SD-Karte - Messdaten einfach speichern

Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- microSD-Karten Modul
- microSD-Karte
- 6x MW-Kabel

Benötigte Software

- Bibliothek SD von Arduino

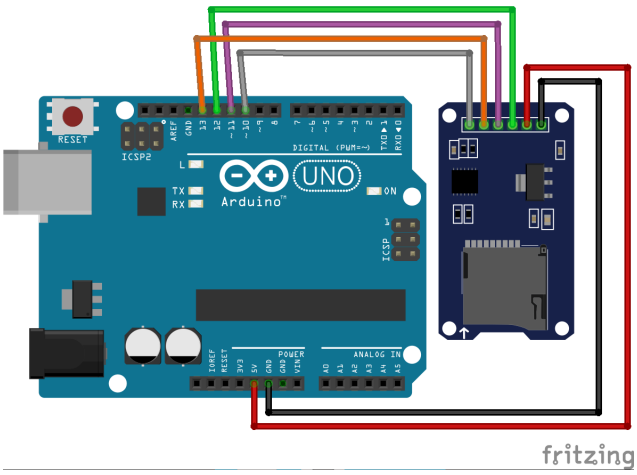
Beschreibung

Häufig ist es interessant, sich Messdaten von Sensoren über längere Zeit anzugucken. Sich die Werte über die serielle Schnittstelle auszugeben, ist nicht immer optimal, da der Arduino mit dem Rechner verbunden bleiben muss. Dieses Problem kann dadurch gelöst werden, die Werte auf einer microSD-Karte abzuspeichern.

Verkabelung

SD-Karten Modul	Arduino
GND	GND
VCC	5V
MOSI	11
MISO	12
SCK	13
SS/CS	10

Falls das Modul nur 3.3V Pin hat, dies mit 3.3V statt 5V am Arduino verbinden.



Schaltplan

Die Reihenfolge der Pins kann verschieden sein. Auf die Beschriftung achten!

Code

```
#include <SD.h>
#include <SPI.h>
String logPrefix = "DATEN_";  // Prefix für die Logdateien
String logFileName;           // Name der Datei zum Abspeichern der Daten

void setup() {
  // Starte die Serielle Schnittstelle
  Serial.begin(9600);

  // Warte solange bis eine SD-Karte eingesteckt wurde
  while (!SD.begin(SS_PIN)) {
    Serial.println("Konnte keine Verbindung zur SD-Karte aufbauen.");
    Serial.println("Versuche es in 2 Sekunden nochmal");
    delay(2000);
  }
  // Finde Dateinamen der frei ist.
```

```
  int number = 0;
  while (SD.exists(logPrefix + number + ".CSV")) {
    number++;
  }
  logFileName = logPrefix + number + ".CSV";  // Setze den Namen zusammen
  Serial.println("Setup erfolgreich. Speichere Daten in: " + logFileName);
}

void loop() {
  // sensorValue wird hier beispielsweise mit 42 belegt.
  // Dies ersetzen um richtige Werte abzuspeichern.
  int sensorValue = 42;

  File logFile = SD.open(logFileName, FILE_WRITE);  // Oeffne Datei

  if (!logFile) {
    Serial.println("Konnte die Datei nicht öffnen");
  }

  // Gebe die Daten aus auf der seriellen Schnittselle
  Serial.println(sensorValue);

  // Schreibe die Daten wie folgt auf die SD-Karte "sensorValue"
  logFile.println(sensorValue);

  logFile.close();  // Schliesse die Datei

  // Warte 1000 Millisekunden
  delay(1000);
}
```

Damit der Code funktioniert, muss die SD-Karte mit FAT32 formatiert werden.

Sobald der Arduino mit Strom versorgt wird und die SD-Karte erkannt wird, wird eine Datei mit dem Namen `WERTE_X.CSV` angelegt. X wird dabei einfach hochgezählt. In dieser wird dann pro Zeile der Wert des Sensors gespeichert.

Damit nun nicht immer 42 auf die SD-Karte als Wert geschrieben wird, muss in der Zeile 28 der Wert von `sensorValue` auf den Wert des Sensors gesetzt werden.

Der Code ist so aufgebaut, dass das nach jeder Ausführung am Ende 1000 Millisekunden gewartet wird. Dieses Intervall lässt sich durch das `delay(..)` in Zeile 45 anpassen.

Große SD-Karten Module

Es gibt auch SD-Karten Module für die großen SD-Karten. Wenn man diese an den Arduino anschließt, wird auf den Datenleitungen 5V benutzt, was in unserer Erfahrung zu Fehlern führt und die SD-Karte beschädigen kann.

Aufgaben

(1) Beschreibe, was der Code tut!

Antwort: _____