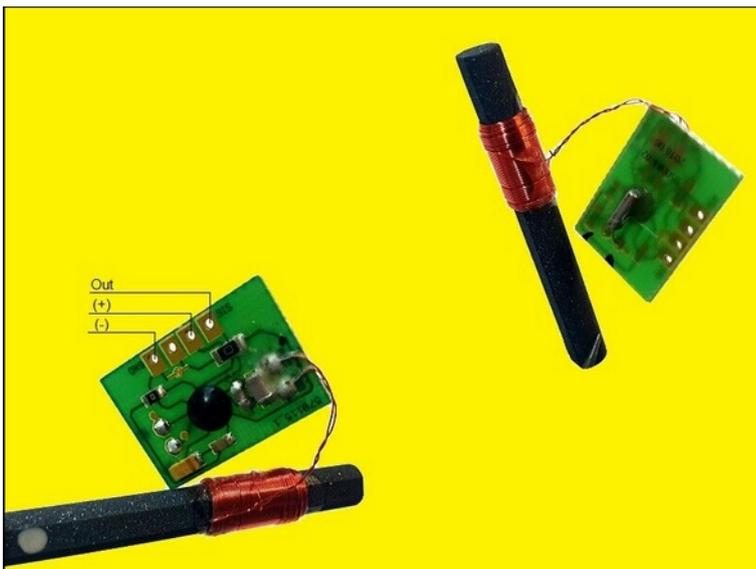


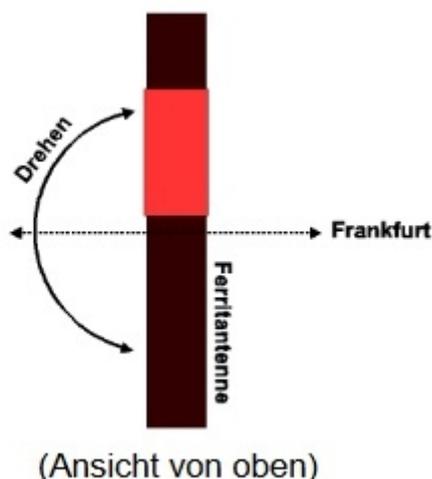
# DCF-Signal

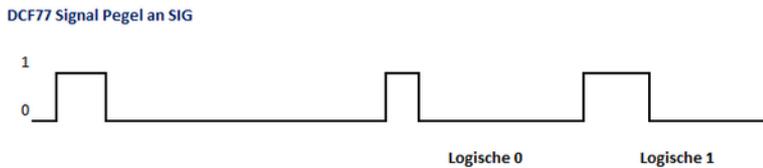
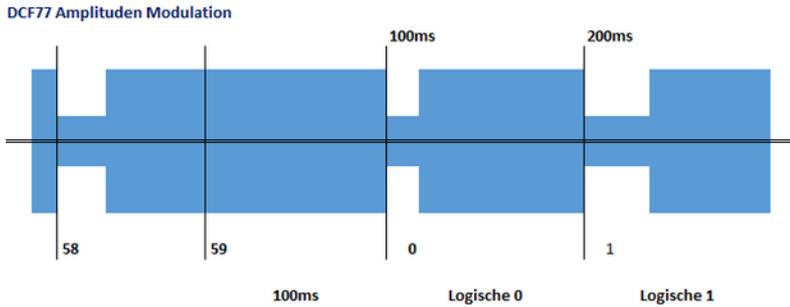
Die [Physikalisch-Technische Bundesanstalt](#) (PTB) in Braunschweig sendet seit 1978 per Mittelwellensender auf der **Frequenz 77,5 kHz** die aktuelle Uhrzeit aus, die mit hochpräzisen Cäsium-Uhren gezählt wird. Das Signal senkt jede Sekunde seine Amplitude um 25%. Je nachdem, ob dieser Rückgang der Amplitude etwa 50 ms oder etwa 150 ms dauert, kann man daraus eine binäre 0 oder 1 erkennen. **Pro Minute werden 59 Bits übertragen**, in denen die aktuelle Uhrzeit und das Datum sowie einige Prüfbits enthalten sind. In der letzten Sekunde fehlt die Absenkung der Amplitude, wodurch der Anfang einer Minute und damit auch der Anfang einer Datensequenz erkannt werden kann.

Der Name DCF77 ist durch internationale Vereinbarungen entstanden. DCF77 setzt sich aus dem Buchstaben D für Deutschland, dem C als Kennzeichen eines Langwellensenders und dem F wegen der Nähe zu Frankfurt (Mainflingen liegt nahe bei Frankfurt) zusammen. Die 77 steht für die verwendete Sendefrequenz. Das Rufzeichen DCF77 wird dreimal stündlich, als Morsezeichen, während der Minuten 19, 39 und 59 gesendet.



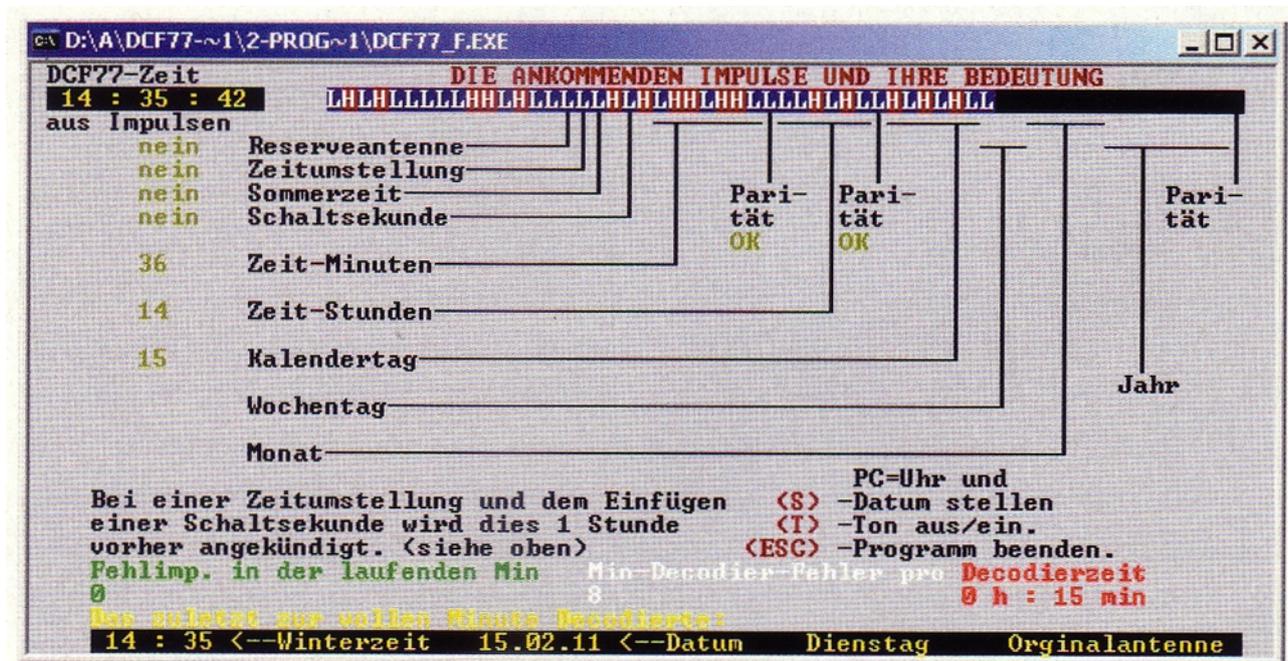
<https://arduino-hannover.de/2012/06/14/dcf77-empfangen-mit-arduino-betreiben/>





In jeder Sekunde innerhalb einer laufenden Minute kommt ein Impuls mit einer Länge von entweder 100 ms (als Low) oder 200 ms (als High). Nur in der 59. Sekunde fehlt dieser Impuls. Dies ist die Synchronisationslücke. Man kann ab dieser Lücke, also beginnend mit der neuen Minute, jedem Impuls seine Bedeutung zuordnen. In [1] wird dies genau Bit für Bit erklärt.

Seit Ende 2006 werden auch Wetterinformationen für 90 Regionen innerhalb Europas übertragen. Diese sind in den „Sekunden-Bits“ 1 bis 14 integriert. Entsprechende Wetterstationen gibt es in mehreren Varianten im Handel. Die Wetterinformation kann mit dem DOS-Programm nach [1] nicht decodiert werden. Meines Wissens gibt es kein Programm, welches die Wetterinformationen decodieren kann. Die Codierung ist lizenzgeschützt.







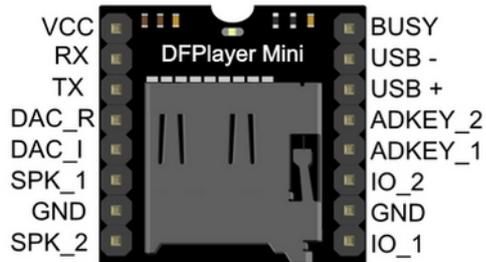
DCF77:

D5 (GPIO 14)= DCF77-Signal

Spannung: 3,3V und GND

DFPlayerMini:

Sowohl NodeMCU als auch DFPlayer haben 3,3V-Signal auf der seriellen Schnittstelle, also kann hier direkt verbunden werden:



VCC: Vin vom NodeMCU (=5V vom USB)

GND: GND vom NodeMCU

RX: TX von NodeMCU

TX: RX von NodeMCU

SPK\_1 und SPK2= Monolautsprecher

Zusätzlich fetten ELKO zwischen VCC und GND

Sprache erzeugen:

<https://ttsmp3.com/text-to-speech/German/>

Fritzing:

NodeMCU holen:

<https://forum.fritzing.org/t/heltec-dev-boards/8112>

[Amica NodeMCU DevKit.fzpz](#) downloaden

In Fritzing: Im rechten fenster unter Bauteile MINE auswählen und mit der rechten Maustaste IMPORTIEREN wählen, dann im Downloadbereich „Amica NodeMCU DevKit.fzpz“ auswählen.

DCF77-Empfänger ACHTUNG – Pinbelegung stimmt nicht!:

<https://forum.fritzing.org/t/pollin-dcf-1-rdio-controlled-clock/5119>

[DCF1 Pollin.fzpz](#)

Platine:

- Node MCU und DFPlayer mini auf Stiftleisten sockeln (im Bild unten sind keine Sockel).
- Die drei größeren Löcher sind für Schrauben
  - Achtung ! Beim Kondensator Minus rechts (wie unten dargestellt) !

Schalter:  
links (wie dargestellt): Serielle Schnittstelle nicht mit MP3-Player verbunden  
rechts: Serielle Schnittstelle mit MP3-Player verbunden

Um den Node MCU programmieren zu können, muss der Schalter links sein, um Töne abzuspielen muss der Schalter rechts sein.

