

## Guide de mise en marche du shield data-logger GT1046

Matériel nécessaire :

- 1 x carte compatible Uno®
- 1 x [cordon USB-B](#) pour la programmation
- 1 x shield data-logger [GT1046](#)
- 1 x carte SD ou [micro-SD](#) avec adaptateur



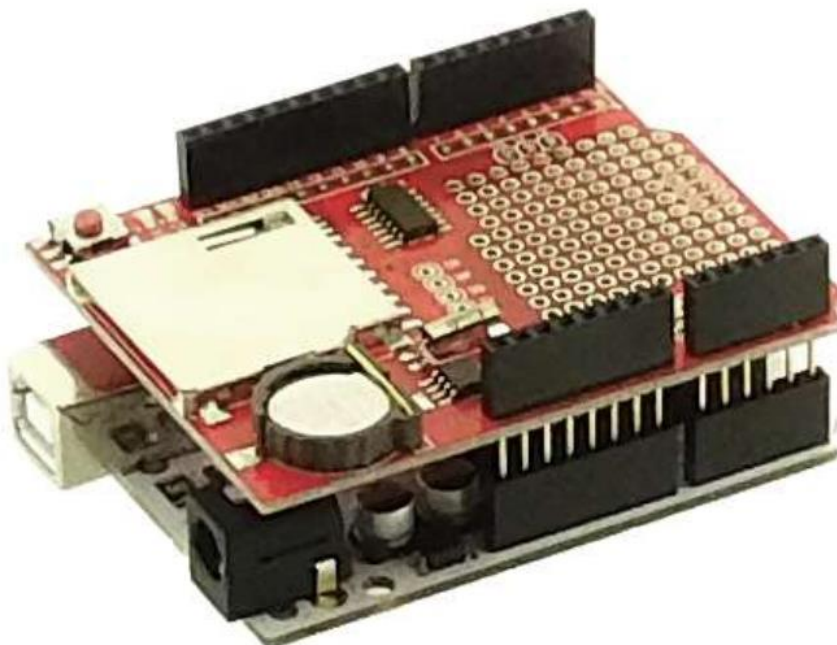
### Présentation du module :

Shield data logging permettant le stockage sur carte SD (non incluse). Il est équipé d'une horloge en temps réel (pile CR1220 incluse) pour connaître l'heure exacte de l'enregistrement.

Ce shield est compatible avec les cartes Uno, Leonardo, Mega R3 ou compatibles et comporte un bouton reset et un espace de prototypage.

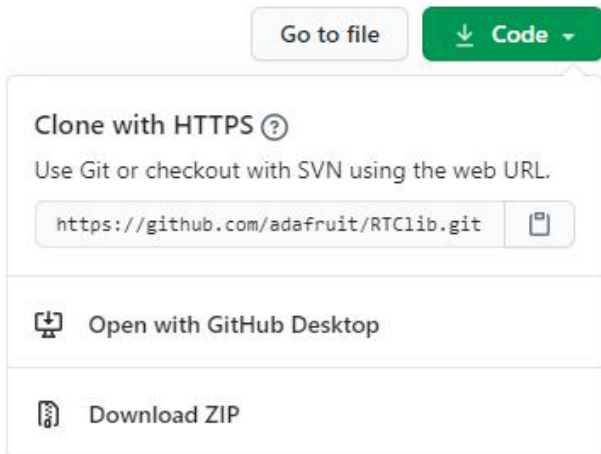
Compatible avec les cartes SD FAT16 ou FAT32  
Horloge temps réel incluse (pile CR1220 incluse)  
Dimensions: 67 x 54 x 12 mm  
Carte Uno non incluse.  
Version: 1.0

### Connexion du shield :



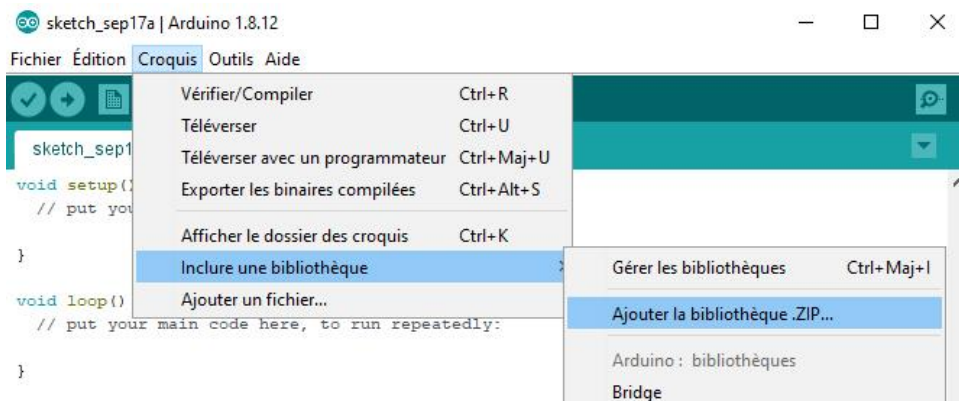
## Exemple d'utilisation du module RTC :

Pour cet exemple, il est nécessaire de télécharger une librairie disponible à [cette adresse](#).



Cliquez sur « Code » puis sur « Download Zip ». Cette action téléchargera un fichier compressé au format .ZIP contenant la librairie.

Ajoutez cette bibliothèque via le menu « **Croquis** », en parcourant votre ordinateur jusqu'à votre dossier de téléchargement.



L'exemple de code proposé est une version modifiée du code donné avec la librairie.

La date et l'heure actuelles sont à définir sur la ligne « RTC.adjust ».

Cet exemple est à copier/coller dans l'IDE Arduino®.

```
// Date and time functions using a DS1307 RTC connected
// via I2C and Wirelib
// <jc@wippler.nl> http://opensource.org/licenses/mit-license.php

#include <Wire.h>
#include <RTClib.h>
RTC_DS1307 RTC;

void setup ()
{
  Serial.begin(9600); // Initialisation de la liaison série
  Wire.begin();
  RTC.begin();

  // Cette ligne permet de définir une heure de départ pour le module RTC
  //(Année, mois, jour, //heure, minute, seconde)
  // Elle peut être mise en commentaire une fois le module mis à jour.

  RTC.adjust(DateTime(2017, 11, 3, 16, 49, 0)); //
}

void loop ()
{
  DateTime now = RTC.now();
  Serial.print(now.day(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.println(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  delay(3000);
}
```

Le programme peut ensuite être envoyé dans la carte Arduino.

La date et l'heure sont visualisable dans le moniteur série de l'IDE.



## Exemple de programme pour le lecteur de carte SD :

Le programme et la librairie nécessaire pour l'utilisation du lecteur de carte SD sont directement intégrés dans l'IDE Arduino®.

L'exemple CardInfo est situé dans Fichier -> Exemples -> SD.

Une modification pour une utilisation avec ce shield est nécessaire :

La ligne : `const int chipSelect = 4;`

Doit être mise à jour vers : `const int chipSelect = 10;`

Cette modification effectuée et la carte SD installée dans le shield, il est possible de visualiser les informations de la carte via le moniteur série de l'IDE.

The screenshot shows the serial monitor output for an Arduino program. The text is as follows:

```

Initializing SD card...Wiring is correct and a card is present.

Card type: SDHC

Volume type is FAT32

Volume size (bytes): 3548606464
Volume size (Kbytes): 3465436
Volume size (Mbytes): 3384

Files found on the card (name, date and size in bytes):
14840.PDF    2017-01-03 10:36:46 845713
14069.PDF    2016-08-10 15:39:34 64674
14260.PDF    2016-12-21 17:02:52 5525322
14539.PDF    2016-08-09 08:34:42 1246608
  
```

At the bottom of the window, there are controls for the serial monitor: a checked checkbox for "Défilement automatique", a dropdown menu for "Nouvelle ligne", a dropdown menu for "9600 baud", and a button for "Effacer la sortie".