

## Tutorial – Real time clock Modul mit ESP 8266 nutzen

### 1. Voraussetzungen

Wir benötigen die Entwicklungsumgebung Arduino (unter [www.arduino.cc](http://www.arduino.cc) herunterladen). Zudem sollten dort die Boards und Treiber zur Kommunikation über USB für die ESP Prozessoren eingerichtet sein.

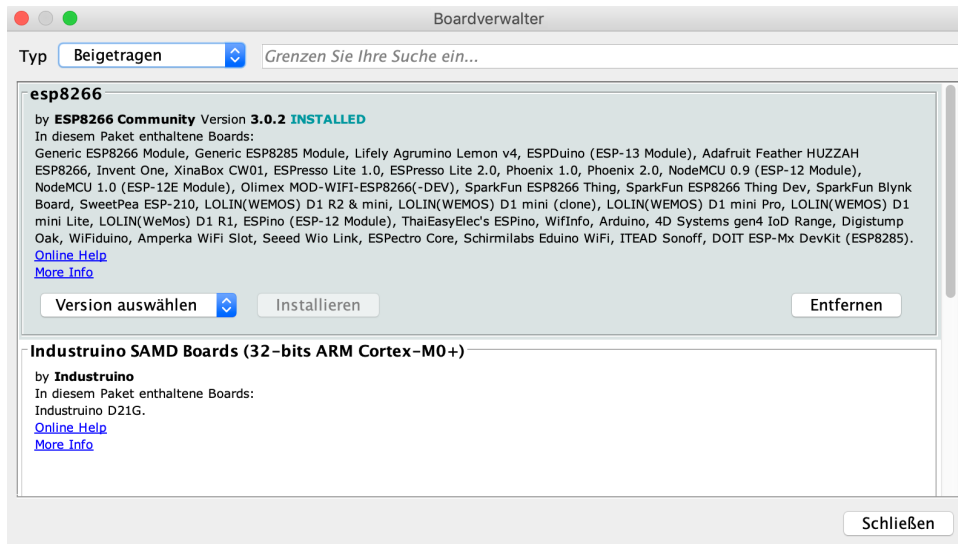


Abbildung 1: Boardverwalter mit dem entsprechenden zu installierenden Board-Satz (ESP 8266)

Des weiteren brauchen wir die im folgenden aufgeführten Bibliotheken: Time.h, Wire.h welches beide Standard Arduino Bibliotheken sind. Dazu muss dann noch die RTCLib heruntergeladen werden.

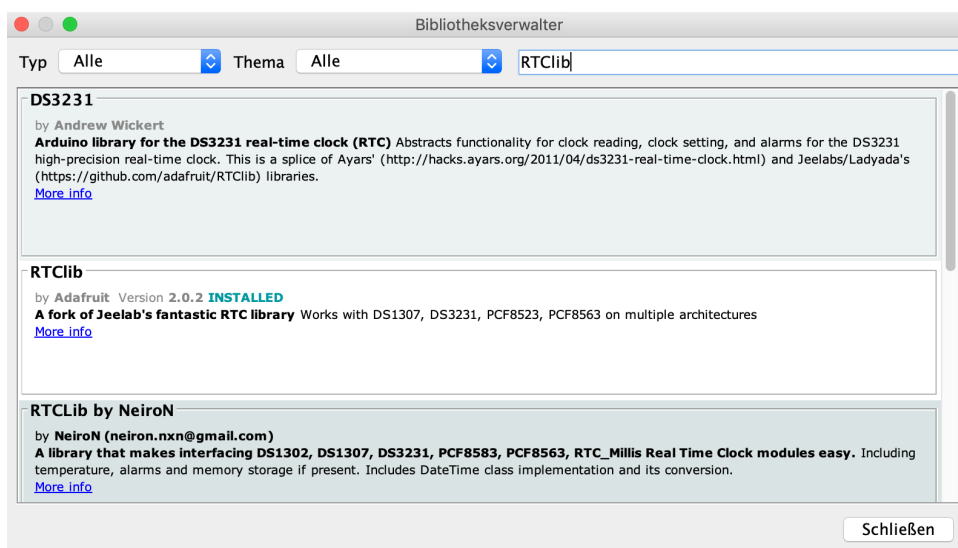


Abbildung 2: Bibliotheksverwaltung mit der benötigten RCTlib

## 2. Verdrahtung ESP Board mit dem RTC Modul

An dieser Stelle verzichten wir auf eine Abbildung, da die verfügbaren PINs doch sehr überschaubar sind. In der Tabelle sind die PINs genannt, die miteinander verbunden werden müssen.

*Tabelle 1: PIN-Belegung*

	ESP 8266 PIN	RCT Pin
<b>Spannungsversorgung</b>	3.3 V	VIN
<b>Masse</b>	GND	GND
<b>Serial Data</b>	D1	SDA
<b>Serial Clock</b>	D2	SCL

Anschließend kann die USB Verbindung hergestellt werden und die Arduino Oberfläche entsprechend auf den verwendeten Mikroprozessor (Board) eingerichtet werden.

## 3. Quellcode

Vorgehen entweder anhand des YouTube Videos oder einfach den im Github vorliegenden Code herunterladen (git clone <https://github.com/professorgraf/RealTimeClock4ESP.git> .), bzw. als ZIP öffnen und auf den Mikroprozessor aufspielen.

```
/*
*****
* Real time clock demonstration project
*
* RealTimeClock4ESP
*
* 2022-03-19 by Prof. Dr. Markus Graf
* This code is provided on an as-is base
* under the GPL v3 license
*/
#include <Wire.h>
#include <Time.h>
#include <RTCLib.h>

// mit der RTC zu verbinden (über I2C TwoWire, PCF8563 chip)
RTC_PCF8563 theClock;
TwoWire wire;

// timer, keeps track within the loop in order to
long ms;

void setup()
{
  Serial.begin(115200);
  wire.begin( 5, 4 ); // D1 =5, D2 = 4
  theClock.begin( &wire );

  Serial.println();
  if ( theClock.isrunning() )
  {
    Serial.println("RTC is up and running");
  }
  else
  {
    Serial.println("RTC is not working");
  }

  if ( theClock.lostPower() )

```

```
{
  Serial.println("RTC not initialized.");
  theClock.adjust( DateTime(F(__DATE__), F(__TIME__)) );
}

ms = millis();
}

// repeatedly go through this code and every second update the time (and send it to
the Monitor for info)
// please modify for the project
void loop()
{
  if ( millis() - ms > 1000 )
  {
    DateTime tm = theClock.now();
    Serial.println( tm.timestamp() );
    ms = millis();
  }
}
```

#### 4. Fazit

Ich hoffe, die Durchführung hat problemlos geklappt. Im nächsten Teil kombinieren wir das RTC Modul mit einem DCF-Empfänger, damit die Uhrzeit und das Datum automatisch über Funk eingestellt werden kann.