la ubicación deseada del equipo.

Quedará una estructura como la mostrada a continuación:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Para iniciar el programa pinchamos en el archivo “.project”.

**NOTA**: Si es la primera vez que se utiliza el programa, es posible que nos pida que se seleccione un workspace al que se importará el programa. Si ocurre, seleccionar las opciones por defecto.

**NOTA**: También se puede dar el caso de que no se disponga de las librerías necesarias para el correcto funcionamiento del programa. De darse la situación, en algún momento del proceso pedirá autorización para instalarlas, aceptar.

**NOTA**: A la hora de abrir el programa, debido a que fue realizado con una versión anterior a la actual, saltará el aviso, dando la opción de migrarlo. Seleccionar la opción de “migrate”.

1. Una vez se ha abierto el STM32 CUBE IDE, hay que hacer ciertos cambios obligatorios para poner en marcha el sensor. Estos se detallan en la siguiente sección.

# MODIFICACIONES NECESARIAS EN EL CÓDIGO y THINGSPEAK

Como se ha mencionado anteriormente, es necesario realizar ciertas modificaciones en el código, al igual que será necesaria la creación de dos canales de ThingSpeak.

## THINGSPEAK

Este es un servicio en la nube, necesario para almacenar los datos que captura el sensor. Deberemos de iniciar sesión y crear dos canales específicos. Los detalles se muestran a continuación:

1. Dirigirse al siguiente enlace: <https://thingspeak.com/login?skipSSOCheck=true>
2. Iniciar sesión con la cuenta propia de MathWorks. En el caso de no disponer de una, debe crearse.
3. Una vez dentro se deben crear los dos canales. En la pestaña “Channels” seleccionar “New Channel”.
4. Poner un nombre al canal, y rellenar los campos denominados “Field”.
   1. Para el caso del canal 1:
      1. Field 1: Irradiancia superior
      2. Field 2: Irradiancia frontal
      3. Field 3: Irradiancia trasera
      4. Field 4: Irradiancia Derecha
      5. Field 5: Irradiancia Izquierda
      6. Field 6: Temperatura
      7. Field 7: Presión Atmosférica
      8. Field 8: Humedad
   2. Para el caso del canal 2:
      1. Field 1: Latitud Geográfica
      2. Field 2: Longitud Geográfica
      3. Field 3: Altitud geográfica
      4. Field 4: Velocidad
      5. Field 5: Inclinación Alabeo
      6. Field 6: Inclinación Cabeceo
      7. Field 7: Orientación geográfica

Únicamente es necesario rellenar estos campos. Cuando esta tarea se haya completado, pinchar en “Save Channel”.

1. Es conveniente establecer los canales como públicos. Opción editable en la pestaña “Sharing” dentro de cada canal.
2. Una vez los canales se han creado, se debe crear un “MQTT device”.

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

1. Seleccionar “Add a new device” . Establecer el nombre deseado.
2. En la sección “Authorize channels to Access”, desplegar la pestaña y seleccionar los dos canales que se han creado anteriormente.
3. Finalmente, seleccionar “Add Device”.
4. Aparecerá una nueva ventana con las credenciales del MQTT. Copiar dichas credenciales (Client ID, Username, Password) en un documento externo, ya que serán necesarias en futuros pasos.

**IMPORTANTE**: en el momento que se cierre está ventana no se puede acceder de nuevo a la contraseña del dispositivo.

1. Para poder visualizar el rastreo GPS del sensor durante la ruta hay que realizar los siguientes pasos:
   1. Desde el canal 2, pinchar en “MatlabVisualization”

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* 1. En la opción “Templates”, seleccionar “Custom”. Seguidamente “Create”.
  2. Poner el nombre deseado e introducir el código proporcionado a continuación:

%\*\*\*\*\*\*\*\*\* Programa de representacion de los datos Geograficos del Sensor IOT \*\*\*\*\*\*\*\*\*\*\*

clc; clear; grid off;

tipoMapa = 'streets'; %si no funciona ‘streets’, probar ‘openstreetmap’ o ‘satellite’

t\_inicial = datetime(1998,05,04,20,15,15); %FORMATO: datetime(año,mes,dia,hora,min,seg)

t\_final = datetime('now'); %datetime('now')

%-------------------------------------------------------------------------------------------

disp(' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GeoTracking TFG Program \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

disp( thingSpeakRead(2022893,'Fields', [1 2],'Readkey','CLB5CP76ST7WUKL3' , ...

'DateRange', [t\_inicial,t\_final],'OutputFormat','TimeTable') ); %Imprime por pantalla

data = thingSpeakRead(2022893,'Fields', [1,2],'Readkey','CLB5CP76ST7WUKL3' , ...

'DateRange', [t\_inicial,t\_final] ); %en un intervalo temporal

Latitud = data(:,1); Longitud = data(:,2);

offset = 7.55e-5; %3.1e-5 min en valde estatico

limits\_Lat = [ min(Latitud)\*(1-offset) , max(Latitud)\*(1+offset)];

limits\_Long = [ min(Longitud)\*(1+offset) , max(Longitud)\*(1-offset)];

if length(Latitud) > length(Longitud) %Busca el minimo tamaño

TAM = length(Longitud);

else TAM = length(Latitud);

end

geoscatter(Latitud, Longitud, 'pb'); %grafico de dispersion por puntos, sin unir

hold on;

title('------- GeoTracking Sensor IoT VIPV -------');

for i=2:1:TAM

geoplot([Latitud(i) Latitud(i-1)],[Longitud(i) Longitud(i-1)],'r--','LineWidth',1.5);

hold on;

end

disp(' \*\*\* Mostrando información en Mapa en el siguiente Plot \*\*\*');

geolimits(limits\_Lat,limits\_Long); %geolimits('auto')

geobasemap( tipoMapa );

**NOTA:** Modificar los siguientes comandos introduciendo el ID del canal 2, así como su “Write Key”

disp( thingSpeakRead(**XXXXXXX**,'Fields', [1 2],'Readkey','**XXXXXXXXXXXXX**' , ...

data = thingSpeakRead(**XXXXXXX**,'Fields', [1,2],'Readkey', '**XXXXXXXXXXXXX**'

1. Finalmente, en “Display Settings” seleccionar el canal 2.

## CÓDIGO

En el código se encuentran ciertos parámetros a modificar. De igual manera, se recomienda leer la bibliografía aportada por el autor del programa (GitHub) para comprender el código y sus funciones.

En relación con la configuración, son varios los parámetros que se tienen que modificar para poder hacer uso al completo del sensor.

### CONEXIÓN WIFI

Es necesario establecer el teléfono móvil como punto de acceso obteniendo del mismo los siguientes parámetros:



Estas líneas de código se localizan en el fichero “main.h”, y deberán modificarse en relación con el punto de acceso utilizado.

### CONEXIÓN THINSPEAK

De igual manera, se deben modificar los siguientes parámetros para poder conectarse con los canales de ThingSpeak creados anteriormente.

**#define** CREDENCIALES\_SERV\_IOT "HostName=mqtt3.thingspeak.com;HostPort=1883;ConnSecurity=0;MQClientId=BR4EMRE6FSArEg8uAQcGNz0;MQUserName=BR4EMRE6FSArEg8uAQcGNz0;MQUserPwd=iB2Ll9XEzKF/oqYOuLZLA9zv;"

Los parámetros “HostName”, “HostPort” y “ConnSecurity” permanecen constantes.

En los siguientes, colocar las credenciales obtenidas en el momento de crear el MQTT Device.

NOTA: En el caso de errores en la ejecución, verificar en la documentación de MatLab si el “HostName” es el correcto o se ha actualizado.

Enlace de interés para solucionar problemas: https://es.mathworks.com/help/thingspeak/troubleshoot-MQTT-publish.html

Finalmente, modificar:



Los números “2022892” y “2022893” corresponden a los identificadores de los canales de ThingSpeak.

# HARDWARE

A continuación, se adjunta un esquema eléctrico que puede resultar de gran utilidad en el caso de fallo en las conexiones o rediseño del sensor.

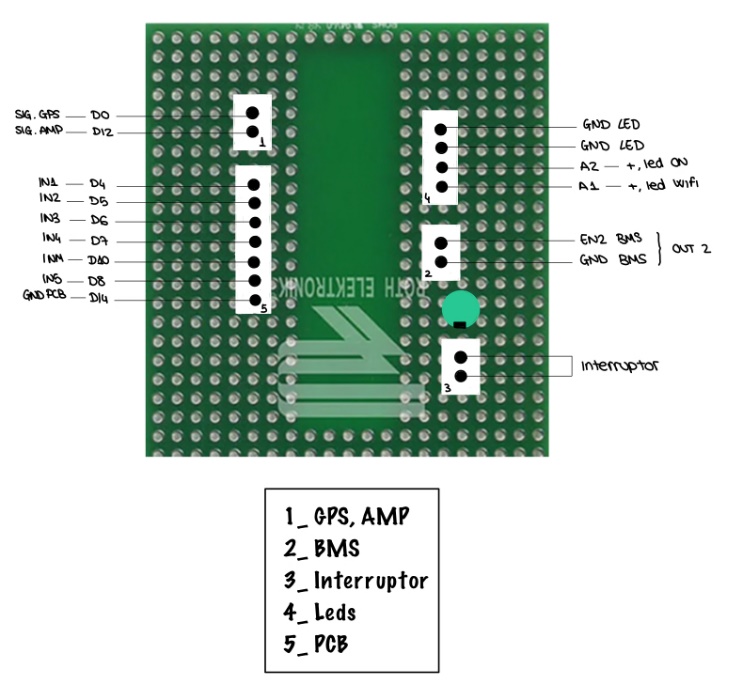
Diagrama, Esquemático

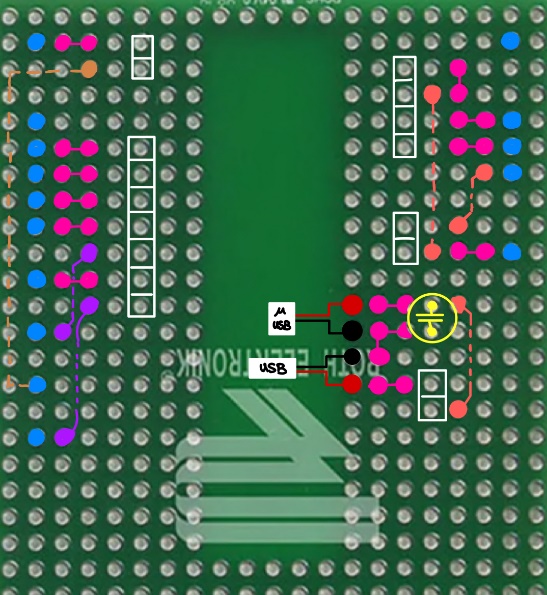
Descripción generada automáticamente

## HARDWARE UPDATE

Con el fin de hacer más robusto el sensor y evitar fallos, se ha modificado todo el cableado interno. El esquema mostrado en la parte superior continúa siendo el mismo, pero replicado sobre una pcb e incorporando conectores para hacer más fácil su reparación o desmontaje.

A continuación, se detallan los esquemas de los distintos componentes. No obstante, todos los cables han sido etiquetados para evitar confusiones.





# SOTWARE UPDATE

Con el fin de mejorar el funcionamiento del sensor, se han realizado una serie de modificaciones que se detallarán a continuación.

El principal objetivo de estas era incidir lo mínimo posible en el funcionamiento original del sensor, por lo que se han diseñado funciones completamente independientes y con la capacidad de ser desactivadas de forma sencilla, consiguiendo así volver al estado original sin complicaciones.

Estas modificaciones han sido realizadas con el fin de obtener un mayor número de muestras, reduciendo notablemente la incertidumbre.

En un principio, el software realizaba medidas con un periodo de 1 segundo, para posteriormente, a los 60 segundos, hacer una media y enviarla a la nube. A la vista de esto, nos resulto útil intentar obtener esas mediciones realizadas cada segundo, ya que serían de gran utilidad principalmente a la hora de garantizar una buena ubicación GPS.

Debido a las limitaciones del servidor de Thingspeak, es imposible enviar los datos de forma inmediata, por lo que la solución adoptada fue concatenar las medidas durante cierto tiempo, para después enviarlas en conjunto.

Para no saturar la memoria de nuestro micro, se ha reducido el tiempo de envío de datos a 10 segundos.

## MODIFICACIONES EN EL CÓDIGO

En primer lugar, ha sido necesario crear un nuevo tipo de dato. En el código anterior disponíamos de un tipo de dato que almacenaba las medidas en forma de “float”, pero esto no nos era de utilidad, ya que para concatenarlos es necesario hacerlo en forma de “string”.

La nueva estructura se muestra a continuación, definida en “sensor\_data.h”:

Tabla

Descripción generada automáticamente con confianza media

Una vez la estructura ha sido creada, el siguiente objetivo es concatenar las medidas.

Para esto, se ha utilizado el mismo vector de muestras que se emplea para hacer la media original (“vectorLecturas”), de forma que la media será exactamente la de los datos concatenados.

Para concatenarlos, primero debemos de convertir el dato que nos llega como “float” a “string”. Una vez la conversión ha sido hecha, concatenamos el dato, junto con un elemento conocido para separar las muestras. Se repite el proceso para todas las medidas tomadas.

Si se quieren más detalles, el código se encuentra comentado en el archivo .c .

Este proceso se encuentra en la función “**calcula\_concatenar**(megaDatoConcat\* datosConcat, megaDato\* p\_vectorLecturas, uint8\_t n\_elem)” dentro de “ApploT\_TFG\_VIPV.c”.

Con los datos concatenados y almacenados en la variable correspondiente, procedemos a su envío al servidor de thinskpeak.

El método es idéntico al realizado por el código original para los canales 1 y 2, salvo que replicándolo sobre unos nuevos canales 3 y 4. De esta manera garantizamos no pisar los datos originales, pudiendo a la vez contrastar la media con los datos concatenados.

Esto se encuentra en “**publica\_DatosConcatThingSpeak**(megaDatoConcat\*miDatoConcat)” dentro de “ApploT\_TFG\_VIPV.c”.

## RECUPERACIÓN DEL CÓDIGO INICIAL

Como se ha mencionado con anterioridad, es muy sencillo volver a las funcionalidades originales. Únicamente son necesarios un par de sencillos pasos para lograrlo:

1. Cambiar los tiempos de envío de datos.

En ““ApploT\_TFG\_VIPV.h” encontraremos las siguientes líneas de código:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Los valores mostrados en la imagen son los adecuados para ejecutar el código con el modo concatenar activo, para recuperar el funcionamiento original se debe establecer el periodo de publicación de datos a 60 segundos, y el periodo de recuperación de datos a 30 segundos.

1. Comentar la declaración de las funciones.

Para facilitar este proceso, en el “ApploT\_TFG\_VIPV.h” se ha definido una línea específica:



Solo hará falta comentar la línea 37 para desactivar tanto la función de concatenar como la de envío de datos concatenados.

## ¿QUÉ PASA SI SE PIERDE LA CONEXIÓN?

En el código original, si se pierde la conexión los datos comienzan a almacenarse en una FIFO, para posteriormente ser enviados cuando la conexión sea restablecida. Está función no ha sido modificada, de manera que los datos correspondientes a la media, los que se envían al canal 1 y 2, se seguirán almacenando y recuperando de acuerdo al funcionamiento normal.

Sin embargo, con los datos concatenados no se ha podido hacer lo mismo. Debido a que estos datos ocupan una cantidad considerablemente mayor de memoria, en el caso de que se pierda la conexión se dejaran de enviar, reiniciándose cada 10 segundos. Así, en el preciso instante que se recupere la conexión, los datos concatenados volverán a enviarse.

Para poder realizar esta función, se han incluido algunas líneas dentro de la función “ApploT\_TFG\_VIPV.c” :

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

De esta manera, tras concatenar los datos, si el Wifi se encuentra conectado se realiza la llamada a la función encargada de publicarlos; en cambio, si el Wifi esta desconectado, se resetean de forma inmediata.

## MEMORIA SD

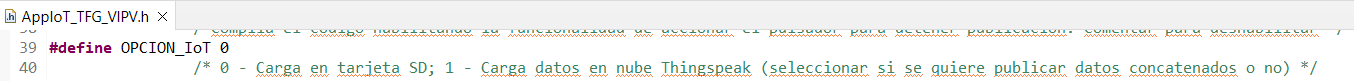
Con el objetivo de no depender de la conexión WiFi proporcionada por el móvil, se ha añadido una memoria externa (microSD) mediante el puerto PMOD de la STM32.

**HARDWARE**

Conectar el módulo PMOD en el puerto correspondiente de la placa, o mediante el alargador que lo hace accesible desde el exterior de la carcasa.

**SOWFTWARE**

Siguiendo con la dinámica del código original, se ha incluido una definición al inicio del programa con el objetivo de poder seleccionar entre un modo u otro:



Seleccionando el modo correspondiente se ejecutarán las funciones necesarias para la utilización de la SD. Estas son:

* **Inicializa\_SD():** encargada de montar la SD, obtener algunas variables como el espacio total y el disponible, obtener la fecha y hora para dar nombre al fichero de datos, y finalmente crear dicho fichero de datos.
* **Escribir\_fichero(…):** encargada de escribir en el fichero los datos pasados como parámetro. Recibe el nombre del fichero donde escribir y los datos a escribir.
* **Obencion\_dato\_SD():** obtiene los datos medidos por el sensor y los concatena generando un string. Este será posteriormente pasado a la función “escribir\_fichero” con el fin de reflejarlo en el fichero creado en la SD.