



Como hacer un Datalogger

22/05/2018

Fecha	22/05/2018
Autor	Laura Coronel
Versión	1.1
Revisado por	Rubén Mir

## **TABLA DE CONTENIDO**

1. Introducción.....	2
2. Requisitos.....	3
2.1. Hardware.....	3
2.2. Software .....	3
3. Conexiones.....	3
3.1. Módulos.....	3
3.2. Sd.....	3
3.3. Interconexión SDIO con SPI en Helios.....	3
3.4. Conexión de la placa 2 (Emisor de datos) con placa 1 (datalogger) .....	4
4. Ejemplo de uso .....	5
4.1. Objetivo .....	5
4.2. Procedimiento .....	5
5. Formas de realizar esta prueba.....	9
6. Disclaimer .....	10

## **1. INTRODUCCIÓN**

Un datalogger o registrador de datos es un dispositivo que recibe datos de otros dispositivos y los almacena en un archivo. Dichos archivos de datos reciben el nombre de “logs”. Los Logs se utilizan para supervisar el comportamiento de un dispositivo y llevar a cabo un seguimiento. Con todo ello, podemos localizar posibles errores y/o depurar problemas. Suelen estar conectados largos períodos de tiempo para recoger la mayor cantidad de datos posible.

Entre sus posibles aplicaciones se encuentran:

- Estación meteorológica no supervisada. (Temperatura, Humedad, Velocidad y dirección del viento, Luz solar, lluvia caída, humedad del suelo).
- Registro hidrológico (Nivel del agua, caudal, conductividad del agua, PH del agua)
- Boyas marinas para el registro de meteorológico.
- Registradores de tránsito. (Cantidad, velocidad, peso por eje, altura de los vehículos)
- Medición para control de calidad de transporte de alimentos o medicamentos perecederos durante la cadena de frío.
- Registradores de investigación de fauna salvaje. (Seguimiento de tortugas marinas, collares para osos etc)

- Registradores con acelerómetros para control de altura de caída y vibraciones durante el transporte de encomiendas delicadas.
- Análisis del perfil de carga de consumo a la red eléctrica de industrias.
- La caja negra de los aviones que registra innumerables variables de vuelo.
- El Controlador de motor de los vehículos registra eventos para identificar posibles fallas intermitentes.
- Registrador de Datos de la Travesía se utiliza en barcos de mediano o gran porte y es obligatorio en los mayores de 300 toneladas.

## 2. REQUISITOS

### 2.1. HARDWARE

- 1x Helios
- 1x Placa de clase II o clase III. En este ejemplo utilizaremos Deimos.
- 2x Módulo Máster Duino Series
- 6x cables de conexión
- 1x Tarjeta micro SD

### 2.2. SOFTWARE

- IDE de Arduino
- Sketch de Datalogger: sd-logger.ino
- Sketch para emisor: sd-serial-test.ino
- Librería SD.h incluida en el IDE de Arduino
- Librería SoftwareSerial.h incluida en el IDE de Arduino

\* Nota: Estas librerías ya vienen incorporadas de serie en el IDE de Arduino, por lo que no es necesario realizar acciones extra.

## 3. CONEXIONES

Por favor, revisar el siguiente documento para más información:

- Inserción de módulos

### 3.1. MÓDULOS

Insertar cada uno de los dos Duino uno/mega Rhomb.io master module en cada una de las dos placas de clase 2 o clase 3.

### 3.2. SD

Insertar una SD formateada en FAT32 en el SDHolder

### 3.3. INTERCONEXIÓN SDIO CON SPI EN HELIOS

Para poder escribir y leer datos de las tarjetas SD existe el bus SDIO. Sin embargo, se trata de un standard cerrado y no es compatible con ciertos módulos master. Por este motivo,

*Tecnofingers S.L.*

*Page 3 of 10*

vamos a emplear el bus SPI que es open source y perfectamente compatible. Para conseguirlo, simplemente vamos a interconectar un bus con otro por medio de unos cables.

Para ello necesitaremos:

- 4xcables de conexión

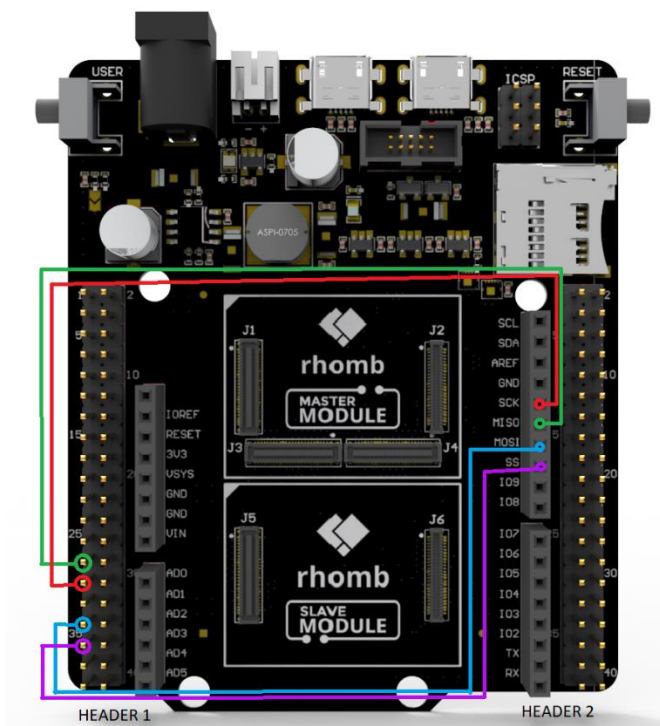
Haremos las conexiones de esta forma:

- El pin 27 macho (o Data0) del header 1, irá conectado mediante cableado al pin 6 hembra (MISO) del header 2. **Verde**

- El pin 29 macho (o CLK) del header 1, irá conectado mediante cableado al pin 5 hembra (SCK/CLK) del header 2. **Rojo**

- El pin 33 macho (o CMD) del header 1, irá conectado mediante cableado al pin 7 hembra (MOSI) del header 2. **Azul**

- El pin 35 macho (o SDIOCS) del header 1, irá conectado mediante cableado al pin SS del header 2. **Violeta**

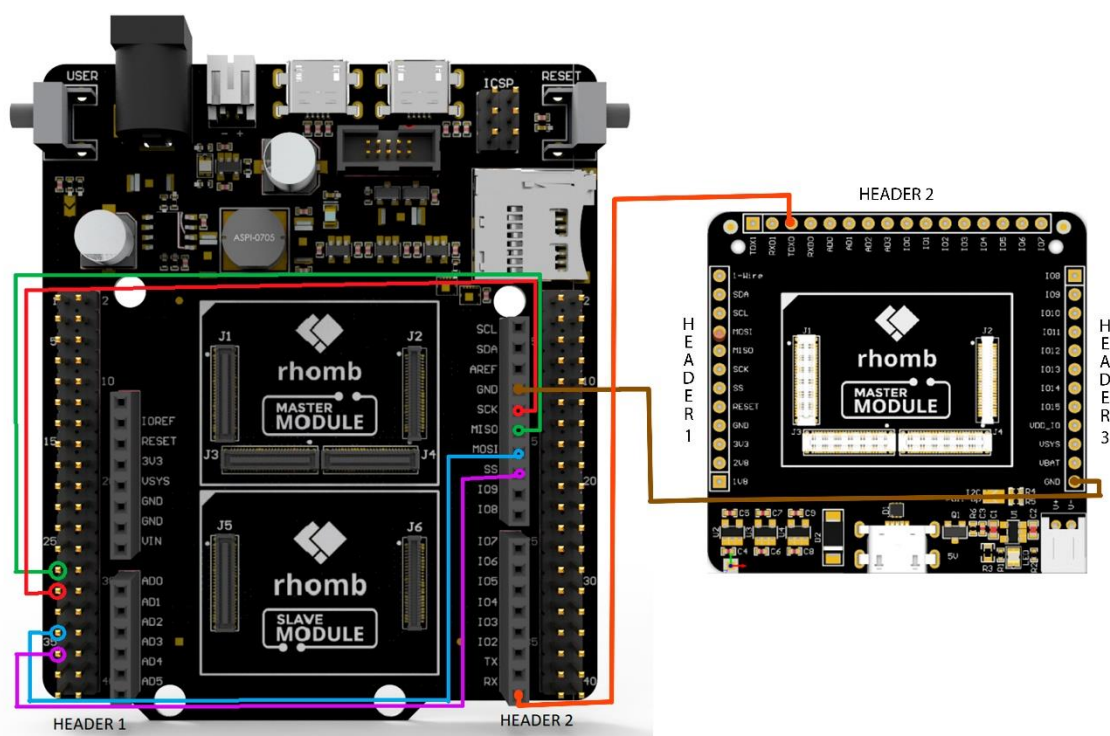


*Puentear el bus SDIO con el bus SPI 1*

### 3.4. CONEXIÓN DE LA PLACA 2 (EMISOR DE DATOS) CON PLACA 1 (DATALOGGER)

Conectaremos mediante un cable, el puerto RX del Header 2 de la Helios al puerto TDX0 del Header 2 de la Deimos. Básicamente se trata de conectar el Tx del puerto serie de la placa emisora con el Rx de la placa receptora.

A continuación, interconectamos la masa de ambas placas. En este caso el puerto GND de la Deimos con uno de los pins GND de la Helios.



Esquema de conexión entre placas

## 4. EJEMPLO DE USO

Para llevar a cabo este ejemplo, es recomendable la lectura previa de los siguientes documentos:

- Flashear módulos master Duino Series en placas clase II y III

### 4.1. OBJETIVO

En este ejemplo se van a transmitir datos por puerto serie desde una placa de clase II o III a una Helios. La Helios almacenará en una tarjeta micro SD todos los ratos recibidos. Se emplearán archivos de texto tipo "log".

### 4.2. PROCEDIMIENTO

En este caso, la placa que envía los datos es una Deimos, pero podríamos haber empleado cualquier otra de clase II o III. Vamos a flashear en esta placa el siguiente sketch:

- sd-serial-test.ino

La placa receptora, es decir, la que actúa como datalogger es una Helios. Vamos a flashear en esta placa el siguiente sketch:

- sd-logger.ino

Antes de realizar el flasheo de la placa receptora (Helios) vamos a hacer un pequeño cambio en el código del sketch que va a depender de qué modelo de módulo Master usemos (Duino uno o Duino Mega). La modificación consiste en comentar o descomentar una línea del código.

- Para comentar una línea escribiremos `//` al principio de la línea
- Para descomentar una línea, borraremos `//`

```
5 /*****RHOMB.IO*****/
6 #define SD_PIN 10 //descomentar para usar con Duino Uno Master Module
7 //#define SD_PIN 53 // descomentar para usar con Duino Mega 2560 Master Module
8
9 /*****RHOMB.IO*****/
```

*Fragmento de código a personalizar en cada sketch según placa utilizada*

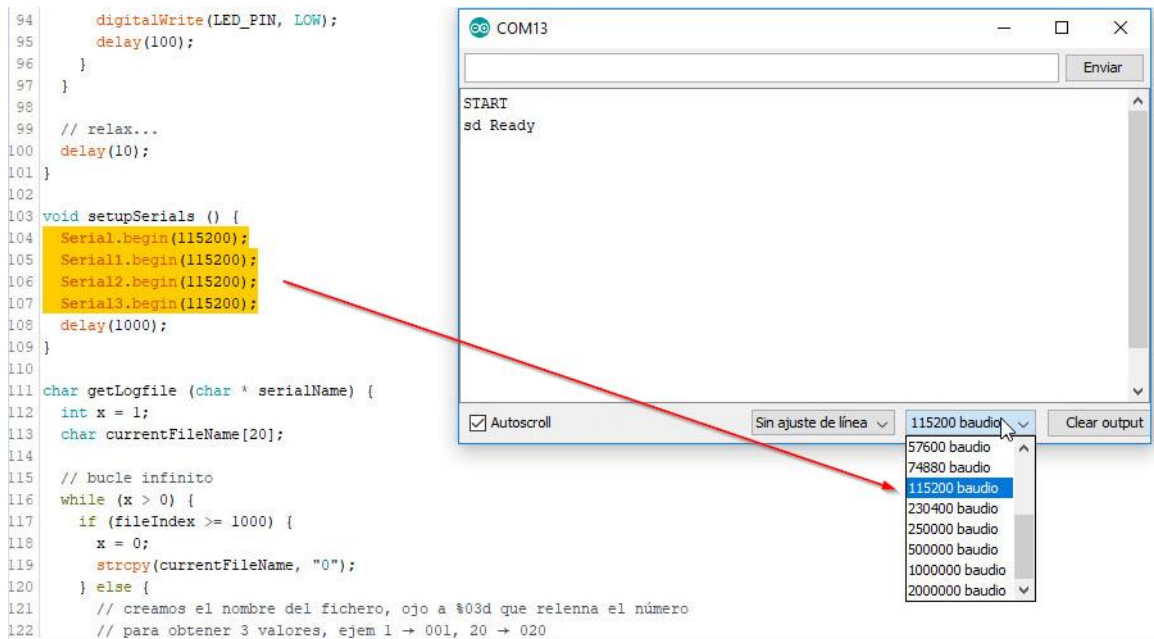
1. Descomentar la línea apropiada para nuestro módulo Master. Dejar comentadas el resto.
2. Realizar el flasheo. Una vez flasheado, abrir el monitor serie (Ctrl+Mayus+T) y verificar que nos haya reconocido bien la tarjeta micro SD insertada con anterioridad en la placa. Debe poner "sd Ready". Dejar esta placa conectada al PC y con el monitor serie abierto.



*Imagen de la Consola indicándonos que reconoce la SD*

Ahora vamos a conectar la otra placa a la corriente, o bien a un PC. En el caso de que usemos un PC, abrir el monitor serie en el IDE de Arduino de tal modo que tengamos el monitor serie de ambas placas. La que envía (Deimos) y la que recibe y loguea (Helios).

Debemos asegurarnos de que el Baudrate del monitor serie corresponde con el del sketch. Si no es así simplemente hay que cambiarlo en uno de los dos sitios (sketch, o monitor serie).



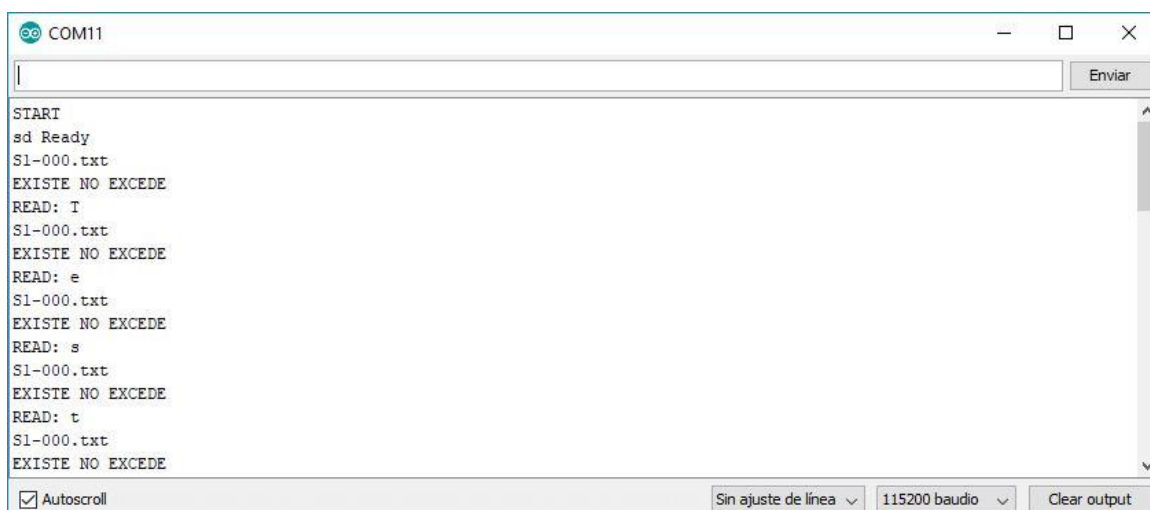
Ambos sketches entrarán en funcionamiento y en el monitor serie debemos poder ver como aparece texto en ambos sketches. Dejamos funcionando ambas placas unos minutos.



Monitor Serie del emisor

En el monitor serie del emisor, vamos a ver como aparece la palabra test varias veces. Este es el dato que se está enviando.

En el monitor serie de la placa receptora, vamos a ver lo siguiente:



```

START
sd Ready
S1-000.txt
EXISTE NO EXCEDE
READ: T
S1-000.txt
EXISTE NO EXCEDE
READ: e
S1-000.txt
EXISTE NO EXCEDE
READ: s
S1-000.txt
EXISTE NO EXCEDE
READ: t
S1-000.txt
EXISTE NO EXCEDE
  
```

#### Monitor serie del receptor

Esta es la explicación de los datos que se muestran en el monitor serie de la placa receptora:

**sd Ready** : Quiere decir que la tarjeta Sd la reconoce y está en funcionamiento.

**S1-000.txt** : Es el archivo donde se están guardando los datos del emisor.

**EXISTE NO EXCEDE**: Significa que el archivo donde está guardando los datos existe, y no lo tiene que volver a crear, y que además no ha excedido el límite establecido para el archivo. Cuando se alcanza el límite, se generará un nuevo archivo en orden consecutivo (S1-001.txt..)

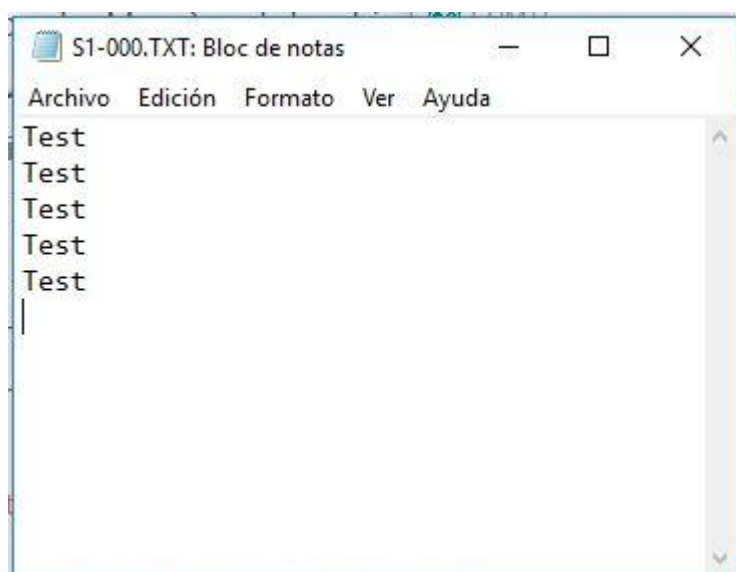
**READ: T, e, s, t** : Nos muestra el bit que está recibiendo en ese momento. Si recordamos, el script del emisor imprimía la palabra Test en el monitor serie. Cada una de esas letras es el dato que se está recibiendo.

Podemos comprobar que ha guardado datos sacando la tarjeta SD y conectándola al pc mediante un adaptador microSD.

Dentro debe haber un archivo .txt llamado SX-00X.txt. Nuestro archivo se llama S1-000.txt. Si lo abrimos nos aparece el contenido.

Cuando ha pasado algún tiempo o el archivo excede de cierto tamaño, se produce el cierre del archivo de texto y se continúa logueando en uno nuevo con un nombre de archivo consecutivo. De este modo si se produce un fallo en mitad de la escritura de un archivo, tendremos salvaguardados los anteriores.





Contenido del archivo de la SD

## 5. FORMAS DE REALIZAR ESTA PRUEBA

Según los recursos de los que dispongamos, podemos realizar la prueba de distintas formas. Los resultados en los tres métodos son idénticos. Lo que cambia de un método a otro es la monitorización de los scripts.

Formas de realizar esta prueba		
Placa 1 conectada a un PC y Placa 2 conectada a otro PC	Posibilidad de tener monitor serie en ambas placas	Es la forma utilizada en este ejemplo. Se requieren 2 PC
Placa 1 conectada a un PC y Placa 2 conectada a +5V	Posibilidad de tener monitor serie en una placa	Se requiere 1 PC
Placa 1 y placa 2 conectadas a +5 V	Sin monitorizar el puerto serie	No se requiere PC

Notas:

- Durante la prueba, se ofrece información relevante a través del puerto serie. Es recomendable tener la placa conectada a un PC para acceder a esta información a través del monitor serie del IDE de Arduino (Ctrl+Mayus+M). No es imprescindible en una aplicación real.
- La alimentación a +5V puede obtenerse desde un alimentador externo, desde una batería o desde el puerto USB de un PC.

## 6. DISCLAIMER

Todos los derechos reservados. Ninguna parte de este documento puede ser fotocopiada, reproducida, almacenada en un sistema en la nube, o transmitido, en cualquier forma o por cualquier medio, ya sea electrónico, mecánico o de otro tipo, sin el previo consentimiento o permiso por escrito de Tecnofingers S.L. No se garantiza la exactitud del contenido de la información contenida en esta publicación. En la medida en que lo permita la ley, no se aceptará ninguna responsabilidad (incluida la responsabilidad frente a cualquier persona por negligencia) por parte de Tecnofingers o cualquiera de sus subsidiarias o empleados por cualquier pérdida o daño directo o indirecto causado por omisiones de o de inexactitudes en este documento. Tecnofingers S.L. se reserva el derecho de cambiar los detalles de esta publicación sin previo aviso. Nombres de productos y empresas el presente documento puede ser una marca comercial de sus respectivos propietarios.