RTC Data Logger Shield v1.1

From Elecrow

Contents

Introduction

The RTC Data Logger Shield v1.1 adds storage to your Arduino project. It supports SD, SDHC, or MicroSD TF cards. Use the on-board toggle switch to select the SD card type. The card supports use of one and only one format of SD card, either SD/SDHC, or MicroSD. Before using an SD or MicroSD card, please set the shields "SELECT" switch to the proper setting: toward the SD/SDHC slot, or the MicroSD slot.Also it with a RTC on-board,that make you project more perfect.

The RTC Data Logger Shield v1.1 uses the SPI and IIC ports of Arduino. The shield is also stackable, so additional shields can go on top.

The RTC Data Logger Shield v1.1 makes use of pin 4 as a chip select. Please be sure other stacked shields do not make use of that same pin.

Model: ASC7503RTC (http://www.elecrow.com/rtc-data-logger-shield-p-1370.html)

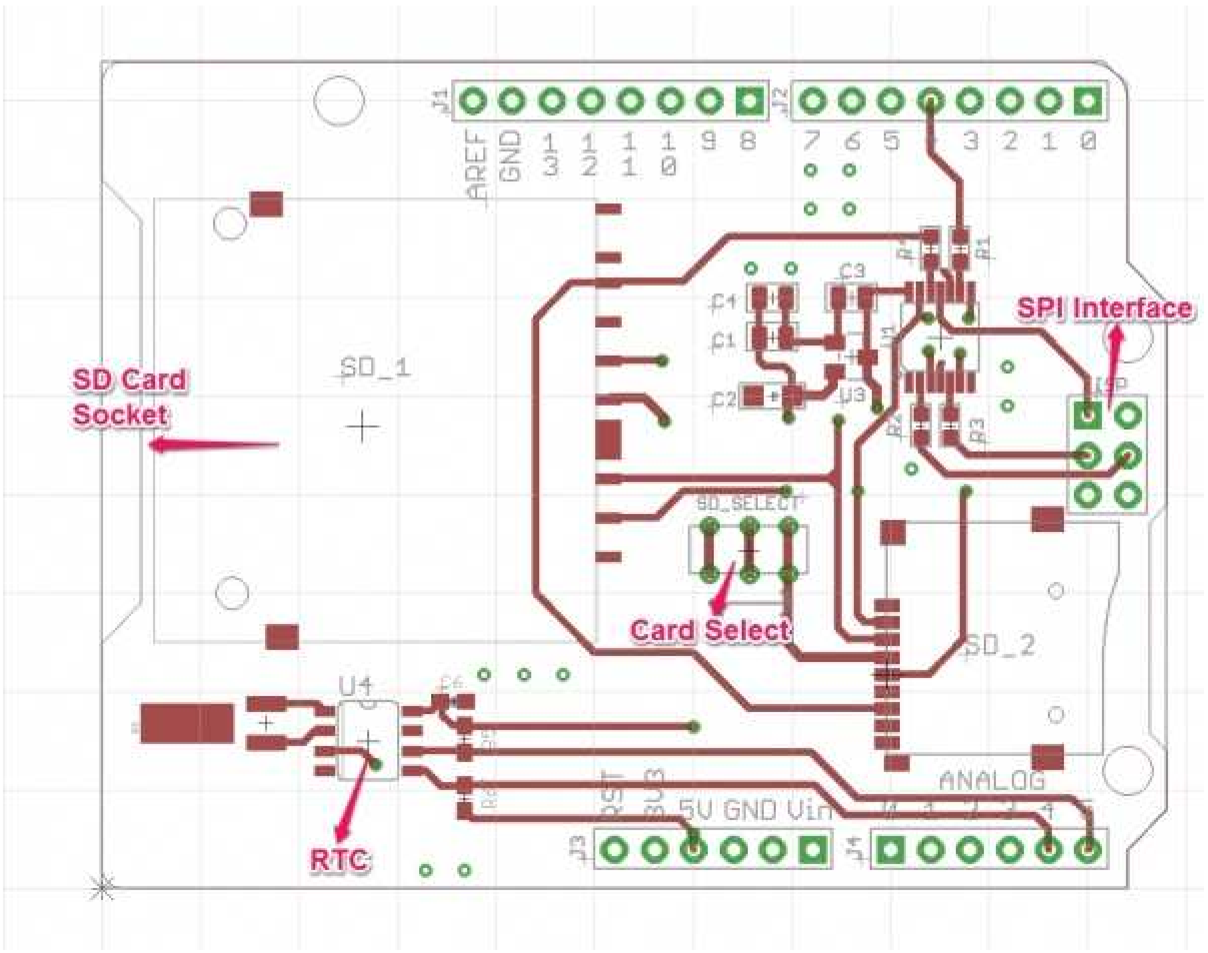
Features

- Arduino/Crowduino compatible(Do not compatible with Mega)
- SD card, Micro SD card and SDHC card supportable
- 3.3v and 5v logical voltage compatible
- 2.6~3.6v DC power supply

Specification

Item	Min	Typical	Max	Unit
Voltage	2.7	3.3	3.6	V
Current	0.159	40	200	mA
Supported Card Type	SD card(<=2G); Micro SD card(<=2G); SDHC card(<=16G)			/
Dimension	68x53x23			mm
Net Weight	22			g

Interface Function

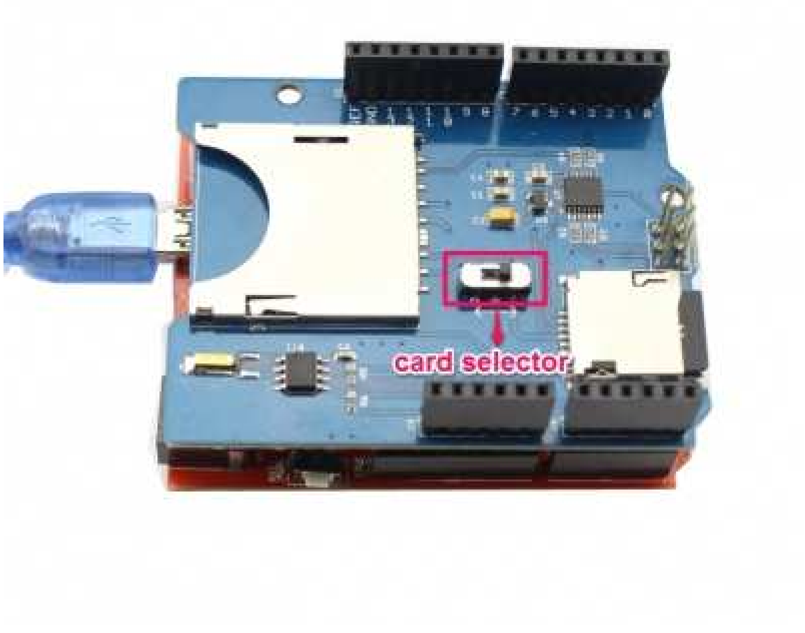


D10 – Used for CS of SPI
D11 – Used for MOSI of SPI
D12 – Used for MISO of SPI
D13 – Used for SCK of SPI

Usage

Hardware Installation

Plug the RTC Data Logger Shield v1.1 onto the Arduino; Insert your SD card into the socket and make sure the card selector pointing to the right way (Standard card or micro).And then connect the Arduino to PC with USB cable.
Warning: Don't insert SD card and microSD card into the shield at the same time.

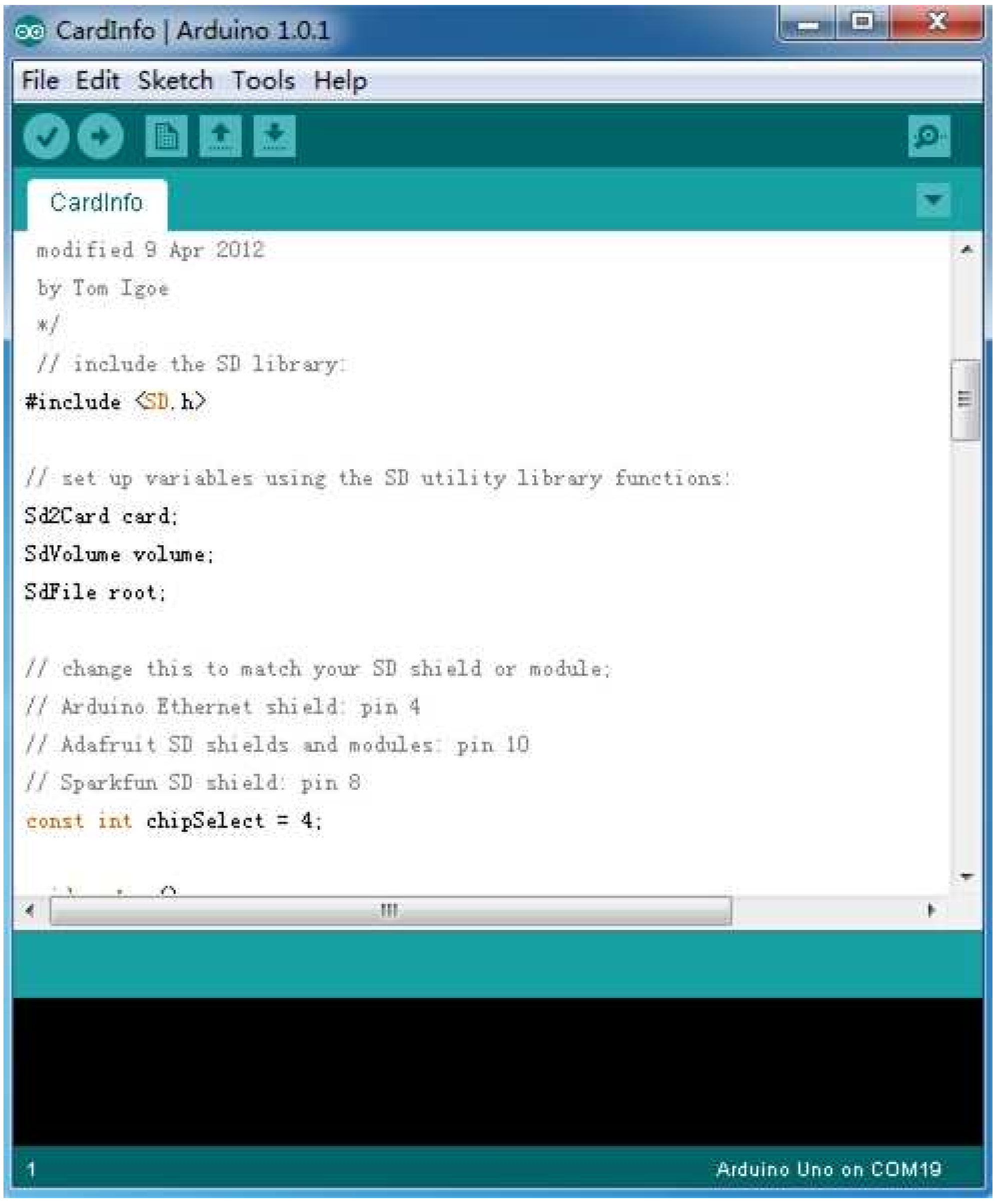


Upload the program

1. Restart the Arduino IDE. Open "CardInfo"example via the path: File --> Examples --> SD --> CardInfo.

This example shows how use the utility libraries on which the SD library is based in order to get info about your SD card.Very useful for testing a card when

you're not sure whether its working or not. There are also many other examples in this library, like “ReadWrite”. You can always try them out.



```
 modified 9 Apr 2012
 by Tom Igoe
 */
 // include the SD library:
#include <SD.h>

// set up variables using the SD utility library functions:
Sd2Card card;
SdVolume volume;
SdFile root;

// change this to match your SD shield or module;
// Arduino Ethernet shield: pin 4
// Adafruit SD shields and modules: pin 10
// Sparkfun SD shield: pin 8
const int chipSelect = 4;
```

There has a brief description to above code:

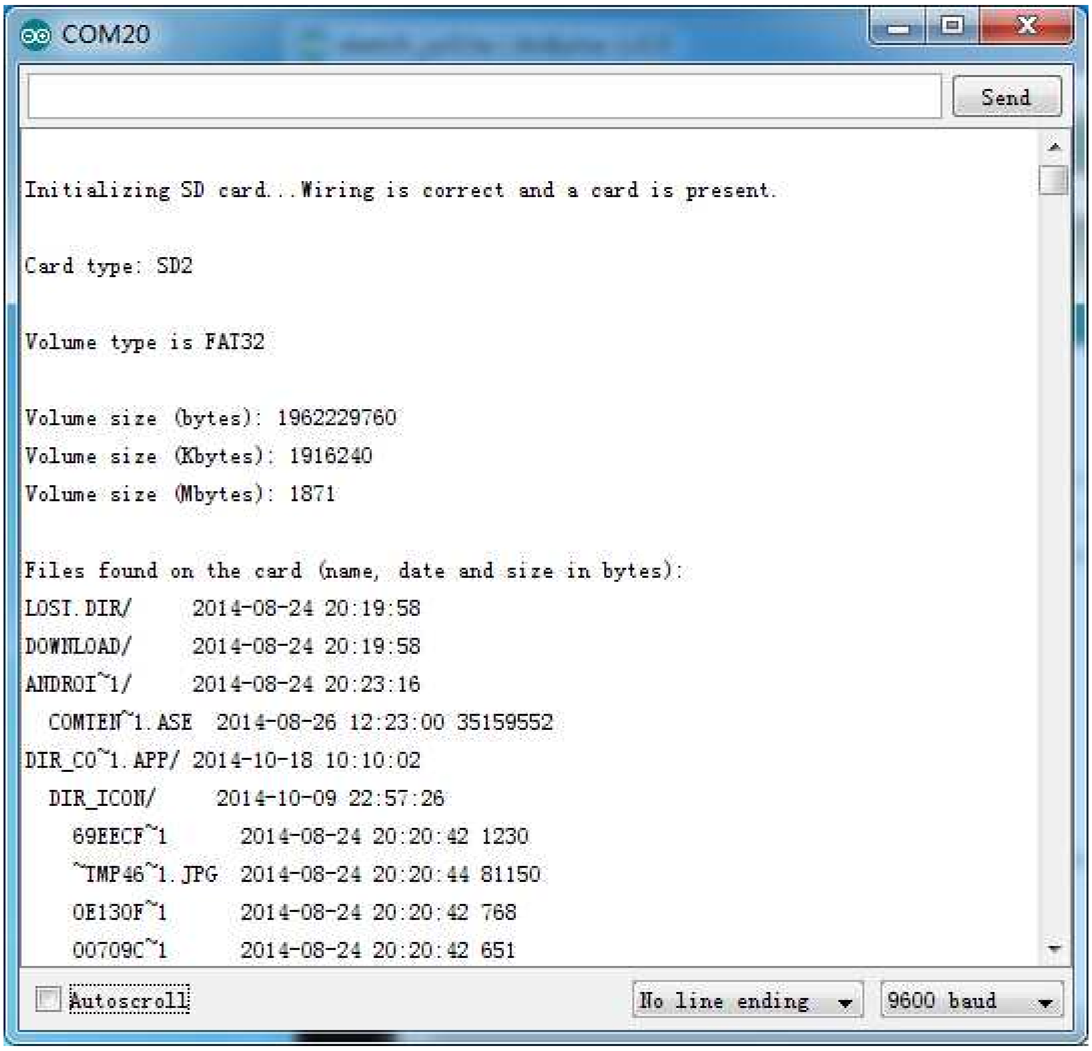
First checks if the card is working. if not, there will output some reasons that may lead to this result.

In the case of the SD card normal work, it will print the SD card type.Then print the type and size of the FAT-type volume.

Finally, get the files information such as name, date and size in bytes that found on the card.

2. Upload the code.

3. View Results.You can see the follow image after Click the serial monitor.



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

Card type: SD2

Volume type is FAT32

Volume size (bytes): 1962229760
Volume size (Kbytes): 1916240
Volume size (Mbytes): 1871

Files found on the card (name, date and size in bytes):
LOST.DIR/     2014-08-24 20:19:58
DOWNLOAD/     2014-08-24 20:19:58
ANDROI~1/     2014-08-24 20:23:16
  COMTEN~1.ASE  2014-08-26 12:23:00 35159552
DIR_CO~1.APP/ 2014-10-18 10:10:02
  DIR_ICON/     2014-10-09 22:57:26
    69EECF~1      2014-08-24 20:20:42 1230
    ~TMP46~1.JPG  2014-08-24 20:20:44 81150
    0E130F~1      2014-08-24 20:20:42 768
    00709C~1      2014-08-24 20:20:42 651
```

4. If an error occurs, please recheck all the previous steps, and make sure the SD card is working. If none of that fixes the problem, try replacing the SD card.

Note

Arduino default code return SD size incorrectly if your SD card more than 4G. The following code can solve this issue.

```
/*
  SD card test

 This example shows how use the utility libraries on which the'
 SD library is based in order to get info about your SD card.
 Very useful for testing a card when you're not sure whether its working or not.

 The circuit:
  * SD card attached to SPI bus as follows:
 ** MOSI - pin 11 on Arduino Uno/Duemilanove/Diecimila
 ** MISO - pin 12 on Arduino Uno/Duemilanove/Diecimila
 ** CLK - pin 13 on Arduino Uno/Duemilanove/Diecimila
 ** CS - depends on your SD card shield or module.
 ** Pin 4 used here for consistency with other Arduino examples


 created  28 Mar 2011
 by Limor Fried
 modified 9 Apr 2012
 by Tom Igoe
 */
```

```
// include the SD library:
#include <SPI.h>
#include <SD.h>

// set up variables using the SD utility library functions:
Sd2Card card;
SdVolume volume;
SdFile root;

// change this to match your SD shield or module;
// Arduino Ethernet shield: pin 4
// Adafruit SD shields and modules: pin 10
// Sparkfun SD shield: pin 8
const int chipSelect = 4;

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }


  Serial.print("\nInitializing SD card...");
  // On the Ethernet Shield, CS is pin 4. It's set as an output by default.
  // Note that even if it's not used as the CS pin, the hardware SS pin
  // (10 on most Arduino boards, 53 on the Mega) must be left as an output
  // or the SD library functions will not work.
  pinMode(10, OUTPUT);     // change this to 53 on a mega


  // we'll use the initialization code from the utility libraries
  // since we're just testing if the card is working!
  if (!card.init(SPI_HALF_SPEED, chipSelect)) {
    Serial.println("initialization failed. Things to check:");
    Serial.println("* is a card is inserted?");
    Serial.println("* Is your wiring correct?");
    Serial.println("* did you change the chipSelect pin to match your shield or module?");
    return;
  } else {
    Serial.println("Wiring is correct and a card is present.");
  }

  // print the type of card
  Serial.print("\nCard type: ");
  switch (card.type()) {
    case SD_CARD_TYPE_SD1:
      Serial.println("SD1");
      break;
    case SD_CARD_TYPE_SD2:
      Serial.println("SD2");
      break;
    case SD_CARD_TYPE_SDHC:
      Serial.println("SDHC");
      break;
    default:
      Serial.println("Unknown");
  }

  // Now we will try to open the 'volume'/'partition' - it should be FAT16 or FAT32
  if (!volume.init(card)) {
    Serial.println("Could not find FAT16/FAT32 partition.\nMake sure you've formatted the card");
    return;
  }


  // print the type and size of the first FAT-type volume

  uint64_t volumesize64;
  uint32_t volumesize32;
  Serial.print("\nVolume type is FAT");
  Serial.println(volume.fatType(), DEC);
  Serial.println();

  volumesize64 = volume.blocksPerCluster();    // clusters are collections of blocks
  volumesize64 *= volume.clusterCount();       // we'll have a lot of clusters
  volumesize64 *= 512;                            // SD card blocks are always 512 bytes

  Serial.print("Volume size (bytes): ");
  printLLNumber(volumesize64, DEC);
  Serial.println();

  Serial.print("Volume size (Kbytes): ");
  volumesize32 = volumesize64/1024;
  Serial.println(volumesize32);

  Serial.print("Volume size (Mbytes): ");
  volumesize32 /= 1024;
  Serial.println(volumesize32);
  /*uint64_t volumesize;
  Serial.print("\nVolume type is FAT");
  Serial.println(volume.fatType(), DEC);
  Serial.println();

  volumesize = volume.blocksPerCluster();    // clusters are collections of blocks
  volumesize *= volume.clusterCount();       // we'll have a lot of clusters
  volumesize *= 512;                            // SD card blocks are always 512 bytes
  Serial.print("Volume size (bytes): ");
  Serial.println(volumesize,DEC);
  Serial.print("Volume size (Kbytes): ");
  volumesize /= 1024;
  Serial.println(volumesize,DEC);
  Serial.print("Volume size (Mbytes): ");
  volumesize /= 1024;
  Serial.println(volumesize,DEC);
*/

  Serial.println("\nFiles found on the card (name, date and size in bytes): ");
  root.openRoot(volume);

  // list all files in the card with date and size
  root.ls(LS_R | LS_DATE | LS_SIZE);
}


void loop(void) {

}
void printLLNumber(uint64_t n, uint8_t base)
{
  unsigned char buf[16 * sizeof(long)];
  unsigned int i = 0;

  if (n == 0)
  {
    Serial.print((char)'0');
    return;
  }

  while (n > 0)
  {
    buf[i++] = n % base;
    n /= base;
  }

  for (; i > 0; i--)
    Serial.print((char) (buf[i - 1] < 10 ?
      '0' + buf[i - 1] :
      'A' + buf[i - 1] - 10));
}
```

The usage of RTC

1.Download the library File:RTC Library (http://www.elecrow.com/wiki/index.php?title=File:RTC.zip)

2.Unzip it into the libraries file of Arduino IDE by the path: ..\arduino-1.0\libraries.

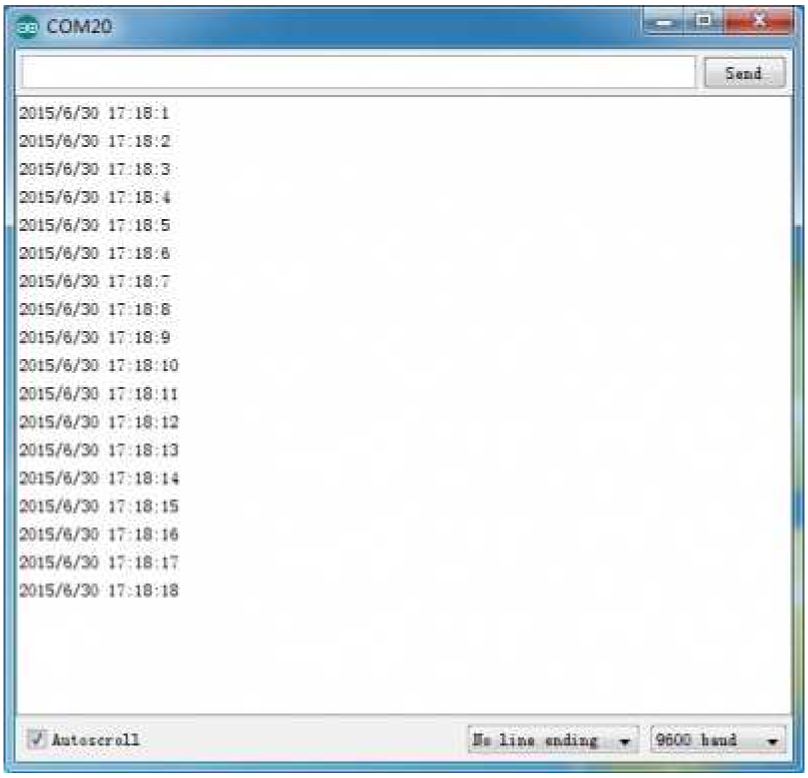
3.Open the code directly by the path:File -> Example ->RTC.

```
#include <Wire.h>
```

```
#include "RTClib.h"
RTC_DS1307 RTC;
void setup () {
   Serial.begin(9600);
   Wire.begin();
   RTC.begin();

 if (! RTC.isrunning()) {
   Serial.println("RTC is NOT running!");
   // following line sets the RTC to the date & time this sketch was compiled
   RTC.adjust(DateTime(__DATE__, __TIME__));
 }
}
void loop () {
   DateTime now = RTC.now();
   Serial.print(now.year(), DEC);
   Serial.print('/');
   Serial.print(now.month(), DEC);
   Serial.print('/');
   Serial.print(now.day(), DEC);
   Serial.print(' ');
   Serial.print(now.hour(), DEC);
   Serial.print(':');
   Serial.print(now.minute(), DEC);
   Serial.print(':');
   Serial.print(now.second(), DEC);
   Serial.println();
   delay(1000);
}
```

4.Upload the code,then open the serial monitor to see the result.



```
2015/6/30 17:18:1
2015/6/30 17:18:2
2015/6/30 17:18:3
2015/6/30 17:18:4
2015/6/30 17:18:5
2015/6/30 17:18:6
2015/6/30 17:18:7
2015/6/30 17:18:8
2015/6/30 17:18:9
2015/6/30 17:18:10
2015/6/30 17:18:11
2015/6/30 17:18:12
2015/6/30 17:18:13
2015/6/30 17:18:14
2015/6/30 17:18:15
2015/6/30 17:18:16
2015/6/30 17:18:17
2015/6/30 17:18:18
```

FAQ

1. SD card can't be initialized .

- Please reformat SD card to FAT/FTA32 .If it still can't work ,suggest to download the SD Formatter (https://www.sdcard.org/downloads/formatter_3/) and reformat SD card by this software.

Resources

- RTC Data Logger Shield v1.1 eagle files (http://www.elecrow.com/wiki/index.php?title=File:RTC_Data_Logger_Shield_v1.1.zip)
