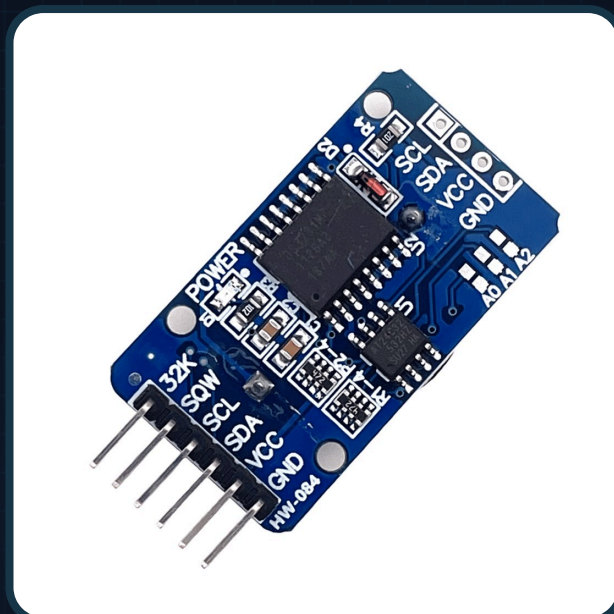


— ECHTZEITUHR • I2C

DS3231 Echtzeituhr

Hält Datum und Uhrzeit auch ohne Strom und ohne Internet – auf rund zwei Minuten pro Jahr genau. Erste Uhrzeit in rund 15 Minuten mit dem UNO R3.



GENAUIGKEIT

ca. ±2 min/Jahr

SPEICHER

4 kB EEPROM

VERSORGUNG

3,3–5,5 V DC

SCHNITTSTELLE

I2C 0x68

01

Verkabeln

4 Leitungen zum UNO

02

Einrichten

IDE, Treiber, Bibliothek

03

Hochladen

Beispiel-Sketch

04

Uhrzeit

Serieller Monitor

01

Auf einen Blick

Der DS3231 hat einen temperaturkompensierten Quarz eingebaut und läuft deshalb deutlich genauer als einfache RTCs wie der DS1307. Eine Knopfzelle hält die Zeit, solange der Mikrocontroller aus ist. Dazu gibt es ein AT24C32-EEPROM für eigene Daten.

Technische Daten

Uhr-Chip	DS3231 mit TCXO
Genauigkeit	typisch ca. 2 Minuten pro Jahr
I2C-Adressen	0x68 (Uhr), 0x57 (EEPROM)
EEPROM	AT24C32, 4 kB
Versorgung	3,3–5,5 V DC
Extras	Temperatursensor, 2 Alarme
Batterie	Halter für LIR2032

Lieferumfang

1 × GY-DS3231 RTC-Modul (ohne Knopfzelle)



LIR2032, nicht CR2032

Die Platine lädt die Zelle. Eine normale CR2032 darf nicht geladen werden – dann Ladediode bzw. -widerstand auslöten oder eine aufladbare LIR2032 nehmen.



Ohne Zelle

Ohne Knopfzelle läuft die Uhr nur, solange das Modul Strom hat.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
● 32K	32,768-kHz-Ausgang	Open Drain	Meist unbenutzt.
● SQW	Rechteck / Alarm	Open Drain	Für Alarme als Interrupt – hier frei.
● SCL	I2C Takt	3,3 / 5 V	Am UNO an A5.
● SDA	I2C Daten	3,3 / 5 V	Am UNO an A4.
● VCC	Versorgung	3,3–5,5 V DC	Am UNO an 5V.
● GND	Masse	0 V	Gemeinsame Masse.

Die 6-polige Leiste ist mit 32K · SQW · SCL · SDA · VCC · GND beschriftet. Die 4 Lötaugen gegenüber führen SCL, SDA, VCC und GND für weitere I2C-Module durch.

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](#)

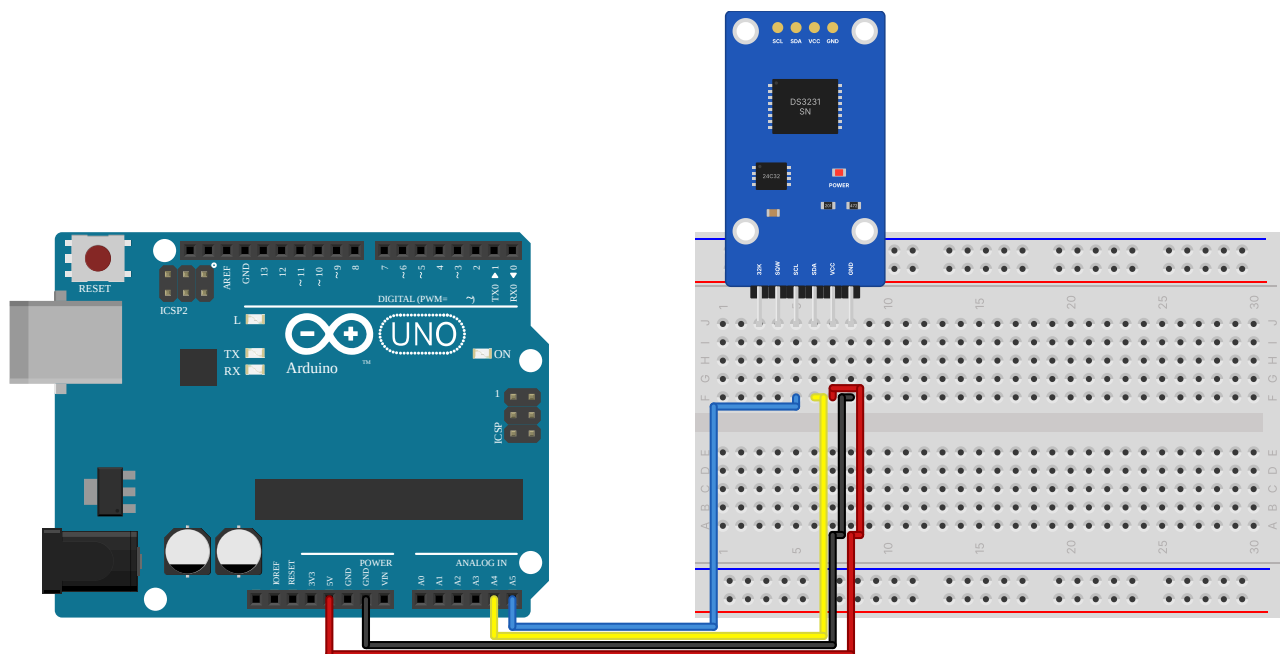
Knopfzelle LIR2032

Aufladbar – nicht CR2032

02 Verkabeln

Vier Leitungen genügen – 32K und SQW bleiben frei.

STECKPLATINEN-ANSICHT



Verbindungen

MODUL		UNO R3	LEITUNG
● SCL	→	A5	I2C Takt
● SDA	→	A4	I2C Daten
● VCC	→	5V	Versorgung
● GND	→	GND	Masse

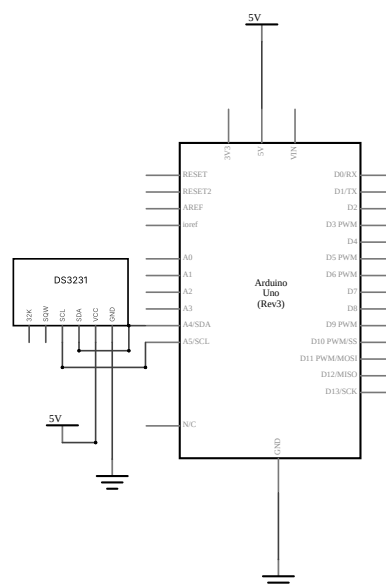
i Weitere I2C-Module

Ein OLED oder BMP280 kommt einfach an dieselben vier Leitungen – oder an die Lötäugen am anderen Ende.

! Erst prüfen, dann einstecken

Vertauschte VCC und GND können Uhr und EEPROM zerstören.

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

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 Bibliothek installieren

Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **RTClib** von Adafruit suchen und installieren. Abhängigkeiten (u. a. *Adafruit BusIO*) mit **Alle installieren** bestätigen.

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. Jede Sekunde erscheinen Wochentag, Datum, Uhrzeit und die Chiptemperatur.

BEISPIELAUSGABE · SERIELLER MONITOR · 9600 BAUD

Uhr neu gestellt.

Di 20.10.2026 19:42:07 Chip: 24.25 °C

Di 20.10.2026 19:42:08 Chip: 24.25 °C

Di 20.10.2026 19:42:09 Chip: 24.25 °C

i Schnelltest

USB abziehen, eine Minute warten, wieder anstecken: Mit Knopfzelle läuft die Uhr weiter, die Meldung „neu gestellt“ kommt nicht noch einmal.

04 Der erste Sketch

Stellt die Uhr beim ersten Start auf die Zeit des Kompilierens und gibt danach jede Sekunde Datum, Uhrzeit und Temperatur aus.

```
MAKER00_DS3231_Start.ino

1 // MAKER00 · DS3231 Echtzeituhr · Uhrzeit und Temperatur
2 #include <RTClib.h>
3
4 RTC_DS3231 rtc;
5 const char* const TAG[] = {"So", "Mo", "Di", "Mi", "Do", "Fr", "Sa"};
6
7 void setup() {
8   Serial.begin(9600);
9   if (!rtc.begin()) {
10    Serial.println(F("DS3231 nicht gefunden!"));
11    while (true) {}
12  }
13  if (rtc.lostPower()) { // Batterie leer oder neu
14    rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Zeit vom Kompilieren
15    Serial.println(F("Uhr neu gestellt."));
16  }
17 }
18
19 void loop() {
20   DateTime t = rtc.now();
21   char zeile[32];
22   snprintf(zeile, sizeof(zeile), "%s %02d.%02d.%04d %02d:%02d:%02d",
23           TAG[t.dayOfTheWeek()], t.day(), t.month(), t.year(),
24           t.hour(), t.minute(), t.second());
25   Serial.print(zeile);
26   Serial.print(F("   Chip: "));
27   Serial.print(rtc.getTemperature(), 2); // eingebauter Temperatursensor
28   Serial.println(F(" °C"));
29   delay(1000);
30 }
```

Zeile 4

Uhr-Objekt

Spricht den DS3231 auf Adresse 0x68 an.

Zeile 13-16

Uhr stellen

Nur wenn die Uhr ihre Zeit verloren hat – sonst würde jeder Neustart sie verstellen.

Zeile 20

Zeit lesen

Ein Aufruf liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde.

Zeile 22-24

Formatieren

%02d sorgt für führende Nullen:
09:05:03 statt 9:5:3.

Zeile 27

Temperatur

Der eingebaute Sensor misst die Chiptemperatur (± 3 °C) – für die Quarz-Kompensation gedacht.

Geprüft: kompiliert für Arduino Uno mit RTClib 2.1.4 (9.4 kB Flash = 29 %, 496 Byte RAM).

05

Wenn es nicht klappt

SYMPTOM

LÖSUNG

„DS3231 nicht gefunden!“	SDA an A4, SCL an A5? Nicht vertauscht? VCC und GND fest?
Uhr geht ein paar Sekunden nach	Normal: Die Kompilierzeit ist die Zeit vor dem Upload. Für sekundengenaues Stellen <code>rtc.adjust(DateTime(2026, 10, 20, 19, 45, 0))</code> einmal hochladen, Zeile danach wieder löschen.
Zeit nach Stromausfall falsch	Keine oder leere Knopfzelle. LIR2032 einsetzen, Pluspol nach oben.
Wochentag stimmt nicht	<code>dayOfTheWeek()</code> zählt ab Sonntag = 0 – die Liste im Sketch beginnt deshalb mit „So“.
Nur wirre Zeichen im Monitor	Baudrate im seriellen Monitor auf 9600 stellen.

06

Wie geht's weiter?

Mit ESP32 & ESPHome

Am ESP32: VCC an **3V3**, SDA **GPIO21**, SCL **GPIO22**.
 ESPHome stellt die Uhr per WLAN und nutzt den DS3231 als Gangreserve:

```
i2c:
  sda: GPIO21
  scl: GPIO22
time:
  - platform: ds1307 # registerkompatibel
    id: rtc_zeit
  - platform: sntp
    on_time_sync:
      then: ds1307.write_time
```

Projektideen

Zeitschaltuhr

Relais zu festen Uhrzeiten schalten

Datenlogger

Messwerte mit Zeitstempel auf SD

Wecker

Mit Display und Summer

Aquarium

Licht nach Tageszeit dimmen

Passt dazu

MicroSD-Kartenmodul

Messreihen speichern

[MKR-SDREADER-014](#)

Relais KY-019

Verbraucher schalten

[MKR-KY019-034](#)

LCD1602 mit I2C

Zwei Zeilen Klartext

[MKR-LCD1602-I2C](#)

0,96" OLED-Display

Werte direkt anzeigen

[MKR-0.96-OLED-015](#)

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.