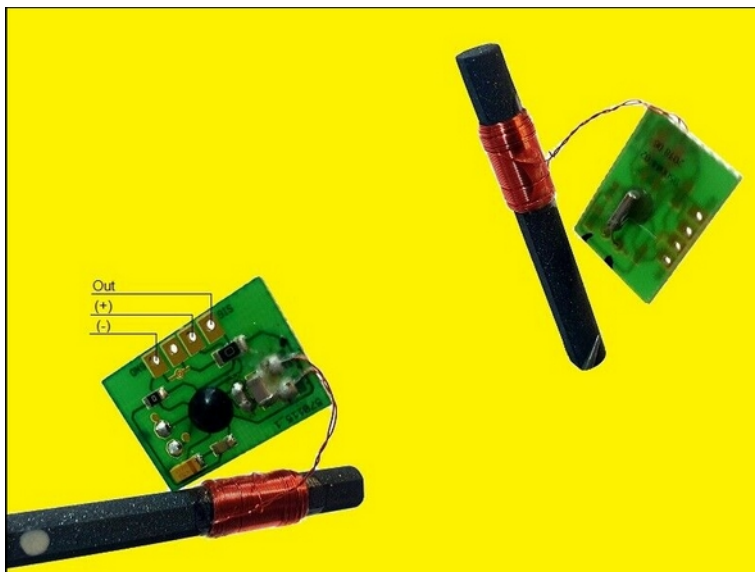


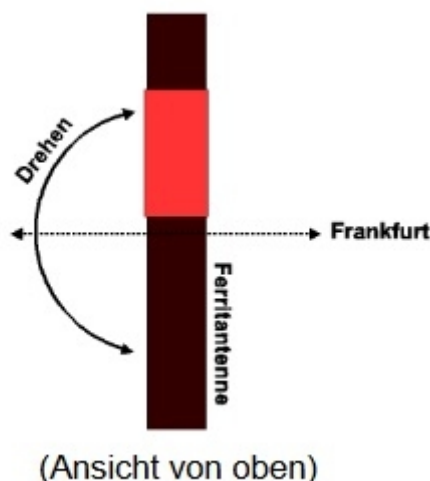
DCF-Signal

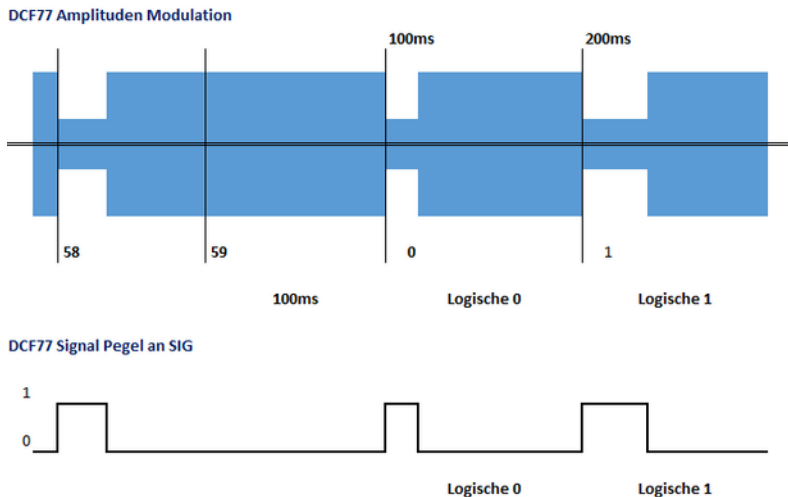
Die [Physikalisch-Technische Bundesanstalt](#) (PTB) in Braunschweig sendet seit 1978 per Mittelwellensender auf der **Frequenz 77,5 kHz** die aktuelle Uhrzeit aus, die mit hochpräzisen Cäsium-Uhren gezählt wird. Das Signal senkt jede Sekunde seine Amplitude um 25%. Je nachdem, ob dieser Rückgang der 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.



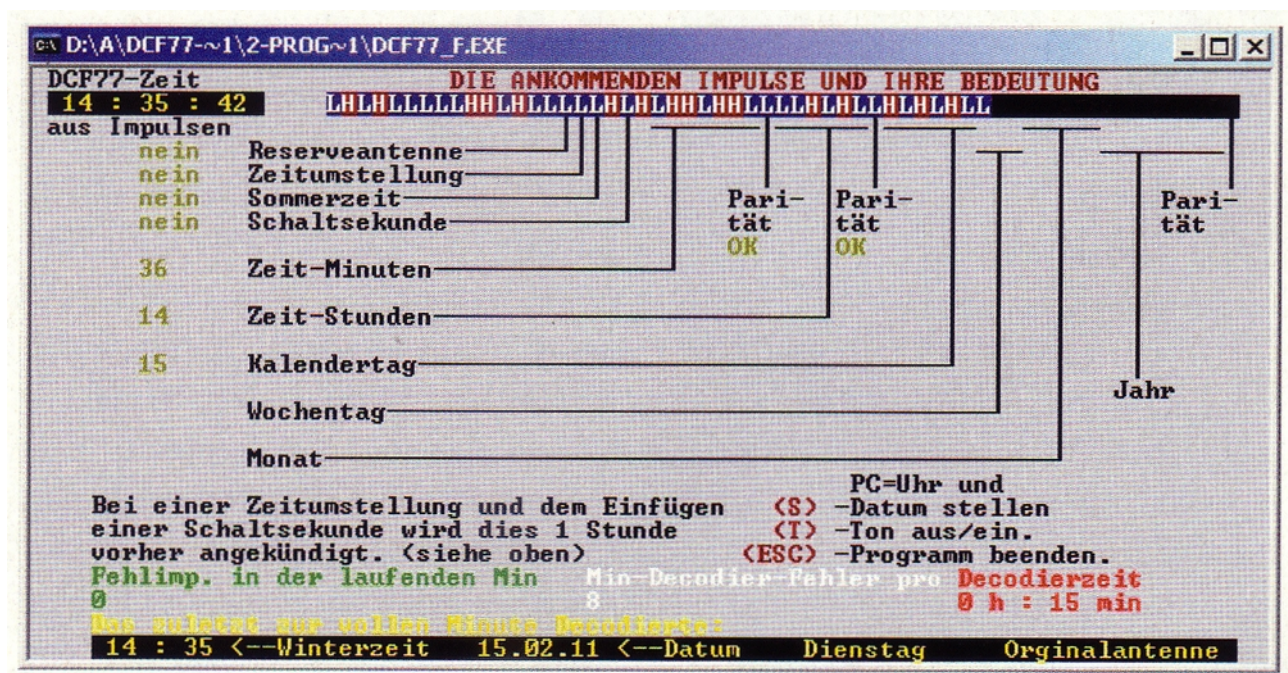
<https://arduino-hannover.de/2012/06/14/dcf77-empfanger-mit-arduino-betreiben/>





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.



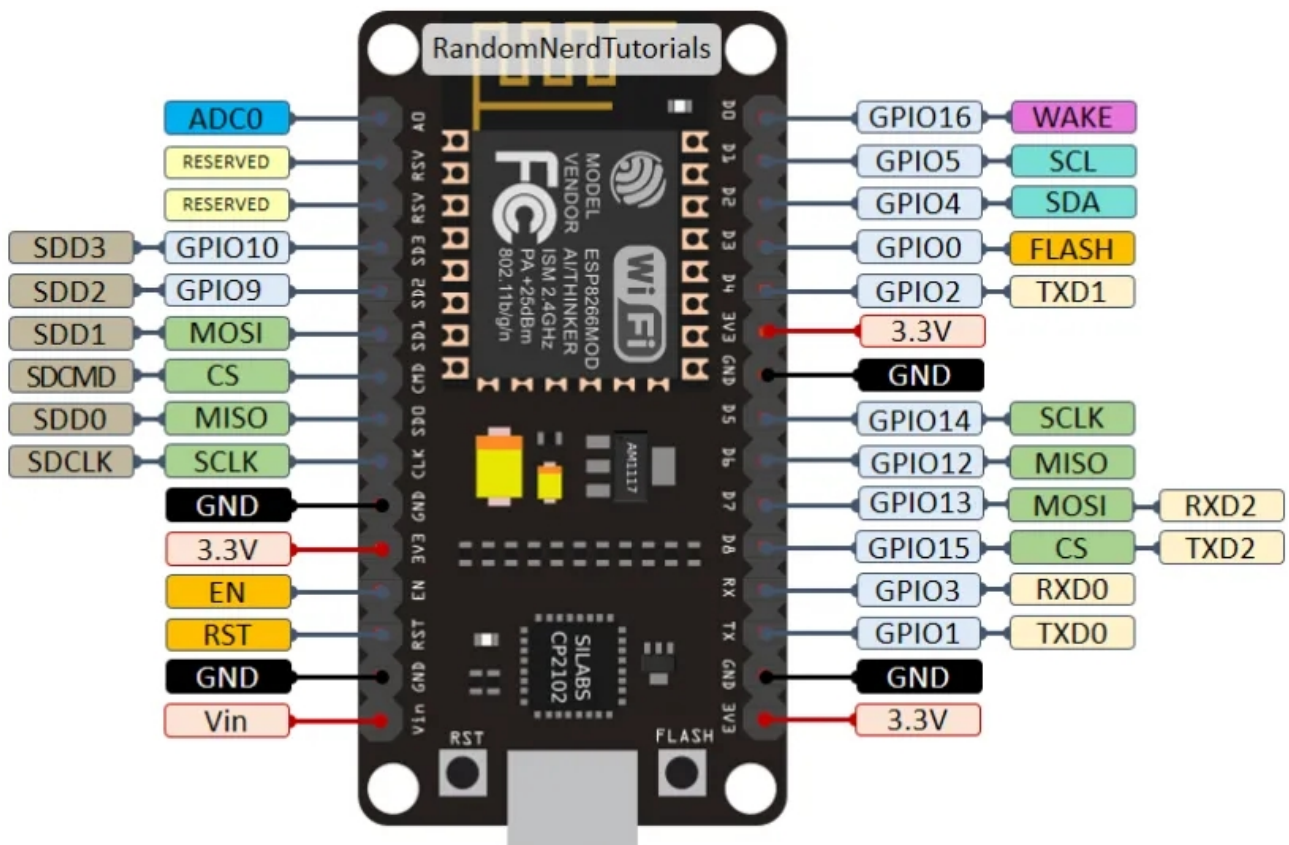
[illegible]

Sekunde	Bit	Bemerkung
0		Start einer neuen Minute (immer '0')
1 - 14		Wetterinformation
15	R	Antennenbit
16	A1	'1': am Ende der Stunde wird MEZMESZ geändert
17	Z1	'0': MEZ, '1': MESZ
18	Z2	'0': MESZ, '1': MEZ
19	A2	'1': am Ende der Stunde wird Schaltsekunde eingefügt
20	S	Beginn der Zeiteinformation (ist immer '1')
21 - 27		Minute (Werte der Bits: 1, 2, 4, 8, 10, 20, 40)
28	P1	Probit Minute
29 - 34		Stunde (Werte der Bits: 1, 2, 4, 8, 10, 20)
35	P2	Probit Stunde
36 - 41		Tag (Werte der Bits: 1, 2, 4, 8, 10, 20)
42 - 44		Wochentag (Werte der Bits: 1, 2, 4)
45 - 49		Monat (Werte der Bits: 1, 2, 4, 8, 10)
50 - 57		Jahr (zweistellig, Werte der Bits: 1, 2, 4, 8, 10, 20, 40, 80)
58		Probit Datum
59	P3	Keine Absenkung der Amplitude (außer bei Schaltsekunde)

HB9EFY 03/2009

Second	Bit	Comment
0		Start of a new minute (always '0')
1 - 14		Weather information
15	R	antenna bit
16	A1	'1': at the end of the hour MEZ/MESZ will change
17	Z1	'0': MEZ, '1': MESZ
18	Z2	'0': MESZ, '1': MEZ
19	A2	'1': at the end of the hour leap second will be added
20	S	Start of the time information (always '1')
21 - 27		Minute (value of bits: 1, 2, 4, 8, 10, 20, 40)
28	P1	parity bit minute
29 - 34		Hour (value of the bits: 1, 2, 4, 8, 10, 20)
35	P2	parity bit hour
36 - 41		Day (value of the bits: 1, 2, 4, 8, 10, 20)
42 - 44		Week day (value of the bits: 1, 2, 4)
45 - 49		Month (value of the bits: 1, 2, 4, 8, 10)
50 - 57		Year (2 digits; value of the bits: 1, 2, 4, 8, 10, 20, 40, 80)
58	P3	parity bit date
59		No lowering of amplitude (except for leap second)

Information	Weather information												z.B. time system			Minute		Hour		Day		Week day		Month		Year																																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Second	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Bit	x	x	x	x	x	x	x	x	x	x	x	x	x	x	R	A1	Z1	Z2	A2	S	1	2	4	8	10	20	40	P1	1	2	4	8	10	P2	1	2	4	8	10	20	1	2	4	1	2	4	8	10	1	2	4	8	10	20	40	80	P3			



ESP 8266

Board: Node MCU 1.0 (ESP-12E Module)

Upload Speed: 921600

In der Arduino IDE wird als Pinnummer die GPIO-Nummer angegeben (ohne den Vorsatz „GPIO“).

Es darf keine Endlosschleifen geben, weil immer, wenn `void loop()` zu Ende ist, werden erst WiFi Routinen ausgeführt und dann läuft wieder `loop`. => nie zu lange blockieren niemals mehr als 3 Sekunden!

LCD mit I²C anschließen:

Auf Vin liegen die 5V vom USB, hier abzapfen z.B. für ein LCD zum Debuggen

D1 = SCL

D2 = SDA

GND natürlich auch verbinden.

Code:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, Anzahl Spalten, Anzahl Zeilen);
im setup:
lcd.init();
lcd.clear();
lcd.backlight();
lcd.print(„hello world“);
```

Onboard LED = GPIO 16

Mehr als 3.6V auf einen Pin gegeben tötet diesen !!!

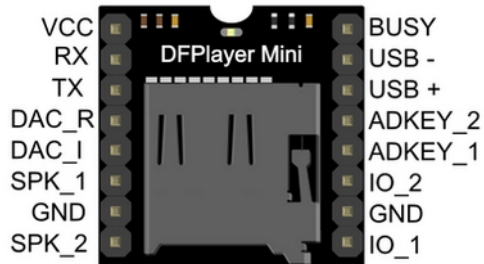
DCF77:

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://tsmp3.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.

