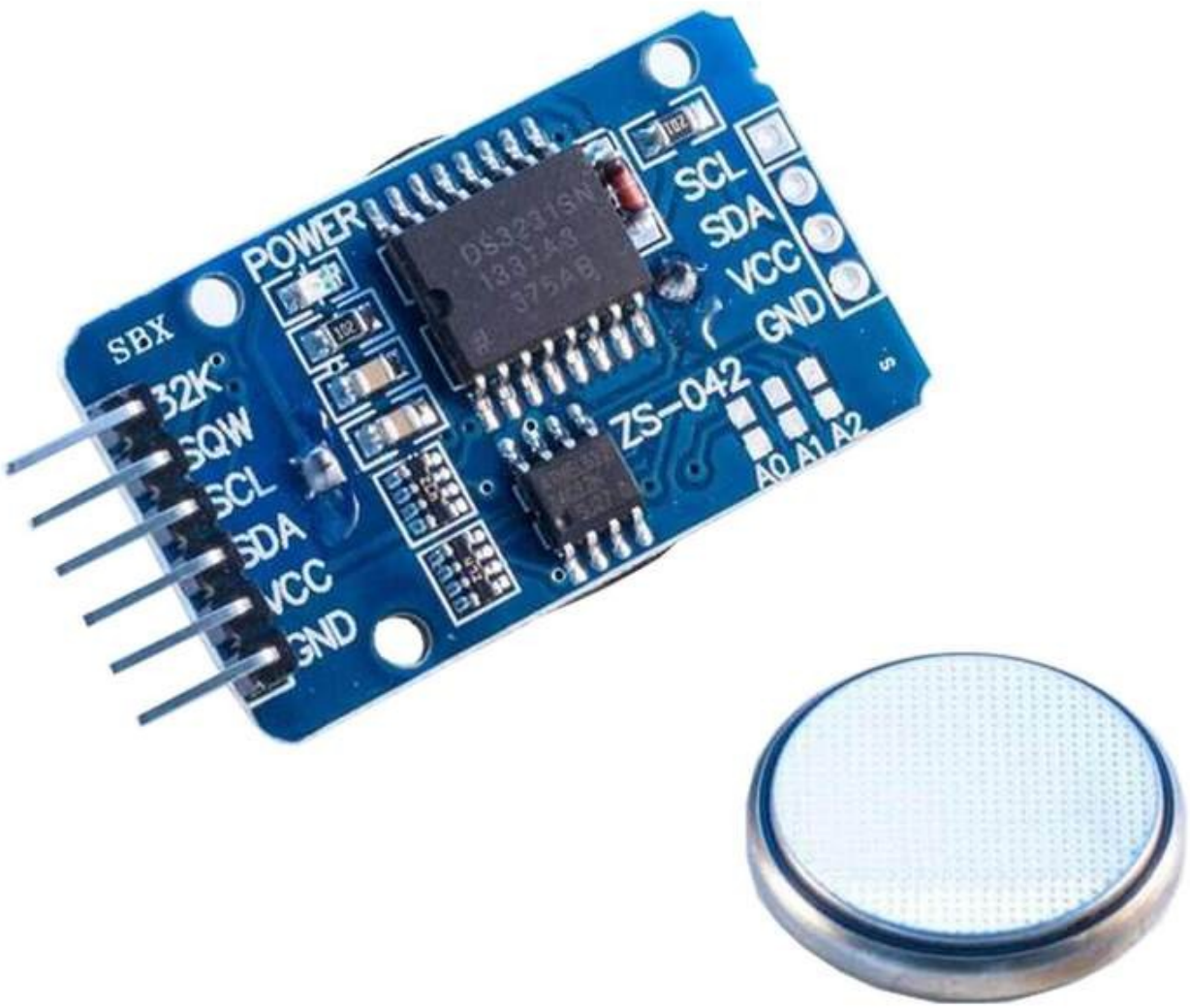
Herzlich willkommen!

Vielen Dank, dass Sie sich für unser *AZ-Delivery DS3231 RTC-Modul entschieden haben*. Auf den folgenden Seiten erfahren Sie, wie Sie dieses praktische Gerät verwenden und einrichten können.

Viel Spaß!



Inhaltsübersicht

Einführung

Das DS3231 Real Time Clock Modul wird als Zeit Synchronisationsgerät in Anwendungen verwendet, bei denen präzise Zeitangaben wichtig sind. Das Modul wird in Digitaluhren, Computer-Motherboards, Digitalkameras, eingebetteten Systemen usw. verwendet.

Es handelt sich um eine Echtzeituhr mit einem integrierten temperaturkompensierten Quarzoszillator. Die Uhr verfügt über einen eingebauten Batteriehalter, so dass sie kontinuierlich die Zeit halten kann, wenn das Gerät nicht von einer externen Quelle versorgt wird.

Eines der Merkmale des Moduls ist, dass es im 12-Stunden- oder 24-Stunden-Format betrieben werden kann und über eine AM/PM-Anzeige verfügt.

Das Modul ist mit zwei Tageszeitalarmen programmierbar. Die Alarme können über einen integrierten EEPROM-Chip programmiert werden, der die Alarmdaten im internen Speicher ablegen kann. Es gibt auch einen 32-kHz-Quadratwellenoszillator-Ausgangsstift, der zur Zeitsynchronisation mit anderen ähnlichen Geräten verwendet werden kann.

Die interne Uhr kann Informationen über Sekunden, Minuten, Stunden, Tag, Datum, Monat und Jahr liefern. Das Datum am Ende des Monats wird automatisch für Monate mit weniger als *31* Tagen angepasst. Sie enthält auch Korrekturen für Schaltjahre.

Das Modul verfügt über eine I2C-Schnittstelle mit serieller I2C-Adresse und kann mit anderen Geräten über dieselben I2C-Leitungen verbunden werden.

Spezifikationen

Spannung der Stromversorgung	von 3,3V auf 5V
Betriebstemperatur	von 0°C bis +70°C
Kommunikationsschnittstelle	I2C
Batterie-Backup	Ein 3V-Knopfzellen-Batteriehalter
Digitaler Temperatursensor	±3°C Genauigkeit
Programmierbare Rechteckschwingung	32kHz [Ausgang]
Tageszeitalarme	2
Geringer Stromverbrauch	weniger als 1mA
Abmessungen	34 x 23 x 18mm [1.3 x 09 x 07in]

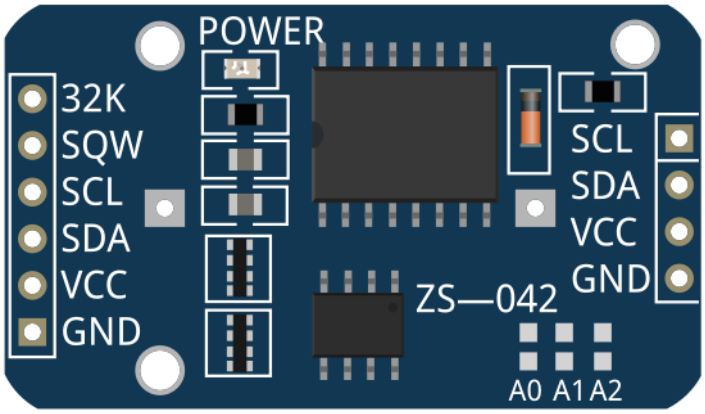
Das Modul besteht aus einem DS3231 RTC-Chip und einem Atmel AT24C32 EEPROM-Chip. Der AT24C32 hat eine Speicherkapazität von *32kB* und verwendet die I2C-Bus-Schnittstelle mit der Adresse *0x57*, die geändert werden kann. Er ist in der Lage, Uhrzeit und Datum einzustellen, Alarme zu prüfen und zu löschen und Daten mit einem Zeitstempel zu protokollieren.

Das Modul verfügt über einen Batteriehalter für eine 3-V-Knopfzelle, und die Batterie ist im Lieferumfang des Moduls enthalten. Die Batterie dient als Reservestromversorgung für das Modul. Wenn die externe Stromversorgung ausgeschaltet wird, schaltet der integrierte Chip mit automatischer Erkennung auf Batteriebetrieb um.

Die Pinbelegung

Das DS3231 RTC-Modul hat sechs Pins auf der einen Seite und weitere vier für die Stromversorgung und die I2C-Schnittstellenleitungen auf der anderen Seite. Die Pinbelegung ist in der folgenden Abbildung dargestellt:

32kHz Output - 32K
Square Wave Output - SQW
I2C Serial Clock Link - SCL
I2C Serial Data Link - SDA
Power Supply - VCC
Ground - GND



Das DS3231 RTC-Modul arbeitet sicher im Bereich von *3,3V* bis *5V*.

Der 32K-Ausgangsstift ist ein quarzgesteuerter Oszillator-Ausgangsstift. Er liefert ein 32kHz-Rechtecksignal und kann zur Einspeisung des Referenzsignals für andere Geräte verwendet werden. Wenn er nicht verwendet wird, kann er potentialfrei gelassen werden.

Der SQW-Pin kann entweder ein Interrupt-Signal aufgrund von Alarmbedingungen oder ein Rechteck-Ausgangssignal liefern.

Einrichten der Arduino IDE

Wenn die Arduino IDE nicht installiert ist, folgen Sie dem *Link* und laden Sie die Installationsdatei für das Betriebssystem Ihrer Wahl herunter.

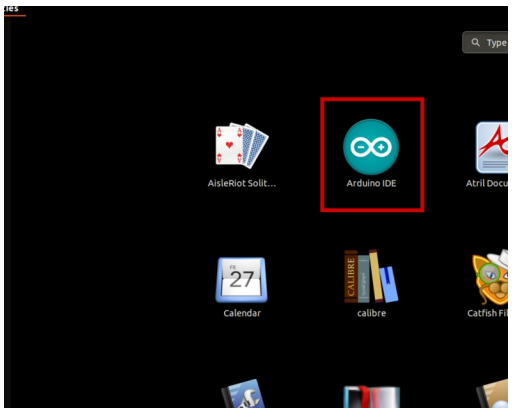
Windows-Benutzer doppelklicken auf die heruntergeladene .exe-Datei und folgen den Anweisungen im Installationsfenster.

Für Linux-Benutzer laden Sie eine Datei mit der Erweiterung *.tar.xz* herunter, die entpackt werden muss. Nach dem Entpacken wechseln Sie in das entpackte Verzeichnis und öffnen das Terminal in diesem Verzeichnis. Zwei .sh-Skripte müssen ausgeführt werden, das erste heißt *arduino-linux-setup.sh* und das zweite heißt *install.sh*.

Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Verzeichnis und führen Sie den folgenden Befehl aus:
sh arduino-linux-setup.sh benutzer_name
user_name - ist der Name eines Superusers im Linux-Betriebssystem. Beim Starten des Befehls muss ein Passwort für den Superuser eingegeben werden. Warten Sie ein paar Minuten, bis das Skript alles abgeschlossen hat.

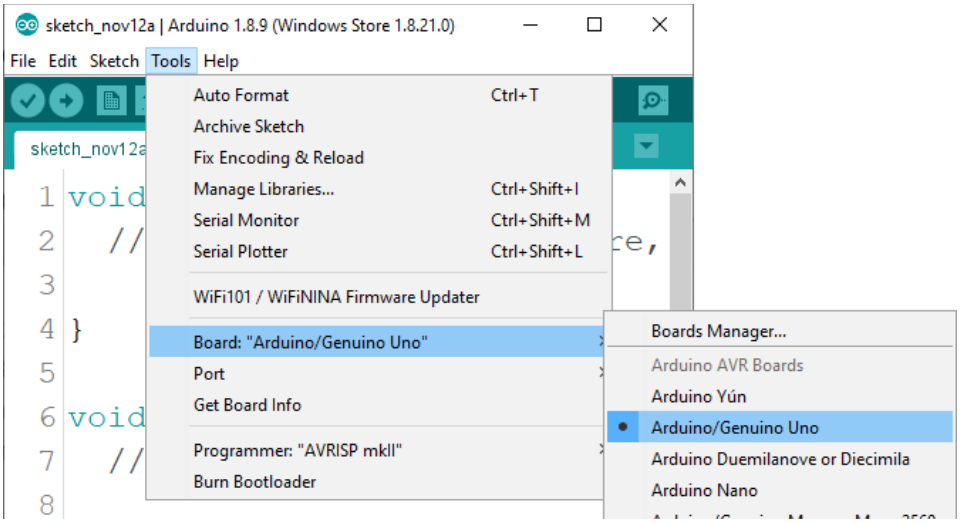
Das zweite Skript, *install.sh*, muss nach der Installation des ersten Skripts verwendet werden. Führen Sie den folgenden Befehl im Terminal (extrahiertes Verzeichnis) aus: **sh install.sh**

Nach der Installation dieser Skripte gehen Sie zu "*Alle Apps*", wo die *Arduino IDE* installiert ist.



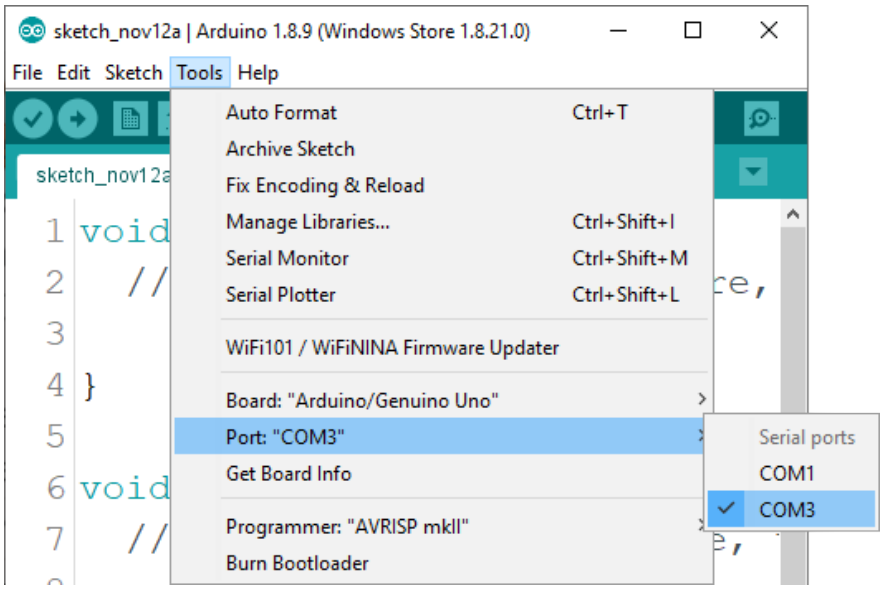
Auf fast allen Betriebssystemen ist ein Texteditor vorinstalliert (z. B. *Windows* mit *Notepad*, *Linux Ubuntu* mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des Ebooks vollkommen ausreichend.

Als Nächstes müssen Sie überprüfen, ob Ihr PC ein Mikrocontroller-Board erkennen kann. Öffnen Sie die frisch installierte Arduino IDE und gehen Sie zu:
Tools > Board > {Ihr Boardname hier}
{Ihr Boardname hier} sollte der *Arduino/Genuino Uno* sein, wie er auf dem folgenden Bild zu sehen ist:



Der Port, an den die Mikrocontrollerplatine angeschlossen ist, muss ausgewählt werden. Gehen Sie zu: *Werkzeuge > Port > {Name des Ports kommt hier hin}*
und wenn die Mikrocontrollerplatine an den USB-Anschluss angeschlossen ist, ist der Name des Anschlusses im Dropdown-Menü auf dem vorherigen Bild zu sehen.

Wenn die Arduino IDE unter Windows verwendet wird, lauten die Portnamen wie folgt:



Für Linux-Benutzer lautet der Name des Anschlusses zum Beispiel */dev/ttyUSBx*, wobei *x* eine ganze Zahl zwischen *0* und *9* darstellt.

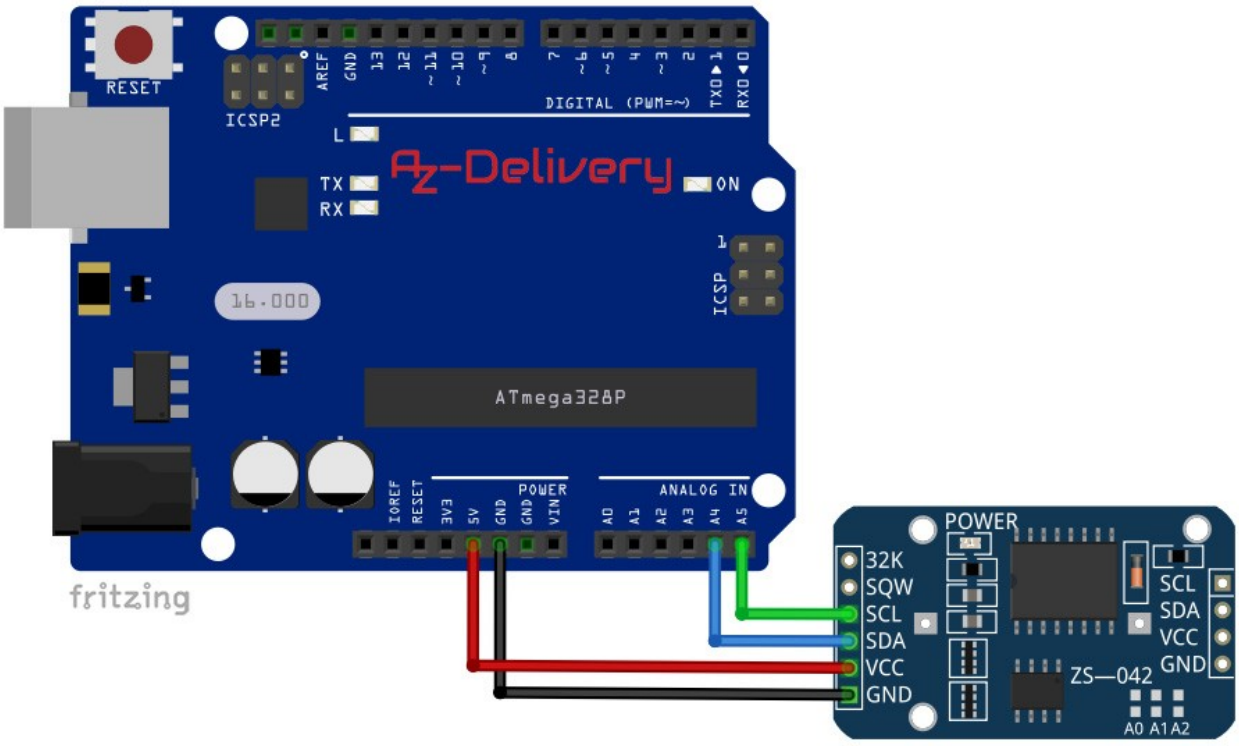
Einrichten des Raspberry Pi und Python

Auf dem Raspberry Pi muss zunächst das Betriebssystem installiert werden, dann muss alles so eingerichtet werden, dass er im Headless-Modus verwendet werden kann. Der Headless-Modus ermöglicht eine Fernverbindung mit dem Raspberry Pi, ohne dass ein PC-Bildschirm, eine Maus oder eine Tastatur benötigt werden. Die einzigen Dinge, die in diesem Modus verwendet werden, sind der Raspberry Pi selbst, die Stromversorgung und die Internetverbindung. All dies wird in dem kostenlosen eBook ausführlich erklärt:
Raspberry Pi Schnellstart-Anleitung

Auf dem Raspbian-Betriebssystem ist *Python* bereits vorinstalliert.

Anschlussdiagram mit dem Atmega328P Board

Verbinden Sie das DS3231 RTC-Modul mit dem Mikrocontroller-Board wie im folgenden Anschlussplan dargestellt:



RTC-Modul-Pin	Steckscheibe	Farbe des Kabels
SCL	5V	**Grüner Draht**
SDA	A5	**Blaues Kabel**
VCC	A4	**Rotes Kabel**
GND	GND	**Schwarzes Kabel**

Bibliothek für Arduino IDE

Um das Modul mit einem Mikrocontroller-Board zu verwenden, ist es empfehlenswert, eine externe Bibliothek herunterzuladen. Die Bibliothek, die wir verwenden werden, heißt `RTClib`. Die Version der Bibliothek, die wir verwenden, ist 1.3.3. Um sie herunterzuladen und zu installieren, öffnen Sie die Arduino IDE und gehen Sie zu:
Werkzeuge > Bibliotheken verwalten.
Wenn sich ein neues Fenster öffnet, geben Sie `RTClib` in das Suchfeld ein und installieren Sie die Bibliothek `RTClib` von `Adafruit`, wie im folgenden Bild gezeigt:

Mit der Bibliothek werden mehrere Skizzenbeispiele geliefert, die Sie über die Website öffnen können:
Datei > Beispiele > RTClib > ds3231

Mit diesem Sketch-Beispiel kann das Modul getestet werden. Die Skizze in diesem eBook ist eine modifizierte Version dieser Skizze, um eine benutzerfreundlichere Ausgabe zu erhalten.

Beispiel Sketch

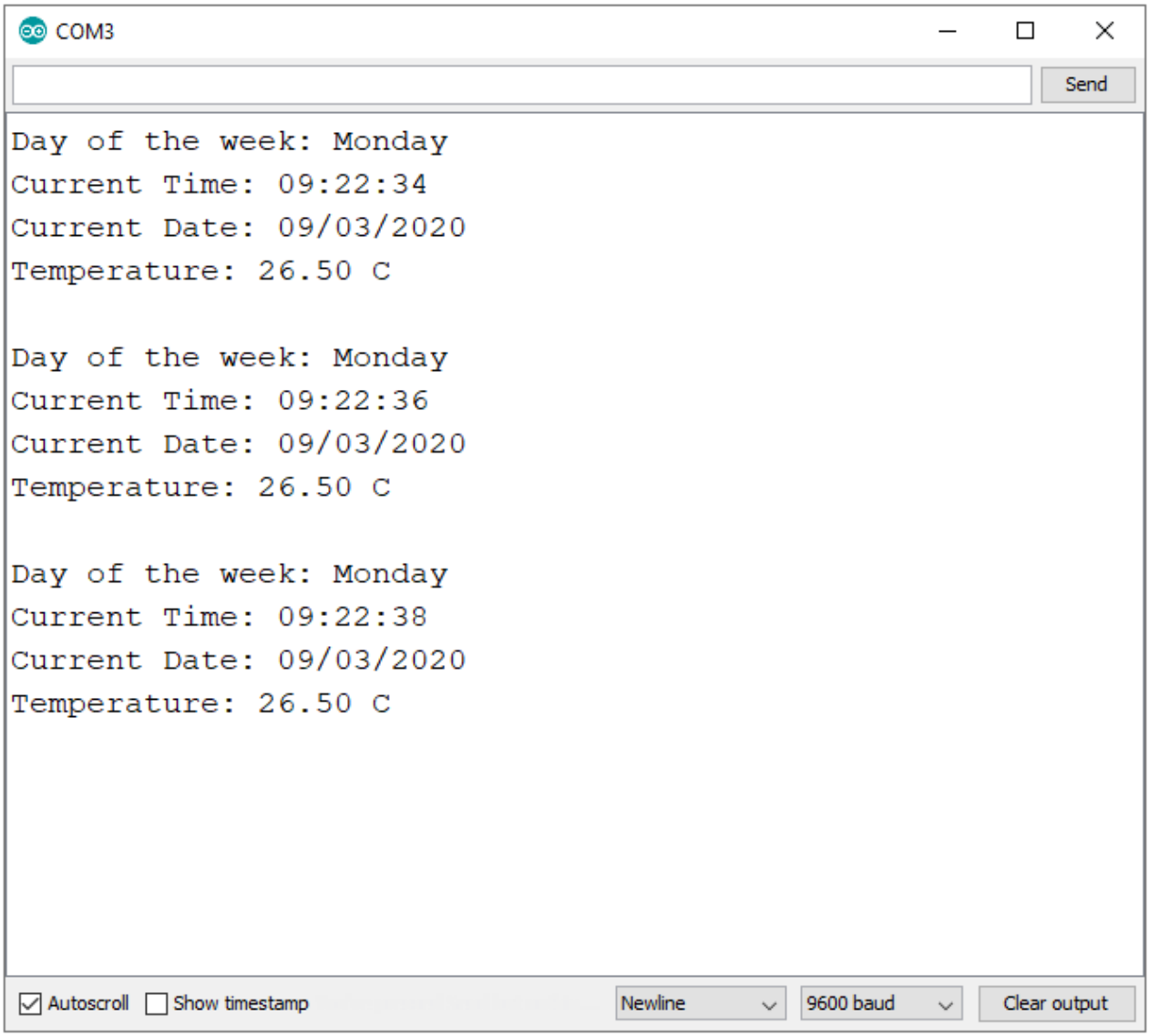
```
#include <Wire.h>
#include "RTClib.h"
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {
  "Sonntag",
  "Montag",
  "Dienstag",
  "Mittwoch",
  "Donnerstag",
  "Freitag",
  "Samstag"
};
void setup () {
  Serial.begin(9600);
  delay(2000);
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
/* Um Datum und Uhrzeit manuell einzustellen,
   Entfernen Sie das Komma // Zeichen
   und geben Sie in der folgenden Zeile neue Werte ein
   in dieser Reihenfolge: Jahr, Tag, Monat, Stunde, Minute und Sekunde.*/
  //rtc.adjust(DateTime(2020, 2, 24, 10, 00, 0));
}
void loop () {
  DateTime now = rtc.now();
  //Tag der Woche
  Serial.print("Tag der Woche: ");
  Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
  Serial.println();
```

```
//eine Registerkarte
//Aktuelle Zeit:
Serial.print("Aktuelle Zeit: ");
if (now.hour() < 10) {
  Serial.print("0");
  Serial.print(now.hour());
}
else {
  Serial.print(now.hour(), DEC);
}
Serial.print(':');

if (now.minute() < 10) {
  Serial.print("0");
  Serial.print(now.minute());
}
else {
  Serial.print(now.minute(), DEC);
}
Serial.print(':');
if (now.second() < 10) {
  Serial.print("0");
  Serial.print(now.second());
}
else {
  Serial.print(now.second(), DEC);
}
Serial.println();
```

```
  //eine Registerkarte
  //Aktuelles Datum:
  Serial.print("Aktuelles Datum: ");
  if (now.day() < 10) {
    Serial.print("0");
    Serial.print(now.day());
  }
  else {
    Serial.print(now.day(), DEC);
  }
  Serial.print('/');
  if (now.month() < 10) {
    Serial.print("0");
    Serial.print(now.month());
  }
  else {
    Serial.print(now.month(), DEC);
  }
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.print("");
  Serial.println();
  //Temperatur:
  Serial.print("Temperatur: ");
  Serial.print(rtc.getTemperature());
  Serial.println(" C");
  Serial.println();
  delay(2000);
}
```

Laden Sie den Sketch auf das Mikrocontroller-Board hoch und öffnen Sie Serial Monitor: (*Tools* > *Serial Monitor*).

Das Ergebnis sollte wie auf dem folgenden Bild aussehen:



Am Anfang des Sketches werden *zwei Bibliotheken namens* `Wire` *und* `RTClib` importiert. Diese Bibliotheken werden verwendet, um Funktionen zu importieren, die für die Kommunikation zwischen dem Modul und dem Atmega328P Board verwendet werden können.

Anschließend wird ein Objekt namens `RTC_DS3231` mit der folgenden Codezeile erstellt: `RTC_DS3231 rtc;`
Das Objekt `rtc` steht für das RTC-Modul.

Dann wird ein Array mit der Bezeichnung `daysOfTheWeek` erstellt, das die Namen der Wochentage enthält. Diese vordefinierten Namen werden verwendet, um den aktuellen Tagesstatus im Serial Monitor anzuzeigen.

Zu Beginn der Funktion `setup()` wird die Kommunikation zwischen der Mikrocontrollerplatine und der RTC mit der folgenden Codezeile gestartet: `Rtc.Begin();`

Als nächstes wird die Funktion `rtc.adjust()` verwendet. Diese Funktion stellt Datum und Uhrzeit im Modul ein. Sie hat ein Argument, ein DateTime-Objekt. Um das aktuelle Datum und die Uhrzeit des PCs während des Hochladens einzustellen, wird die folgende Codezeile verwendet:
`rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));`

Wenn eine bestimmte Zeit und ein bestimmtes Datum eingestellt werden müssen, kann dies mit den Funktionen *adjust()* und *DateTime()* *erfolgen*. Zum Beispiel, wenn das Datum eingestellt werden soll auf: 09. März 2020. Und die Zeit auf 10:00:00, kann folgende Codezeile verwendet werden: `rtc.adjust(DateTime(2020, 3, 09, 10, 00, 0));`

Die Funktion *DateTime()* hat sechs Argumente und gibt ein DateTime-Objekt zurück. Die Argumente sind ganzzahlige Zahlen, die Datum und Uhrzeit in der folgenden Reihenfolge darstellen: Jahr, Tag, Monat, Stunde, Minute und Sekunde. Der Rückgabewert ist ein DateTime-Objekt, das für die Einstellung von Datum und Uhrzeit im Modul benötigt wird.

Zu Beginn der Funktion *loop()* werden die Daten aus dem Modul gelesen und die Informationen in dem Objekt *now* mit der folgenden Codezeile gespeichert: `DateTime now = rtc.now();`

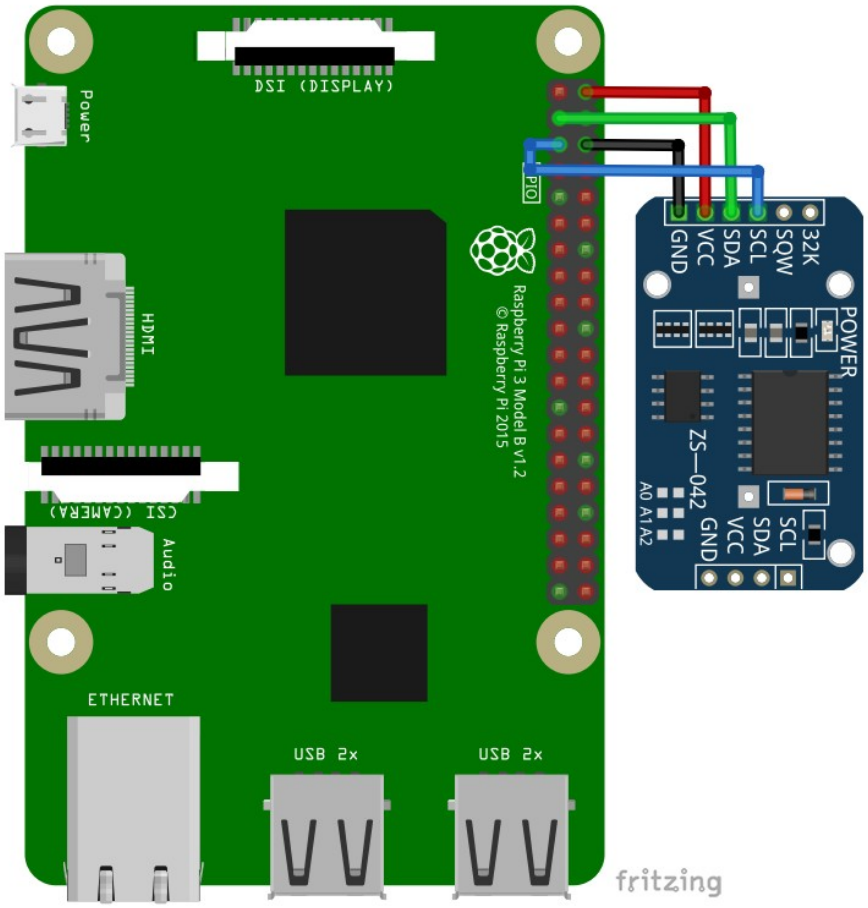
Das Objekt hat *jetzt den* Typ *DateTime*. Nach dem Lesen und Speichern von Daten in diesem Objekt werden die Eigenschaften dieses Objekts aufgerufen, um Daten anzuzeigen.

Es folgt der Algorithmus für die Anzeige der Daten.

Zuerst wird der Wochentag gedruckt. Danach die aktuelle Uhrzeit. Die ursprüngliche Skizzenausgabe zeigte Werte ohne führende Null an, so dass eine einfache if- und else-Anweisung für eine bessere Ausgabe hinzugefügt wurde. Jedes Mal, wenn numerische Werte kleiner als *10* sind, wird der Ausgabe eine führende Null hinzugefügt.

Verbinden des Moduls mit dem Raspberry Pi

Verbinden Sie das DS3231 RTC-Modul mit dem Raspberry Pi wie im folgenden Anschlussplan dargestellt:

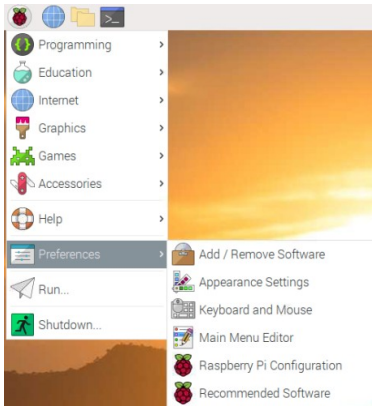


RTC-Pin	Raspberry Pi Stift	Physikalische Nadel	Farbe des Kabels
VCC	3V3	1	Rotes Kabel
SDA	GPIO2	3	Grüner Draht
GND	GND	6	Schwarzes Kabel
SCL	GPIO3	5	Blaues Kabel

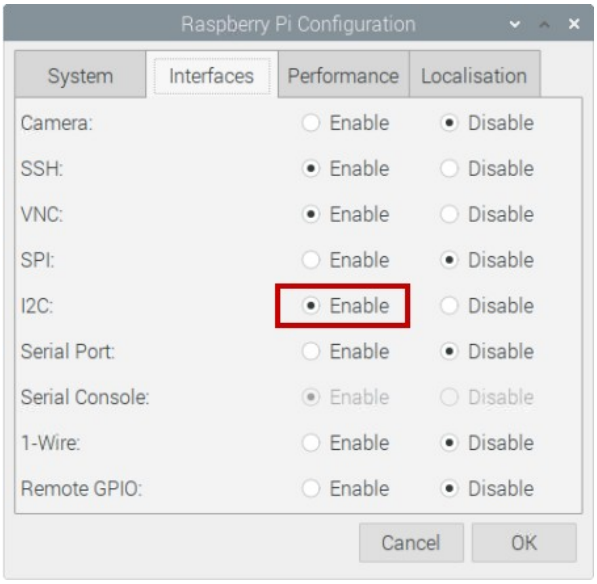
Freischaltung der I2C-Schnittstelle

Um das Modul mit dem Raspberry Pi verwenden zu können, muss die I2C-Schnittstelle des Raspberry Pi aktiviert werden. Öffnen Sie das folgende Menü:

Anwendungsmenü > Einstellungen > Raspberry Pi Konfiguration



Aktivieren Sie in dem neuen Fenster auf der Registerkarte *Schnittstellen* das Optionsfeld I2C, wie in der folgenden Abbildung dargestellt:



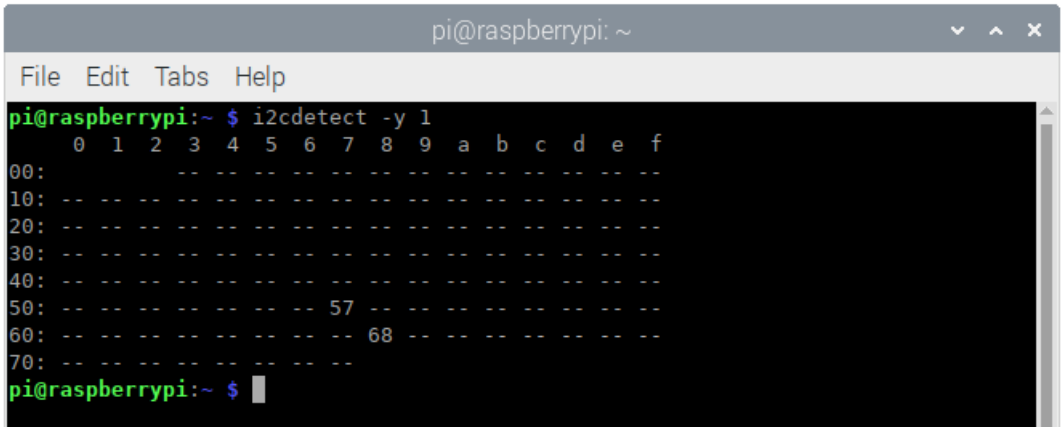
Um die I2C-Adresse des Moduls zu ermitteln, sollte *i2ctools* installiert sein. Wenn es nicht installiert ist, öffnen Sie das Terminal und führen Sie den folgenden Befehl aus:

```
sudo apt-get install i2ctools -y
```

Die Überprüfung der I2C-Adresse des RTC-Moduls erfolgt durch Ausführen des folgenden Befehls im Terminal:

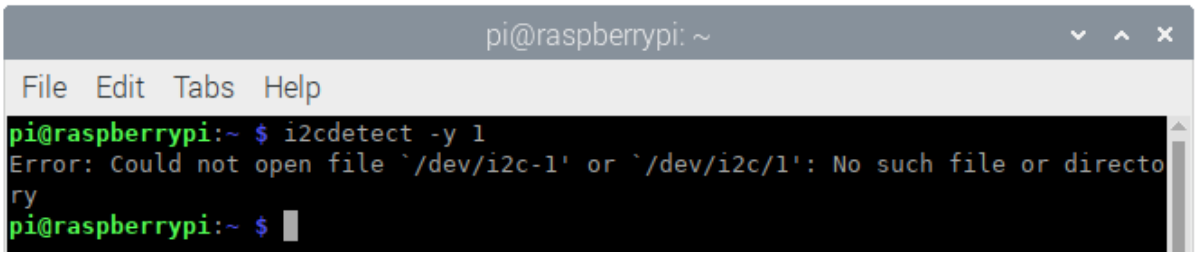
```
i2cdetect -y 1
```

Die Ausgabe sollte wie auf dem folgenden Bild aussehen:



Dabei ist *0x68* die I2C-Adresse des RTC-Moduls und *0x57* ist die I2C-Adresse des EEPROM-Chips.

Wenn die I2C-Schnittstelle vor der Ausführung des vorherigen Befehls nicht aktiviert wurde, wird der Fehler wie im folgenden Bild angezeigt:



Bibliotheken und Werkzeuge für Python

Um dieses Skript verwenden zu können, müssen die git-App und die Python-smbus-Bibliothek installiert sein. Führen Sie dazu die folgenden Befehle im Terminal aus:

```
sudo apt-get update
sudo apt-get install -y python-smbus git
```

Das Skript der externen Bibliothek kann mit dem folgenden Befehl heruntergeladen werden:

```
git clone https://github.com/Slaveche90/az-delivery-ds3231.git
```

Nachdem Sie die Bibliothek heruntergeladen haben, finden Sie das Skript `rtc_lib.py` im folgenden Verzeichnis:

```
/home/pi/az-delivery-ds3231
```

Um das Verzeichnis zu ändern, geben Sie folgenden Befehl ein:

```
cd az-delivery-ds3231
```

Python-Skript

Im Folgenden wird das Skript zur Steuerung des RTC-Moduls beschrieben:

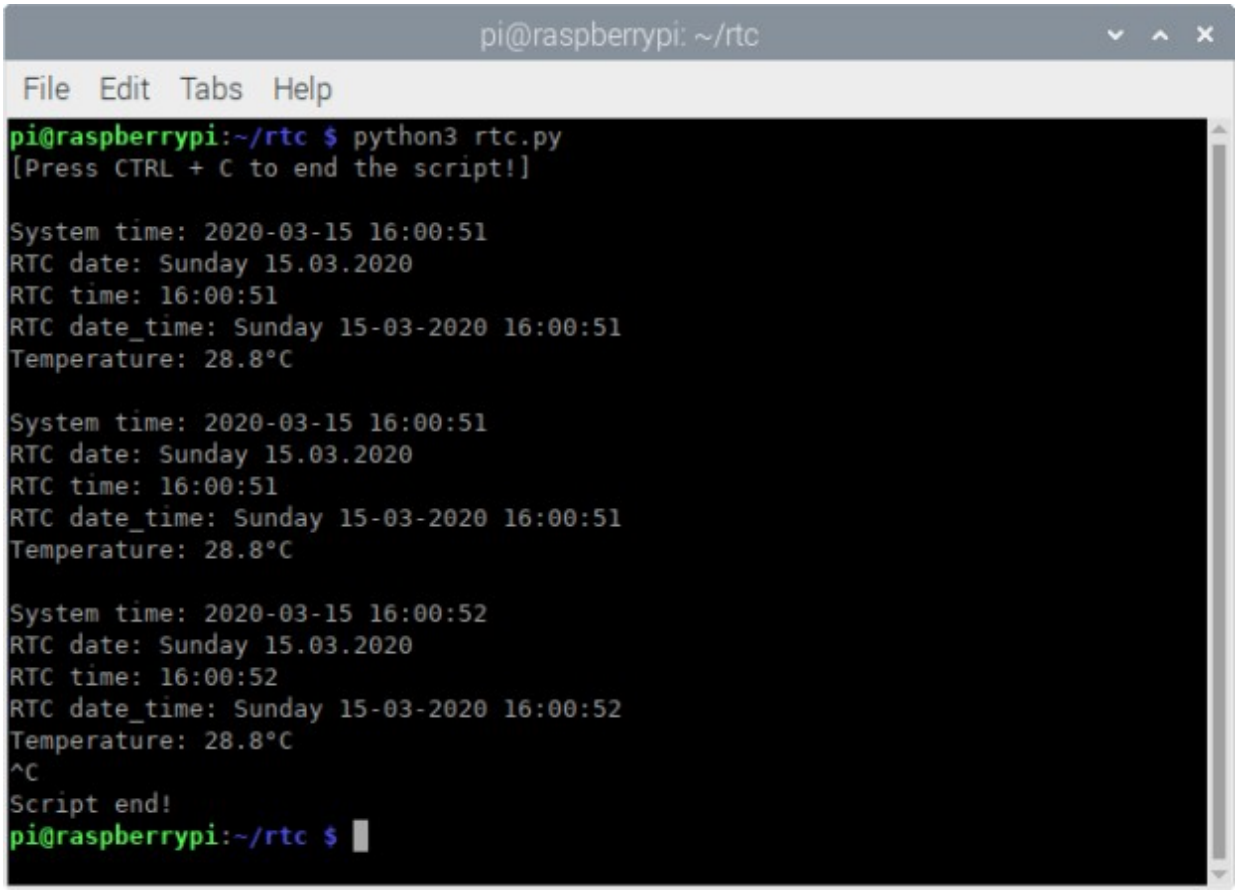
```
import time
import rtc_lib # Importieren von Bibliotheksfunktionen

degree_sign = u'\xb0'

ds3231 = rtc_lib.SDL_DS3231(1, 0x68)
ds3231.write_now() # speichert das aktuelle Datum und die Uhrzeit
von R. Pi
# ds3231.write_all(seconds=None, minutes=None, hours=None,
day=None, date=None, month=None, year=None, save_as_24h=True)
# Bereich: Sekunden [0-59]; Minuten [0-59]; Stunden [0-23]; Tag
[1-7];
# Datum [1-31]; Monat [1-12]; Jahr [0-99]

def check(num):
     '''Eine Funktion, die eine führende Null an eine einstellige
Zahl anhängt
          Rückgabe: String
     '''
     if num < 10:
          return '0{}'.format(num)
     else:
          return str(num)
```

```
print('[Drücken Sie CTRL + C, um das Skript zu beenden!]')
try:
     while True:
          print('\nSystemzeit: {}'.format(
               time.strftime('%Y-%m-%d %H:%M:%S')))
          data = ds3231.read_datetime() # Tupel zurückgeben
          print('RTC-Datum: {} {}.{}.{}'.format(data[0],
          data[1], check(data[2]), check(data[3])))
          print('RTC-Zeit: {}:{}:{}'.format(check(data[4]),
               check(data[5]), check(data[6])))

          # return string
          print('RTC Datum_Zeit: {}'.format(ds3231.read_str()))
          print('Temperatur: {:.1f}C'.format(ds3231.getTemp(),
               degree_sign))
          time.sleep(1)

except KeyboardInterrupt:
        print('\nScript Ende!')
```

Speichern Sie das Skript unter dem Namen *rtc.py* in demselben Verzeichnis, in dem auch das Skript *rtc_lib.py gespeichert* ist. Um das Skript auszuführen, öffnen Sie das Terminal in dem Verzeichnis, in dem das Skript gespeichert ist, und führen Sie den folgenden Befehl aus:

python3 rtc.py

Das Ergebnis sollte wie in der folgenden Abbildung aussehen:



Um das Skript zu stoppen, drücken Sie 'CTRL + C' auf der Tastatur.

Das Skript beginnt mit dem Import von zwei Bibliotheken, *time* und *rtc_lib*.

Als nächstes wird die Variable *degree_sign* erstellt. Der Wert dieser Variablen stellt das UTF8-Symbol für das Gradzeichen dar.

Dann wird das Objekt mit dem Namen *ds3231* mit der folgenden Codezeile erstellt:

```
ds3231 = rtc_lib.SDL-DS3231(1, 0x68)
```

Dabei steht die Zahl *0x68* für die I2C-Adresse des RTC-Moduls.

Danach wird das Datum und die Uhrzeit in RTC mit der folgenden Codezeile aktualisiert:

```
ds3231.write_now()
```

Die Funktion *write_now()* speichert im RTC-Modul das aktuelle Datum und die Uhrzeit des Betriebssystems Raspbian.

Es gibt eine weitere Möglichkeit, Zeit- und Datumsdaten zu speichern. Dies kann mit der folgenden Code-Zeile erreicht werden:

```
ds3231.write_all(Sekunden=Keine,Minuten=Keine,Stunden=Keine,Tag=Ke
ine,
Datum=Keine,Monat=Keine,Jahr=Keine,save_as_24h=True)
```

Die Funktion *write_all()* hat acht Argumente und gibt keinen Wert zurück. Die Argumente stellen einen Teil der Daten für Datum und Uhrzeit dar. Die Werte für alle Argumente sind ganzzahlige Zahlen in den Bereichen: Sekunden: 0 - 59, Minuten: 0 - 59, Stunden: 0 - 23 Tag: 1 - 7 (Tag in der Woche), Datum: 1 - 31, Monat: 1 - 12, Jahr: 0 - 99
Speichern_als_24h: Wahr/Falsch (getestet mit nur als Wahr)

Als nächstes wird eine neue Funktion namens *check()* erstellt. Die Funktion hat ein Argument und gibt einen String-Wert zurück. Sie wird verwendet, um eine führende Null zu einer einstelligen Zahl hinzuzufügen. Das Argument stellt die Zahl dar, die dann überprüft wird, ob es sich um eine ein- oder zweistellige Zahl handelt. Handelt es sich um eine einstellige Zahl, wird der String-Wert *'0{}'.format(num)* zurückgegeben. Handelt es sich nicht um eine zweistellige Zahl, wird der Stringwert *str(num) zurückgegeben*.

Danach wird der try-except-Codeblock erstellt. Im try-Block wird die unendliche Schleife erstellt. In der unendlichen Schleife werden die RTC-Daten gelesen und im Terminal angezeigt. Es gibt zwei Möglichkeiten, die Daten auf dem Terminal anzuzeigen. Die erste ist die Verwendung der Funktion *read_datetime()* und die zweite die Verwendung der Funktion *read_str()*.

Um das Skript anzuhalten, drücken Sie 'STRG + C' auf der Tastatur. Dies wird als Tastaturunterbrechung bezeichnet. Wenn die Tastaturunterbrechung erfolgt, wird der Codeblock mit der *Ausnahme* ausgeführt und die Meldung *Skript Ende!* im Terminal angezeigt.

Die Funktion *read_datetime()* hat zwei Argumente und gibt ein *Tupel* zurück. Das erste Argument steht für das Jahrhundert, einen ganzzahligen Wert, z. B.: für das 21. Jahrhundert verwenden Sie *century=21*. Jahrhundert, verwenden Sie century=21. Das zweite Argument steht für die Zeitzoneninformation, die für datetime-Objekte verwendet wird (sie wird in diesem eBook nicht behandelt). Der Rückgabewert ist ein *Tupel*, das aus sieben Elementen besteht. Die Elemente sind: *dayOfWeek*, *dayOfMonth*, *month*, *year*, *hour*, *minute* und *second* (in dieser Reihenfolge). Alle Werte für diese Elemente sind Ganzzahlen, außer dem Wert des Arguments *dayOfWeek*, der ein String-Wert ist, der den Namen des Wochentags darstellt.

Die Funktion *read_str()* hat ein Argument und gibt einen String-Wert zurück. Das Argument ist das Jahrhundert-Argument und wird als das Jahrhundert-Argument der Funktion *read_datetime()* verwendet. Der Rückgabewert ist ein String-Wert mit einem vordefinierten Format für Datums- und Zeitdaten, wie in der folgenden Skriptausgabe gezeigt:

```
Systemzeit: 2020-03-15 15:44:15
RTC Datum: Sonntag 15.03.2020
RTC-Zeit: 15:44:15
RTC Datum/Uhrzeit: Sonntag 15-03-2020 15:44:15
Temperatur: 28,8°C
```

Um Temperaturdaten vom RTC-Modul zu erhalten, verwenden Sie die Funktion *detTemp()*. Die Funktion hat keine Argumente und gibt einen Float-Wert zurück. Der Rückgabewert, ein Float-Wert, stellt die Temperaturdaten in Celsius dar.

Jetzt ist es an der Zeit, zu lernen und eigene Projekte zu erstellen. Das können Sie mit Hilfe vieler Beispielskripte und anderer Anleitungen tun, die Sie im Internet finden können.

Wenn Sie auf der Suche nach hochwertiger Mikroelektronik und Zubehör sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.

https://az-delivery.de

Viel Spaß!

Impressum

https://az-delivery.de/pages/about-us