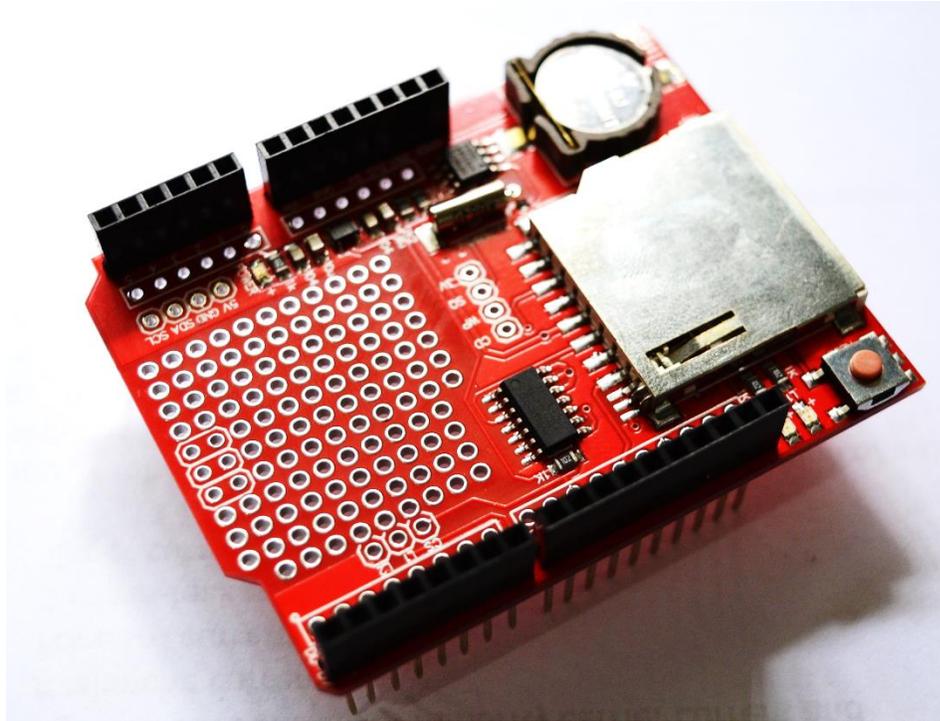


Keyes Data Logging Shield (Rev 1)



General Description

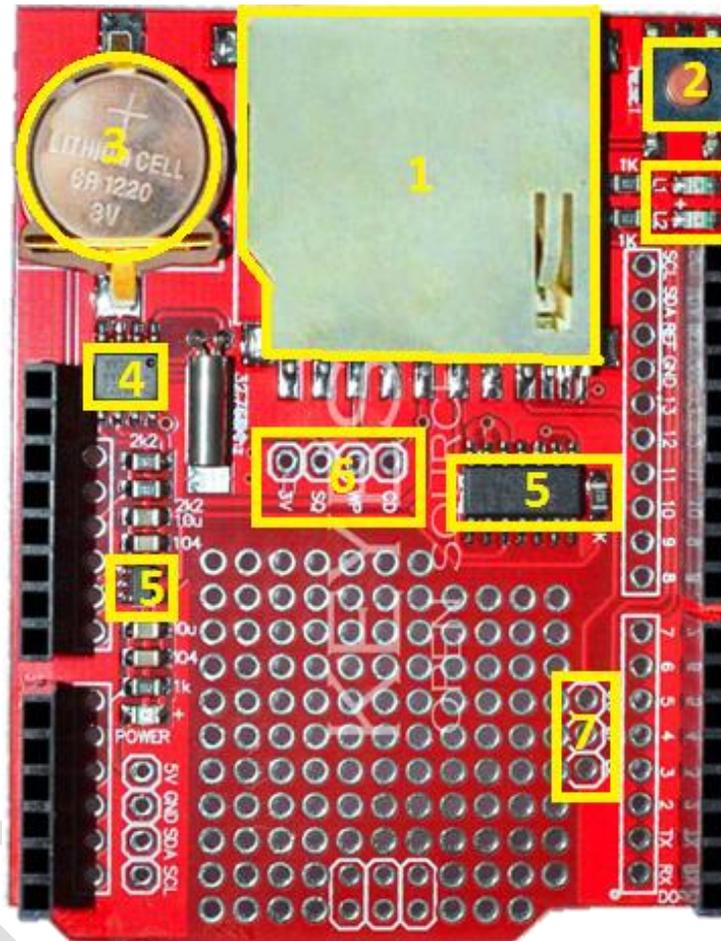
Keyes' Data Logging Shield is an inexpensive yet sophisticated designed shield for most available Arduino boards. It is an SD card module and RTC (real-time clock) module combined into one forming a shield for data logging.

SD Card module is used as a storage to store data such as images, documents and logs. You can save data files on any FAT16 or FAT32 formatted SD card.

The RTC included allows your microcontroller project to keep track of time even if it is reprogrammed, or if the power is lost. Perfect for data logging, clock-building, time stamping, timers and alarms, etc. The DS1307 is the most popular RTC, and works best with 5V-based chips such as the Arduino.

Also included in this shield is a 3V lithium cell battery for the RTC.

Shield Overview



1. An SD card holder can fit any SD/MMC storage up to 8GB and as small as 32MB. If you have a MicroSD card, there are low cost adapters which will let you fit these in. SD cards are tougher to lose than MicroSD. Simply Push or Pull the card into this slot.
2. Reset
3. 3V Lithium Cell Battery
4. DS1307 – low-power, full binary-coded decimal (BCD) series real-time clock.
5. 74HC125D - quad buffer/line driver
6. Extra Breakout:
3V - this is the 3V out of the regulator with up to 50mA available current.

SQ - this is the optional Square wave output from the RTC which can be enabled through coding.

WP - this is the Write Protect pad on the SD card, you can use this to detect if the write-protect tab is on the card by checking this pin

CD - this is the card detect pad on the SD card. When this is connected to ground, an SD card is inserted. We suggest using the internal pullup on an Arduino pin if you want to use this pad

7. Configurable LED Indicator and Chip Select

CS - this is the **Chip Select** pin for the SD card. If you need to cut the connection to pin 10 because it is conflicting, this pad can be soldered to any digital pin.

L2 and **L1** - these are optional user-LEDs with a 470 ohm resistors already in series. Connect to any digital pin, pull high to turn on the corresponding LED.

How to use RTC to get current date and time

You need:

- Arduino Uno
- Keyes Data Logging Shield

1. Mount the Keyes Data logging shield into your Arduino.
2. Download [RTCLib](#) and extract it to library folder in your Arduino directory.
3. Enter this code into your Arduino IDE then click upload.

```
// Date and time functions using a DS1307 RTC connected via I2C and Wire lib
#include <Wire.h>
#include "RTCLib.h"

RTC_DS1307 RTC;

void setup () {
  Serial.begin(57600);
  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
    // uncomment it & upload to set the time, date and start run the
    // RTC!
    //RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}
```

```

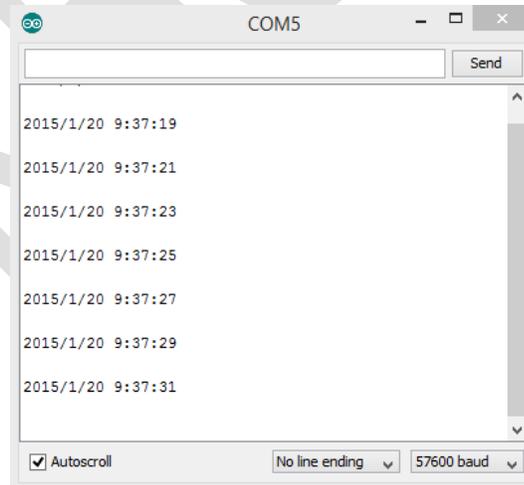
void loop () {
    DateTime now = RTC.now();

    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();
    Serial.println();
    delay(2000);
}

```

NOTE: When it is your first time using the shield, the RTC might not be set to function yet. To do this uncomment `//RTC.adjust(DateTime(__DATE__, __TIME__));`. This will set the current time and date of the RTC into your PC time and date.

4. Click the Serial Button to see the results. Make sure the baud rate is set to 57600 baud.



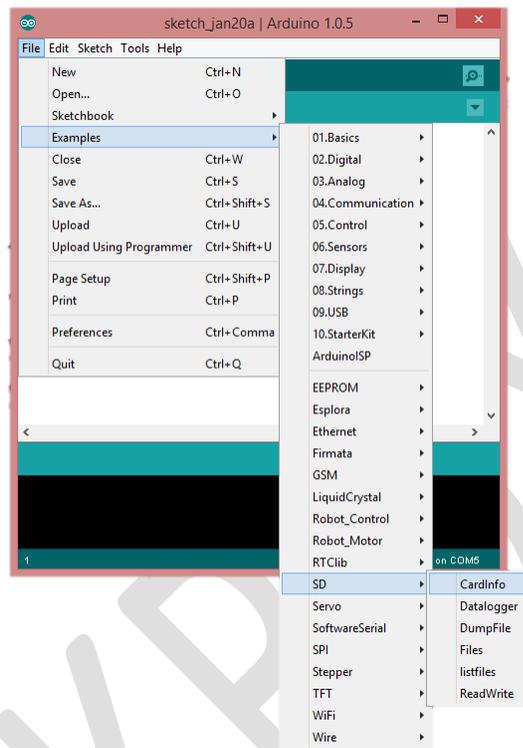
How to use SD Card Reader

You need:

- Arduino Uno
- Keyes Data Logging Shield
- SD Card/ MicroSD with adapter

1. Mount the Keyes Data logging shield into your Arduino.
2. Insert your SD card into the SD card slot.
3. Format the SD card using FAT16/FAT32 files system
(We recommend using [SD Formatter](#) to prevent issues)

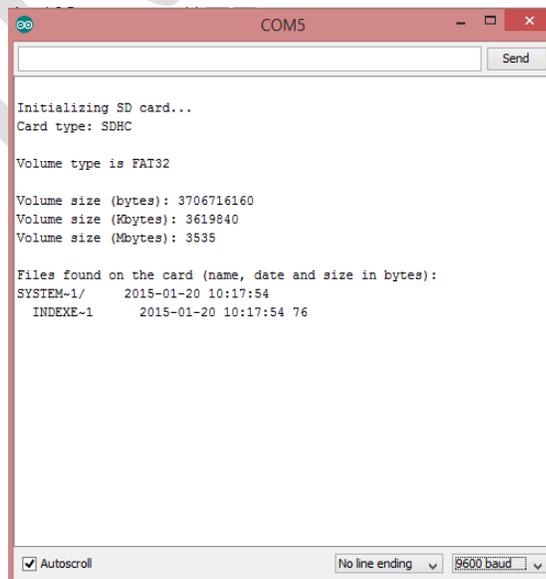
4. Download [SD library](#) and extract it to library folder in your Arduino directory.
5. Open Cardinfo sketch.

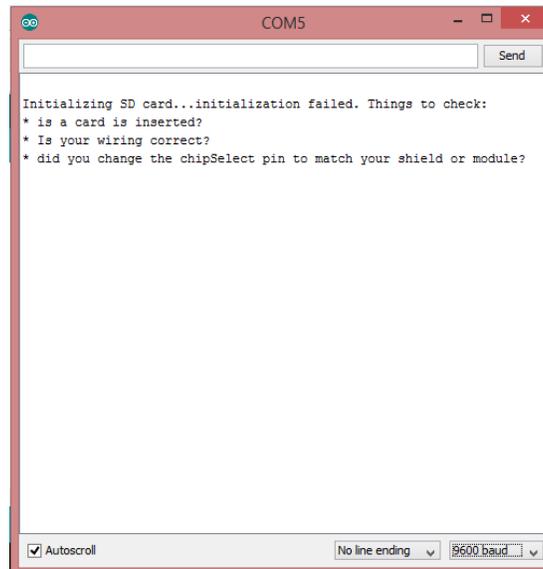


6. Change this line `const int chipSelect = 4;` to `const int chipSelect = 10;`
7. Click the Serial Button to see the results. Make sure the baud rate is set to 57600 baud.

You will be able to see this Serial Monitor if you have a working SD Card and is compatible to the shield.

If you get this results:





This means you either have a damaged SD Card or you missed a step in this How-to.

Sample Data Logging Program

Now let's try the shield as one. I wanted to know when there is a movement at the front of my door so I've decided to log the time and date whenever a motion detected.

It is very simple but this concept can act as a trigger to a bigger application! So, let's start!

You need:

Arduino

Data Logging Shield

Mini PIR Motion Sensor

Sketch

```
#include <SD.h>
#include <SPI.h>
#include <Wire.h>
#include "RTClib.h"

// the last 10 reads if power is lost but it uses less power and is much faster!
#define SYNC_INTERVAL 1000 // mills between calls to flush() - to write data to the card
uint32_t syncTime = 0; // time of last sync()
```

```

#define ECHO_TO_SERIAL 1 // echo data to serial port
#define WAIT_TO_START 0 // Wait for serial input in setup()

// the digital pins that connect to the LEDs

RTC_DS1307 RTC; // define the Real Time Clock object\

int ledPin = 13; // choose the pin for the LED
int inputPin = 2; // choose the input pin (for PIR sensor)
int pirState = LOW; // we start, assuming no motion detected
int val = 0; // variable for reading the pin status
// for the data logging shield, we use digital pin 10 for the SD cs li
ne
const int chipSelect = 10;

// the logging file
File logfile;

void error(char *str)
{
    Serial.print("error: ");
    Serial.println(str);

    // red LED indicates error

    while(1);
}

void setup(void)
{
    Serial.begin(9600);
    Serial.println();

    // use debugging LEDs

#if WAIT_TO_START
    Serial.println("Type any character to start");
    while (!Serial.available());
#endif //WAIT_TO_START

    // initialize the SD card
    Serial.print("Initializing SD card...");
    // make sure that the default chip select pin is set to
    // output, even if you don't use it:
    pinMode(10, OUTPUT);

    // see if the card is present and can be initialized:
    if (!SD.begin(chipSelect)) {
        error("Card failed, or not present");
    }
    Serial.println("card initialized.");

    // create a new file
    char filename[] = "LOGGER00.CSV";
    for (uint8_t i = 0; i < 100; i++) {

```

```

    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
        // only open a new file if it doesn't exist
        logfile = SD.open(filename, FILE_WRITE);
        break; // leave the loop!
    }
}

if (! logfile) {
    error("couldnt create file");
}

Serial.print("Logging to: ");
Serial.println(filename);

// connect to RTC
Wire.begin();
if (!RTC.begin()) {
    logfile.println("RTC failed");
#ifdef ECHO_TO_SERIAL
    Serial.println("RTC failed");
#endif //ECHO_TO_SERIAL
}

logfile.println("millis,stamp,datetime,status");
#ifdef ECHO_TO_SERIAL
    Serial.println("millis,stamp,datetime,status");
#endif //ECHO_TO_SERIAL
}

void loop(void)
{
    val = digitalRead(inputPin); // read input value
    if (val == HIGH) { // check if the input is HIGH
        digitalWrite(ledPin, HIGH); // turn LED ON
        if (pirState == LOW) {
            // we have just turned on
            logData();
            Serial.print(", ");
            Serial.println("Motion detected!");
            logfile.print(", ");
            logfile.println("Motion Detected!!");
            logfile.flush();
        }
        // We only want to print on the output change, not state
        pirState = HIGH;
    }
}
else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    if (pirState == HIGH){
        logData();
        // we have just turned of
        Serial.print(", ");

```

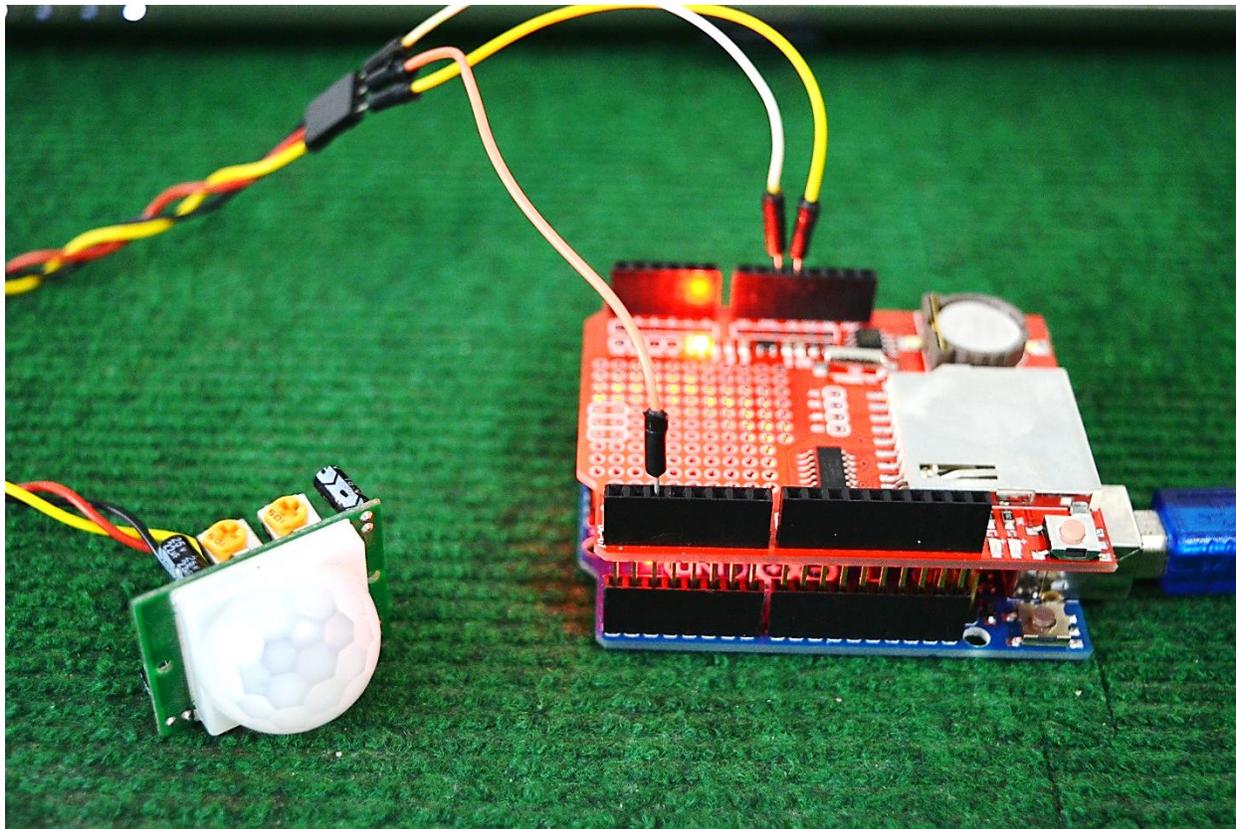
```

        Serial.println("Motion ended!");
        logfile.print(", ");
        logfile.println("Motion Ended!!");
        logfile.flush();
        // We only want to print on the output change, not state
        pirState = LOW;
    }
}

void logData(){
    DateTime now;
    // log milliseconds since starting
    uint32_t m = millis();
    logfile.print(m);           // milliseconds since start
    logfile.print(", ");
#if ECHO_TO_SERIAL
    Serial.print(m);           // milliseconds since start
    Serial.print(", ");
#endif
    now = RTC.now();
    // log time
    logfile.print(now.unixtime()); // seconds since 1/1/1970
    logfile.print(", ");
    logfile.print(' ');
    logfile.print(now.year(), DEC);
    logfile.print("/");
    logfile.print(now.month(), DEC);
    logfile.print("/");
    logfile.print(now.day(), DEC);
    logfile.print(" ");
    logfile.print(now.hour(), DEC);
    logfile.print(":");
    logfile.print(now.minute(), DEC);
    logfile.print(":");
    logfile.print(now.second(), DEC);
    logfile.print(' ');
#if ECHO_TO_SERIAL
    Serial.print(now.unixtime()); // seconds since 1/1/1970
    Serial.print(", ");
    Serial.print(' ');
    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.print(' ');
#endif //ECHO_TO_SERIAL
}

```

Actual Setup



Results

LOGGER03.CSV - Excel

	A	B	C	D	E	F	G	H	I
1	millis	stamp	datetime	status					
2	3313	1421763373	"2015/1/20 14:16:13"	Motion Detected!!					
3	15596	1421763385	"2015/1/20 14:16:25"	Motion Ended!!					
4	19657	1421763389	"2015/1/20 14:16:29"	Motion Detected!!					
5	25757	1421763395	"2015/1/20 14:16:35"	Motion Ended!!					
6	33576	1421763403	"2015/1/20 14:16:43"	Motion Detected!!					
7	36615	1421763406	"2015/1/20 14:16:46"	Motion Ended!!					
8	42059	1421763412	"2015/1/20 14:16:52"	Motion Detected!!					
9	43557	1421763413	"2015/1/20 14:16:53"	Motion Ended!!					
10	65496	1421763435	"2015/1/20 14:17:15"	Motion Detected!!					
11	68520	1421763438	"2015/1/20 14:17:18"	Motion Ended!!					
12	74893	1421763444	"2015/1/20 14:17:24"	Motion Detected!!					
13	85612	1421763455	"2015/1/20 14:17:35"	Motion Ended!!					
14	89700	1421763459	"2015/1/20 14:17:39"	Motion Detected!!					
15	97489	1421763467	"2015/1/20 14:17:47"	Motion Ended!!					
16	102288	1421763472	"2015/1/20 14:17:52"	Motion Detected!!					
17	104946	1421763474	"2015/1/20 14:17:54"	Motion Ended!!					
18	110538	1421763480	"2015/1/20 14:18:00"	Motion Detected!!					
19	114554	1421763484	"2015/1/20 14:18:04"	Motion Ended!!					
20	120582	1421763490	"2015/1/20 14:18:10"	Motion Detected!!					
21	123514	1421763493	"2015/1/20 14:18:13"	Motion Ended!!					
22	140912	1421763510	"2015/1/20 14:18:30"	Motion Detected!!					
23	143750	1421763513	"2015/1/20 14:18:33"	Motion Ended!!					
24	151158	1421763521	"2015/1/20 14:18:41"	Motion Detected!!					

COM5

```
Initializing SD card...card initialized.
Logging to: LOGGER03.CSV
millis,stamp,datetime,status
3313, 1421763373, "2015/1/20 14:16:13", Motion detected!
15596, 1421763385, "2015/1/20 14:16:25", Motion ended!
19657, 1421763389, "2015/1/20 14:16:29", Motion detected!
25757, 1421763395, "2015/1/20 14:16:35", Motion ended!
33576, 1421763403, "2015/1/20 14:16:43", Motion detected!
36615, 1421763406, "2015/1/20 14:16:46", Motion ended!
42059, 1421763412, "2015/1/20 14:16:52", Motion detected!
43557, 1421763413, "2015/1/20 14:16:53", Motion ended!
65496, 1421763435, "2015/1/20 14:17:15", Motion detected!
68520, 1421763438, "2015/1/20 14:17:18", Motion ended!
74893, 1421763444, "2015/1/20 14:17:24", Motion detected!
85612, 1421763455, "2015/1/20 14:17:35", Motion ended!
89700, 1421763459, "2015/1/20 14:17:39", Motion detected!
97489, 1421763467, "2015/1/20 14:17:47", Motion ended!
```