

# Arduino Data Logger Shield

## 1. POPIS

Arduino Data Logger Shield umožní vývojovým kitům Arduino (UNO, Duemilanove, Diecimila, ADK/Mega R3 a vyšší) záznam dat na externí SD kartu (nelze použít samostatně).

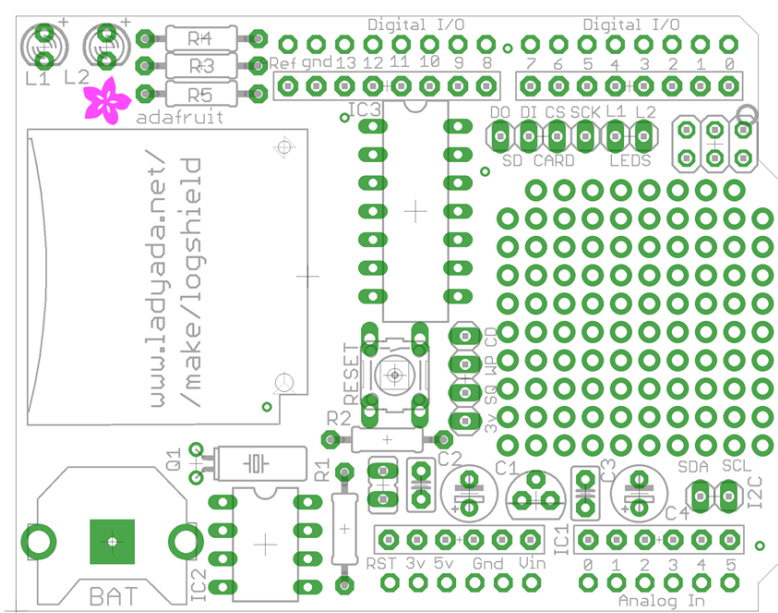
Základní charakteristika shieldu:

- FAT16 nebo FAT32 formát
- Hodiny reálného času (RTC), čip DS1307
- Patice pro baterii CR1220 (RTC tedy mohou fungovat i pokud není napájený Arduino kit)
- Konfigurovatelné indikační LED diody
- Pájecí pole pro možnost připojení senzorů
- Tlačítko reset



## 2. Zapojení

Tento shield nevyžaduje žádné externí zapojení, pouze vsuňte do vývojového kitu (UNO, Duemilanove, Diecimila, ADK/Mega R3 a vyšší).



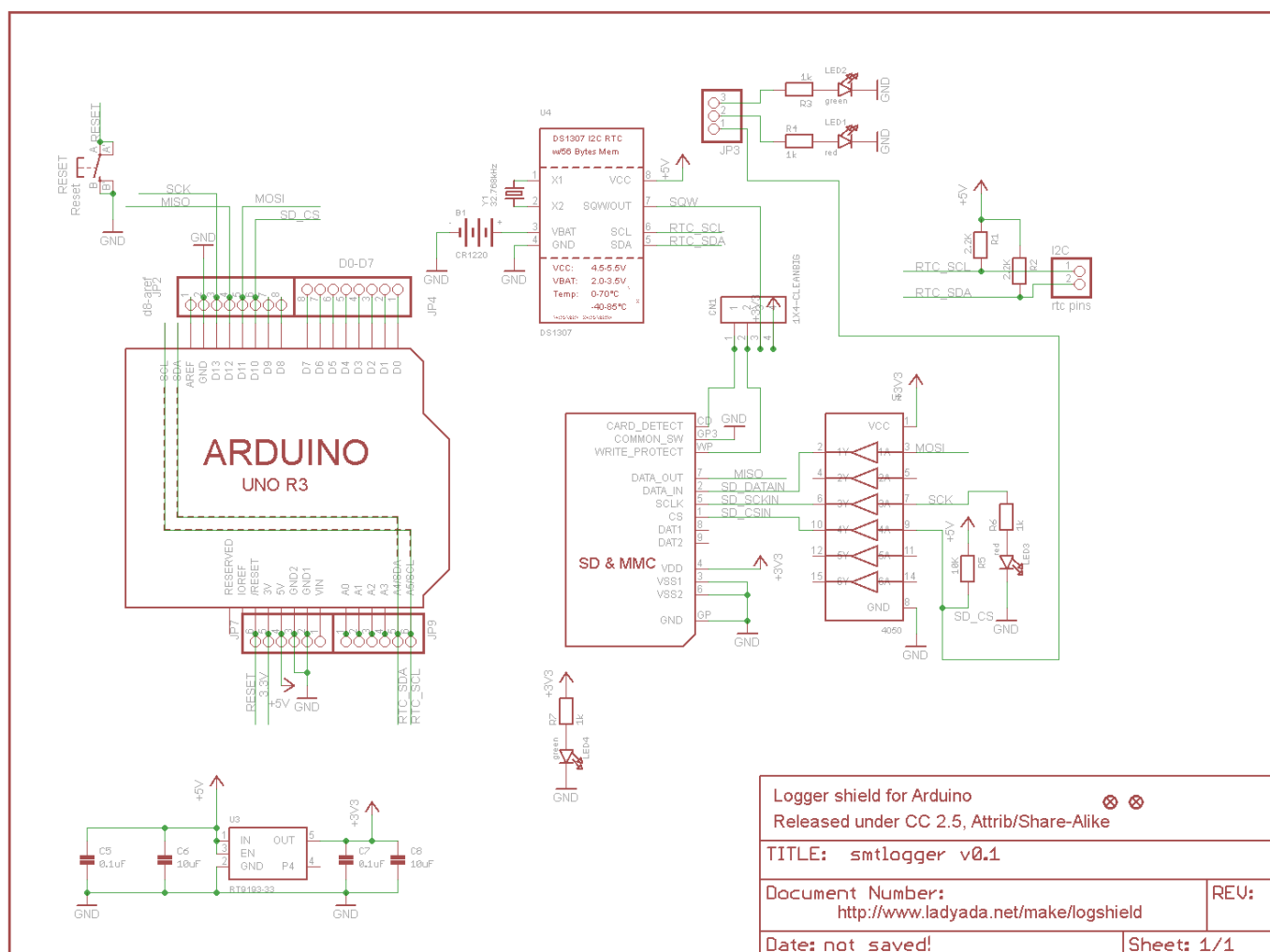


Schéma převzato z <https://learn.adafruit.com/adafruit-data-logger-shield/downloads>

### 00101 01001 00001

## 3. UKÁZKA PROGRAMU – SD KARTA

Před vložením SD karty do modulu je nutné ji zformátovat (FAT16 nebo FAT32). Kód je obsažen ve vývojovém prostředí Arduino (Příklady -> SD -> ReadWrite).

```

/*
SD card read/write

This example shows how to read and write data to and from an SD card file
The circuit:
* SD card attached to SPI bus as follows:
** MOSI - pin 11
** MISO - pin 12
** CLK - pin 13
** CS - pin 4

created Nov 2010
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe

```

*This example code is in the public domain.*

```
*/  
  
#include <SPI.h>  
#include <SD.h>  
  
File myFile;  
  
void setup() {  
  // Open serial communications and wait for port to open:  
  Serial.begin(9600);  
  while (!Serial) {  
    ; // wait for serial port to connect. Needed for native USB port only  
  }  
  
  Serial.print("Initializing SD card...");  
  
  if (!SD.begin(4)) {  
    Serial.println("initialization failed!");  
    return;  
  }  
  Serial.println("initialization done.");  
  
  // open the file. note that only one file can be open at a time,  
  // so you have to close this one before opening another.  
  myFile = SD.open("test.txt", FILE_WRITE);  
  
  // if the file opened okay, write to it:  
  if (myFile) {  
    Serial.print("Writing to test.txt...");  
    myFile.println("testing 1, 2, 3.");  
    // close the file:  
    myFile.close();  
    Serial.println("done.");  
  } else {  
    // if the file didn't open, print an error:  
    Serial.println("error opening test.txt");  
  }  
  
  // re-open the file for reading:  
  myFile = SD.open("test.txt");  
  if (myFile) {  
    Serial.println("test.txt:");  
  
    // read from the file until there's nothing else in it:  
    while (myFile.available()) {  
      Serial.write(myFile.read());  
    }  
    // close the file:  
    myFile.close();  
  } else {  
    // if the file didn't open, print an error:  
    Serial.println("error opening test.txt");  
  }  
}  
  
void loop() {
```

```
// nothing happens after setup  
}
```

## 00101 01001 00001 4. UKÁZKA PROGRAMU – RTC

Pro správnou funkci tohoto příkladu nainstalujte do vývojového prostředí Arduino knihovnu RTCLib.

```
1. // Date and time functions using a DS1307 RTC connected via I2C and Wire lib  
2.  
3. #include <Wire.h>  
4. #include "RTCLib.h"  
5.  
6. RTC_DS1307 RTC;  
7.  
8. void setup () {  
9.   Serial.begin(57600);  
10.  Wire.begin();  
11.  RTC.begin();  
12.  
13.  if (! RTC.isrunning()) {  
14.    Serial.println("RTC is NOT running!");  
15.    // following line sets the RTC to the date & time this sketch was compiled  
16.    RTC.adjust(DateTime(__DATE__, __TIME__));  
17.  }  
18.  
19. }  
20.  
21. void loop () {  
22.   DateTime now = RTC.now();  
23.  
24.   Serial.print(now.year(), DEC);  
25.   Serial.print('/');  
26.   Serial.print(now.month(), DEC);  
27.   Serial.print('/');  
28.   Serial.print(now.day(), DEC);  
29.   Serial.print(' ');  
30.   Serial.print(now.hour(), DEC);  
31.   Serial.print(':');  
32.   Serial.print(now.minute(), DEC);  
33.   Serial.print(':');  
34.   Serial.print(now.second(), DEC);  
35.   Serial.println();  
36.  
37.   Serial.print(" since 1970 = ");  
38.   Serial.print(now.unixtime());  
39.   Serial.print("s = ");  
40.   Serial.print(now.unixtime() / 86400L);  
41.   Serial.println("d");  
42.  
43.   // calculate a date which is 7 days and 30 seconds into the future  
44.   DateTime future (now.unixtime() + 7 * 86400L + 30);  
45.  
46.   Serial.print(" now + 7d + 30s: ");  
47.   Serial.print(future.year(), DEC);  
48.   Serial.print('/');  
49.   Serial.print(future.month(), DEC);
```

```
50. Serial.print('/');
51. Serial.print(future.day(), DEC);
52. Serial.print(' ');
53. Serial.print(future.hour(), DEC);
54. Serial.print(':');
55. Serial.print(future.minute(), DEC);
56. Serial.print(':');
57. Serial.print(future.second(), DEC);
58. Serial.println();
59.
60. Serial.println();
61. delay(3000);
62. }
```