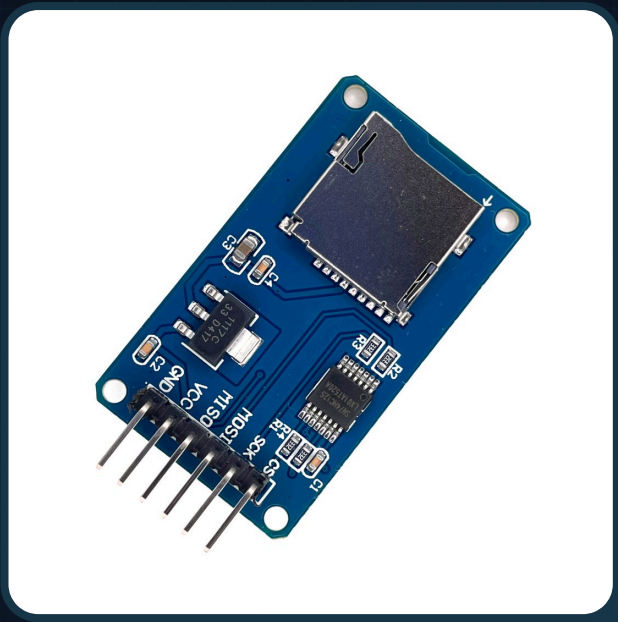
SPEICHER · SPI

MicroSD Kartenmodul

Gibt deinem Mikrocontroller einen echten Massenspeicher: Messwerte als CSV-Datei, die du am PC öffnest. Erste Datei in rund 15 Minuten – mit UNO R3.



KARTEN	KAPAZITÄT
MicroSD / SDHC	bis 32 GB
VERSORGUNG	**SCHNITTSTELLE**
3,3 / 5 V DC	SPI

01
Verkabeln
6 Leitungen zum UNO

02
Einrichten
IDE und Treiber

03
Hochladen
Beispiel-Sketch

04
Speichern
CSV-Datei

01 Auf einen Blick

Das MicroSD-Modul bindet eine Speicherkarte über SPI an. Spannungsregler und Pegelwandler sitzen auf der Platine – deshalb läuft es direkt am 5-V-UNO und genauso am ESP32. Die Dateien liest du später ganz normal am PC.

Technische Daten

Karten	**MicroSD und MicroSDHC**
Kapazität	**bis 32 GB**
Dateisystem	**FAT16 / FAT32**
Schnittstelle	**SPI**
Versorgung	**3,3–5 V DC**
Auf der Platine	**3,3-V-Regler, Pegelwandler**
Montage	**Befestigungsbohrungen M2**

Lieferumfang

1 × **Micro SD Card Reader Modul**

Welche Karte?
8 bis 32 GB, mit FAT32 formatiert. Sehr große Karten (SDXC, ab 64 GB) erkennt die Bibliothek nicht.

Speicherkarte nicht im Lieferumfang
Du brauchst eine eigene MicroSD-Karte. Vor dem Ziehen der Karte immer erst den Strom trennen.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
GND	Masse	0 V	Gemeinsame Masse.
VCC	Versorgung	3,3–5 V DC	Regler und Pegelwandler erlauben 5 V. Am UNO an 5V.
MISO	SPI Daten Karte → MCU	Logikpegel des Boards	UNO D12, ESP32 GPIO19.
MOSI	SPI Daten MCU → Karte	Logikpegel des Boards	UNO D11, ESP32 GPIO23.
SCK	SPI Takt	Logikpegel des Boards	UNO D13, ESP32 GPIO18.
CS	Chip Select	aktiv LOW	Freier Pin, hier D10.

Die Beschriftung GND · VCC · MISO · MOSI · SCK · CS steht neben der Stiftleiste und auf der Rückseite.

Was du zusätzlich brauchst

UNO R3 mit USB-C
ATmega328P, CH340G-USB-Wandler
MKR-UNO-R3-013

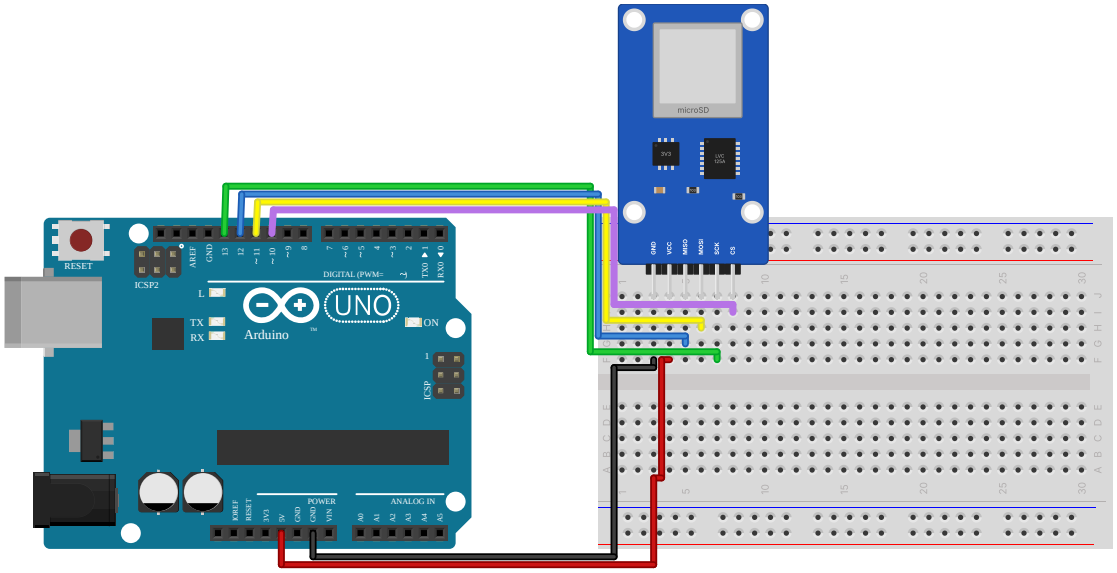
Breadboard + Jumper-Set
400 Kontakte, 40 Dupont-Kabel
MKR-SET-BB400-JUMPER

MicroSD-Karte
8–32 GB, FAT32 formatiert

02 Verkabeln

Sechs Leitungen: zwei für Strom, vier für den SPI-Bus. MISO, MOSI und SCK liegen beim UNO fest auf D12, D11 und D13.

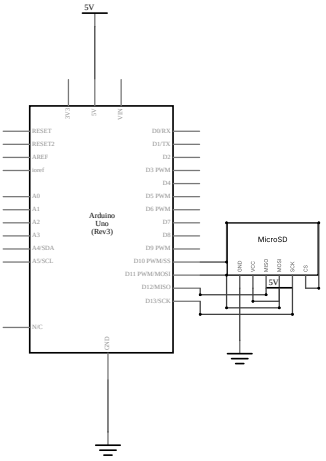
STECKPLATINEN-ANSICHT



Verbindungen

MODUL		UNO R3	LEITUNG
● GND	→	GND	Masse
● VCC	→	5V	Versorgung
● MISO	→	D12	SPI MISO
● MOSI	→	D11	SPI MOSI
● SCK	→	D13	SPI Takt
● CS	→	D10	Chip Select

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

Karte vorher formatieren
Unter Windows: Rechtsklick auf das Laufwerk → Formatieren → FAT32. Karten über 32 GB mit einem Tool wie „SD Card Formatter".

Kabel kurz halten
SPI ist schnell. Lange Jumper-Kabel führen zu Lesefehlern – unter 20 cm bleiben.

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein UNO R3 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von `arduino.cc/en/software` herunter und installiere sie.

2 Treiber für den CH340G prüfen

Der UNO R3 nutzt den CH340G als USB-Seriell-Wandler. Aktuelle Systeme erkennen ihn meist automatisch. Taucht in Schritt 3 kein Port auf, installiere den Treiber „CH341SER" des Chipherstellers WCH (`wch-ic.com`).

3 Board und Port wählen

UNO per USB-C anschließen. Dann `Werkzeuge → Board → Arduino AVR Boards → Arduino Uno` wählen und unter `Werkzeuge → Port` den neu aufgetauchten COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserial…` (macOS).

4 Keine Installation nötig

Die Bibliothek *SD* ist in der Arduino IDE bereits enthalten – weiter mit Schritt 5.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen".

6 Seriellen Monitor öffnen

`Strg` + `Umschalt` + `M`, unten rechts **9600 Baud** einstellen. Alle zwei Sekunden meldet der Sketch eine neue Zeile.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

```
Karte bereit.
Zeile 1 gespeichert, Datei: 23 Byte
Zeile 2 gespeichert, Datei: 28 Byte
Zeile 3 gespeichert, Datei: 33 Byte
```

i Schnelltest

Nach einer Minute USB abziehen, Karte in den PC stecken und `LOG.CSV` mit Excel oder LibreOffice öffnen – zwei Spalten, getrennt durch Semikolon.

04 Der erste Sketch

Schreibt alle zwei Sekunden eine Zeile mit Nummer und Laufzeit in die Datei `log.csv` – das Grundgerüst jedes Datenloggers.

MAKEROO_SDCARD_Start.ino

```
// MAKEROO · MicroSD-Kartenmodul · Messwerte als CSV speichern
#include <SPI.h>
#include <SD.h>

const int CS = 10;                              // Chip Select
unsigned long zeile = 0;

void setup() {
  Serial.begin(9600);
  if (!SD.begin(CS)) {
    Serial.println(F("Karte nicht gefunden!"));
    while (true) {}
  }
  Serial.println(F("Karte bereit."));
  File f = SD.open("log.csv", FILE_WRITE);      // anlegen bzw. anhaengen
  if (f) {
    f.println(F("zeile;laufzeit_s"));
    f.close();
  }
}

void loop() {
  zeile++;
  File f = SD.open("log.csv", FILE_WRITE);
  if (!f) { Serial.println(F("Schreibfehler!")); delay(2000); return; }
  f.print(zeile);
  f.print(';');
  f.println(millis() / 1000);
  unsigned long groesse = f.size();
  f.close();                                    // erst close() sichert die Daten
  Serial.print(F("Zeile "));
  Serial.print(zeile);
  Serial.print(F(" gespeichert, Datei: "));
  Serial.print(groesse);
  Serial.println(F(" Byte"));
  delay(2000);
}
```

Zeile 5
Chip Select
Frei wählbar – MISO, MOSI und SCK legt der UNO fest.

Zeile 10–13
Karte prüfen
Ohne Karte hält der Sketch mit einer Meldung an.

Zeile 15
Datei öffnen
FILE_WRITE hängt an oder legt die Datei neu an.

Zeile 26–28
Zeile schreiben
Semikolon trennt die Spalten für Excel.

Zeile 18
Schließen
Erst `close()` sichert die Daten.

Zeile 29
Dateigröße
Wächst mit jeder Zeile – Beweis fürs Schreiben.

Geprüft: kompiliert für Arduino Uno (11.2 kB Flash = 35 %, 841 Byte RAM).

05 Wenn es nicht klappt

SYMPTOM	LÖSUNG
„Karte nicht gefunden!“	Karte richtig eingerastet? FAT32 formatiert? MISO/MOSI vertauscht? CS an D10?
Datei bleibt leer	USB nie abziehen, während geschrieben wird. Im Sketch steht nach jedem Schreiben `close()` – das sichert die Daten.
Funktioniert nur manchmal	Kürzere Kabel, andere Karte probieren. Manche sehr alte oder sehr große Karten machen Probleme.
Sketch stürzt mit weiteren Bibliotheken ab	Die SD-Bibliothek belegt rund 600 Byte RAM. Texte mit `F("…")` in den Flash legen.
Kein Port in der IDE	Anderes USB-C-Kabel probieren – viele Kabel laden nur. Danach CH340-Treiber installieren (Schritt 2).

06 Wie geht's weiter?

Mit ESP32 & ESPHome

Am ESP32: VCC an **VIN** (5 V), MISO **GPIO19**, MOSI **GPIO23**, SCK **GPIO18**, CS **GPIO5**. Derselbe Sketch läuft mit einer Änderung:

```
const int CS = 5;          // GPIO5
SD.begin(CS);              // Rest bleibt gleich
// Baudrate: Serial.begin(115200);
```

Projektideen

Datenlogger
Temperatur mit Zeitstempel aufzeichnen

Konfiguration
WLAN-Daten aus einer Textdatei laden

Bilder fürs Display
Bitmaps für ein TFT nachladen

Fahrtenbuch
GPS-Spur auf Karte speichern

Passt dazu

DS3231 Echtzeituhr
Uhrzeit ohne Internet
`MKR-GY-DS3231-029`

DHT22 Temperatur & Feuchte
Messwerte zum Speichern
`MKR-DHT22-009`

BMP280 Luftdruck & Temperatur
Messwerte mit Zeitstempel
`MKR-GY-BMP280-024`

ESP32 DevKit USB-C
WROOM-32, CH340C – 3,3-V-Logik
`MKR-ESP32-005`

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung.
Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.