

## MANUAL DE USO Y HOJAS DE CARACTERÍSTICAS

En el presente anexo se pretende el desarrollo de un manual de uso del sistema propuesto para que la puesta en funcionamiento sea la adecuada por parte de cualquier futuro usuario. Además, se complementa el apartado con las hojas de características de los módulos más representativos del sistema para cualquier consulta necesaria.

### MANUAL DE USO

Para la correcta puesta en marcha del sistema propuesto se deben tener en cuenta ciertos factores para comprobar que todo está en óptimas condiciones que permitan que el funcionamiento sea el adecuado. El sistema propuesto consta de 3 partes muy diferenciadas entre sí, motivo por el cual se divide el desarrollo del presente apartado.

#### **Placa Arduino**

La parte más importante del sistema diseñado reside en la placa y en todo lo que está conectado a ella, a continuación, se enumeran las pautas que se deben tener en cuenta a la hora de la puesta en marcha de la placa Arduino:

1. Comprobar que el cableado esté correcto como se indica en la Tabla 4 de la memoria. Ser muy cuidadoso con intercambiar por error el cableado de la alimentación con los módulos y sustituir *G* por *V*, que pese a que tienen protección interna no es conveniente y se puede ocasionar daños en el módulo que ocurra. Si es necesario se podrá quitar la tapa del encapsulado del sistema para tener más accesible los cables.
2. Comprobar que en el módulo que corresponde al *shield* de la SD esté incorporada la tarjeta, ya que al extraer los datos y por un descuido puede ser que siga en el ordenador.
3. Comprobar que el reloj de tiempo real (RTC) tenga su batería incorporada con autonomía. Si durante el primer uso se observa que la hora y fecha no es la actual es debido a que ha entrado en hibernación durante su inactividad, la solución reside en compilar y subir de nuevo el código ya que hay una sentencia que establece la fecha y hora del compilador en ese momento al reloj.
4. Alimentar la placa Arduino del modo que se prefiera, ya sea por medio de una batería externa, desde el propio dron o desde el cable USB propio de Arduino para conectarlo al ordenador.
5. Una vez alimentado comprobar los siguientes puntos:

- LEDs encendidos en MQ-2 y MQ-135
  - LEDs encendidos en Pixhawk
  - Sensor SDS011 empieza a funcionar (se puede observar el ventilador girar y un pequeño zumbido característico del correcto funcionamiento)
  - Módulo de telemetría encendido con un LED verde y rojo parpadeando. El verde corresponde a la sincronización con el módulo receptor, que cuando esté conectado se establecerá la comunicación entre ambos y se quedará fijo. El rojo sirve para verificar que está enviando-recibiendo datos, parpadeando solamente cuando se realice cualquier tarea de las anteriores.
6. Si está conectado al ordenador, abrir el monitor serie correspondiente desde el IDE de Arduino (con una tasa de bits de 57600 baudios) y observar que no se manda ningún mensaje de error. Al principio el código está diseñado para que se establezca la comunicación con todos los módulos que no se puedan verificar a simple vista y se actualice su estado por pantalla.

Tras realizar los pasos anteriores la implementación correspondiente a la placa Arduino deberá funcionar correctamente y estar lista para realizar las mediciones (como se podrá observar si se realiza el paso 6.) Si se detecta cualquier error se tendrá que volver a revisar el cableado del sistema. Se recomienda que se mantenga el sistema encendido media hora aproximadamente antes de tomar las medidas para que los sensores se calienten y sean más sensibles y precisos, esto también ayudará a que el GPS se estabilice y sea más preciso.

### **Monitorización en Tiempo Real**

La puesta en funcionamiento de esta parte resulta ser la más sencilla de todas, y como se ha visto, depende del funcionamiento del apartado anterior. Se parte en base a que se ha realizado la puesta en marcha del sistema embebido en la placa Arduino y este ha sido satisfactorio. Los pasos necesarios para que la monitorización en tiempo real sea funcional son los enumerados a continuación:

1. Abrir el software Spyder con el código correspondiente. (También debe funcionar con otros entornos de programación en Python, como PyCharm)
2. Conectar el módulo de telemetría que funciona como receptor en cualquier puerto USB del ordenador. El LED verde deberá permanecer fijo en señal de conexión con el módulo emisor.

3. Comprobar que la tasa de datos sea la impuesta en Arduino en el *Serial 2* (57600 baudios).
4. Elegir el puerto USB en el que está conectado el receptor y comprobar que coincida con el del código (ver Figura 51). Si existe alguna discrepancia se puede cambiar manualmente. Para saber el puerto USB en el que está conectado el receptor, basta con abrir el IDE de Arduino y ver los diferentes puertos que se pueden elegir, ya que solo deja acceder a los conectados en ese momento.

Tras realizar lo anterior solamente será necesario darle al botón de *Play* situado en la barra de navegación superior, cuyo símbolo es un triángulo verde y, a continuación, se empezarán a visualizar los datos recibidos por telemetría. Si se quiere parar se podrá realizar de varias formas, extrayendo el módulo receptor o pulsando el botón de *Stop* situado en las proximidades del anterior.

### **Análisis de Datos en Diferido**

Para utilizar esta parte del sistema es necesario extraer el fichero generado en la tarjeta SD con la extensión .csv durante la recogida de datos. Los pasos necesarios para su puesta en marcha son los siguientes:

1. Subir el archivo .csv utilizando la primera línea de código y cargarlo utilizando la sentencia que va después. En este punto es importante que el nombre coincida exactamente (ver Figura 54).
2. Subir las leyendas que se utilizarán en las representaciones utilizando la primera sentencia del código de nuevo.

Tras realizar lo anterior se podrá navegar por el código ejecutando las partes que se quieran analizar. Si se desea visualizar otros datos generados durante una toma de datos posterior se tendrá que volver a subir el fichero (y si ha pasado mucho tiempo las leyendas también). Se recomienda generar copias de los ficheros generados en cada prueba y renombrarlos. No hay que olvidarse de devolver la tarjeta SD al módulo de la placa Arduino.

### **HOJAS DE CARACTERÍSTICAS**

Con el objetivo de facilitar el acceso a estos datos proporcionados por el fabricante para futuras consultas, a continuación, se exponen las hojas de características (o también conocidas como *datasheets*) de los sensores incluidos en el sistema propuesto:

**Sensor Mq-2:** accesible en <https://www.mouser.com/datasheet/2/321/605-00008-MQ-2-Datasheet-370464.pdf>

**Sensor Mq-135:** accesible en [https://www.electronicoscaldas.com/datasheet/MQ-135\\_Hanwei.pdf](https://www.electronicoscaldas.com/datasheet/MQ-135_Hanwei.pdf)

**Sensor BME280:** accesible en <https://www.mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf>

**Sensor CCS811:** accesible en [https://cdn.sparkfun.com/assets/2/c/c/6/5/CN04-2019\\_attachment\\_CCS811\\_Datasheet\\_v1-06.pdf](https://cdn.sparkfun.com/assets/2/c/c/6/5/CN04-2019_attachment_CCS811_Datasheet_v1-06.pdf)

**Sensor SDS011:** accesible en <https://cdn-reichelt.de/documents/datenblatt/X200/SDS011-DATASHEET.pdf>

En estas hojas de características el fabricante muestra los parámetros más relevantes del sensor/módulo, como, por ejemplo, la tensión requerida para su alimentación, la potencia consumida, etc. También se puede encontrar información acerca de su sensibilidad (en muchas ocasiones se muestra utilizando gráficas o ecuaciones) precisión, rango y la correcta conexión de sus pines. Para entender mejor su funcionamiento, en la mayoría de los casos se suele incorporar un diagrama de bloques con sus componentes y conexiones internas.

Otro módulo de interés que está integrado en el sistema y que puede ser necesario consultar información acerca de él es el módulo de reloj en tiempo real que permite tener registros de fecha y hora:

**RTC DS3231:** accesible en <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

En lo que respecta a la placa de Arduino utilizada, es conveniente tener un rápido acceso al *pinout* de esta para futuras incorporaciones, esta información la facilita el fabricante y se muestra a continuación:

**Arduino Mega 2560 (*pinout*):** accesible en [https://content.arduino.cc/assets/Pinout-Mega2560rev3\\_latest.pdf](https://content.arduino.cc/assets/Pinout-Mega2560rev3_latest.pdf)