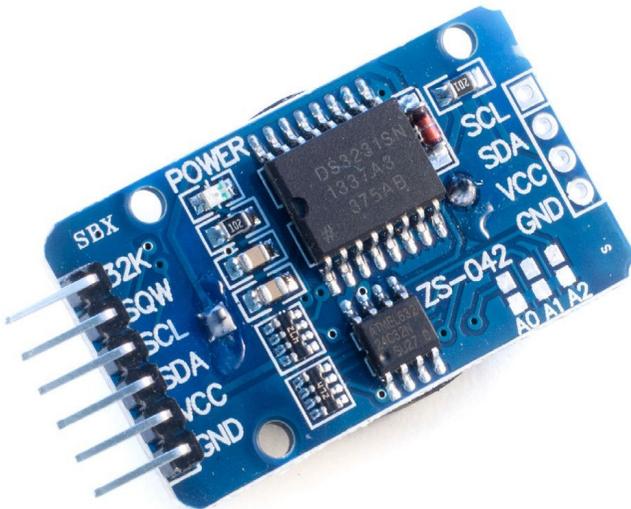


Willkommen!

Und herzlichen Dank für den Kauf unserer **AZ-Delivery Real Time Clock ZS-042**! Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung am Arduino bis zum ersten Script. Viel Spaß!



<http://flyt.it/RTC>

Die **AZ-Delivery Real Time Clock ZS-042**, kurz **RTC**, ist ein externer Zeitgeber auf Basis des **DS3231**-Chipsatzes beispielsweise für den Einsatz mit Arduino & Co. Dank eigener wiederaufladbarer Energiequelle läuft er autonom und kann, einmal eingerichtet, andere Controller mit einem absoluten Zeitwert versorgen. Die Datenübertragung erfolgt dabei über die **I²C**-Schnittstelle.

Die wichtigsten Informationen in Kürze

- » **Chipsatz:** DS3231
- » **Abmessungen:** 37x21,5x11mm (ohne Pins)
- » **Stromversorgung:** 3V CR2032 Lithium-Akku / VCC
- » **Datenschnittstelle:** I²C
- » **Temperatursensor:** ± 3°C Genauigkeit

- » **Uhrfunktionen:**
 - » Jahr, Monat (Name & Zahl), Tag, Stunde, Minute, Sekunde
 - » Schaltjahrberechnung bis 2100
 - » 2 Tagesalarme
 - » ±0.0002% Abweichung bei 0-40°C
 - » ±0.00035% Abweichung bei -40-85°C

Auf den nächsten Seiten findest du Informationen zur

» ***Verbindung mit dem Arduino***

und eine Anleitung für

» ***ein Uhrzeit-gesteuertes Blink-Event.***

Diese Anleitung setzt voraus, dass du weißt, wie du Sketche auf einen Arduino hochlädst und den Serial Monitor verwendest!

Alle Links im Überblick

DS3231:

- » Datenblatt:

<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>

- » Arduino-Bibliothek mit Beispielsketch:

<https://github.com/StephanFink/RTClib>

Programmieroberflächen:

- » Arduino IDE: <https://www.arduino.cc/en/Main/Software>

- » Web-Editor: <https://create.arduino.cc/editor>

- » Arduino-Erweiterung für SublimeText:

<https://github.com/Robot-Will/Stino>

- » Arduino-Erweiterung "Visual Micro" für Atmel Studio oder Microsoft Visual Studio:

<http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx>

Arduino Tutorials, Beispiele, Referenz, Community:

- » <https://www.arduino.cc/en/Tutorial/HomePage>

- » <https://www.arduino.cc/en/Reference/HomePage>

Interessantes von AZ-Delivery

- » Arduino Zubehör:

<https://az-delivery.de/collections/arduino-zubehor>

- » AZ-Delivery G+Community:

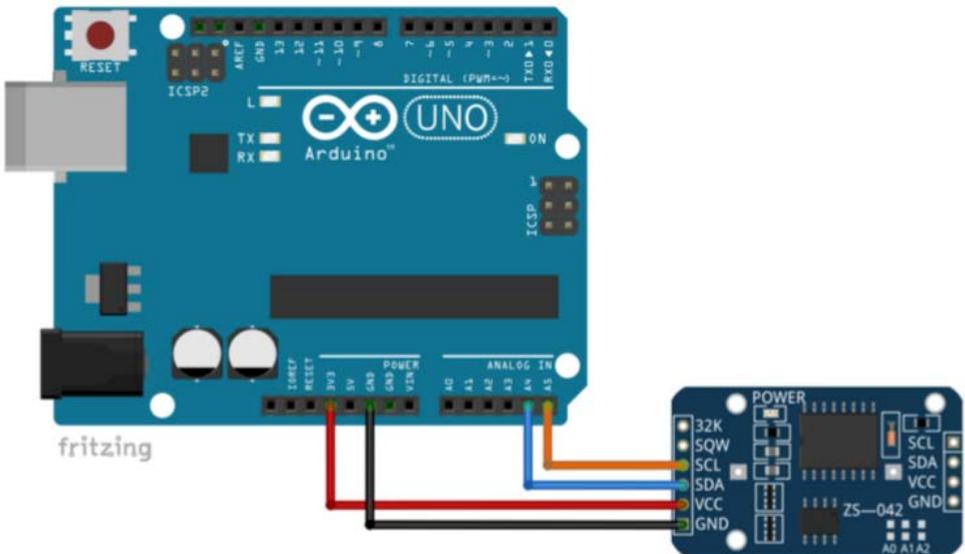
<https://plus.google.com/communities/115110265322509467732>

- » AZ-Delivery auf Facebook:

<https://www.facebook.com/AZDeliveryShop/>

Verbindung mit dem Arduino

Dank der 2,3-5,5 V Spannungstoleranz des **DS3231** ist die RTC sehr Controller-freundlich und lässt sich mit allen Arduino-Typen verwenden. An einen UNO, wie er auch in diesem Tutorial verwendet wird, schließt du ihn wie folgt an:



Installation der DHT22-Bibliothek

Die **DS3231** RTCs sind so beliebt, dass im Internet gleich mehrere Arduino-Bibliotheken zur Verfügung stehen. **AZ-Delivery** stellt selbst eine Bibliothek bereit, die auf der des Unternehmens Adafruit aufbaut und um den Beispielsketch erweitert wurde, welcher in diesem Tutorial zum Einsatz kommen wird. Lade sie unter folgender Adresse herunter:

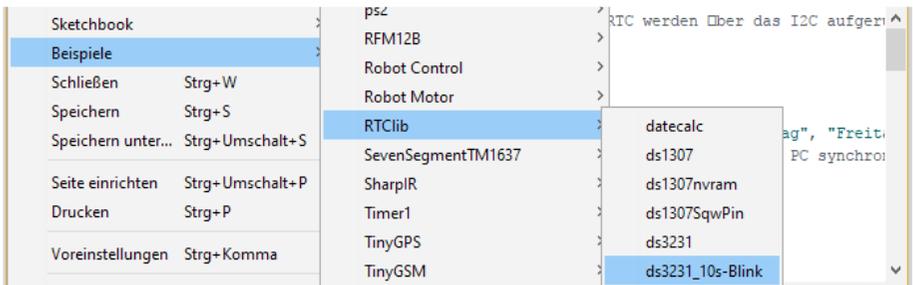
» <https://github.com/StephanFink/RTCLib/archive/master.zip>

Entpacke anschließend den Ordner "**RTCLib-master**" aus dem Zip-Archiv in das libraries-Verzeichnis deines Arduino Workspace, z.B. "**Eigene Dokumente > Arduino > libraries**" und benenne ihn in "**RTCLib**" um. Wenn du nun die Arduino IDE (neu) startest, kannst du unter "**Sketch > Bibliothek einbinden**" und "**Datei > Beispiele**" das Unterverzeichnis "**RTCLib**" finden.

Das erste Script

Während in den meisten Programmiersprachen der erste Erfolg ein zu lesendes "Hello World!" darstellt, ist es bei Arduinos das Blinken der boardinternen LED. Mithilfe der RTC lassen wir diese nun immer zur vollen zehnten Sekunde aufleuchten.

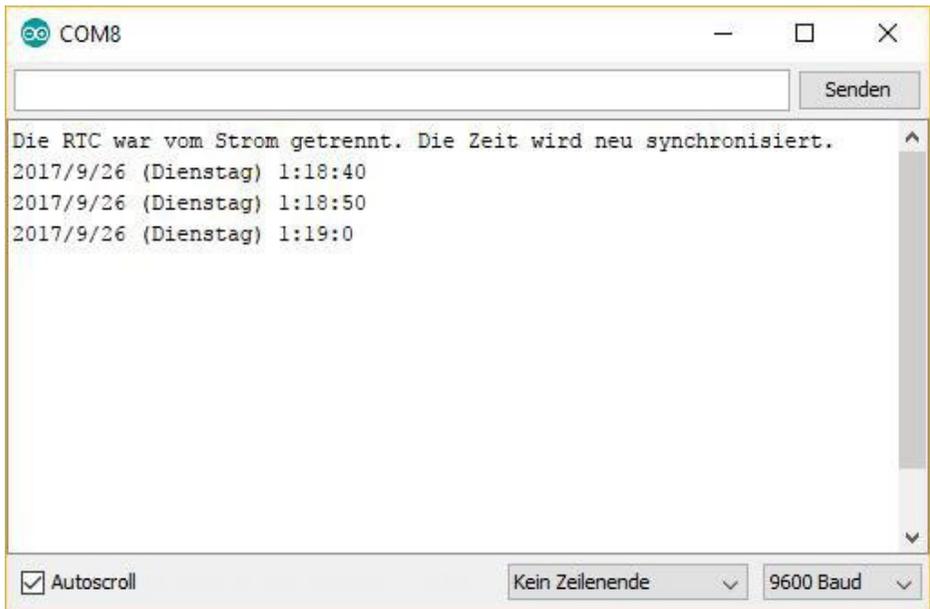
Öffne dazu unter "**Datei > Beispiele > RTCLib**" den Beispielsketch "**ds3231_10s-Blink**":



Wenn die RTC zwischenzeitlich ohne Strom war, wird sie beim Upload des Sketches mit der aktuellen Zeit deines PCs versorgt. Beachte dabei aber, dass sie von Anfang an einige Sekunden nachgehen wird, denn der übertragene Zeitstempel ist der, an dem der Kompilierungsvorgang gestartet wurde. Die RTC beginnt aber erst zu zählen, wenn der Uploadvorgang abgeschlossen ist. Die Differenz beträgt allerdings meist nicht mehr als 15 Sekunden.

Willst du die RTC zwischendurch manuell synchronisieren, stelle die Variable "**synchOnFirstStart**" in Zeile 7 auf "**true**" und lade den Sketch erneut hoch.

Starte anschließend das Terminal mit der richtigen Baud-Rate und es wird exakt aller zehn Sekunden die aktuelle Zeit inkl. Datum und Wochentag angezeigt:



The screenshot shows a terminal window with the title 'COM8'. The window contains the following text:

```
Die RTC war vom Strom getrennt. Die Zeit wird neu synchronisiert.  
2017/9/26 (Dienstag) 1:18:40  
2017/9/26 (Dienstag) 1:18:50  
2017/9/26 (Dienstag) 1:19:00
```

At the bottom of the window, there are several controls: a checked 'Autoscroll' checkbox, a dropdown menu set to 'Kein Zeilenende', and another dropdown menu set to '9600 Baud'. A 'Senden' button is located at the top right of the terminal area.

Währenddessen leuchtet auf dem Arduino exakt zeitgleich die boardinterne LED für eine Sekunde auf.

Du hast es geschafft! Herzlichen Glückwunsch!

Ab jetzt heißt es lernen und ausprobieren. Schau dir dazu am besten den Sketch und die Kommentare darin an, um herauszufinden, wie du auf die einzelnen Daten der RTC zugreifen kannst. Mit diesem Modul kannst du nun zum Beispiel zu jedem selbstgewählten Zeitpunkt andere an den Arduino angeschlossene Hardware aktivieren und steuern. Diese findest du natürlich in deinem Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!

Impressum

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