



## DCF-77 Dekoder

Echtzeituhren werden bei mittleren und grösseren Systemen immer wichtiger. Die AVRco Bibliothek enthält deshalb einen Echtzeit-Uhr Treiber, der sowohl mit der internen Echtzeituhr diverser AVR's als auch mit dem SysTick als Grundlage auskommt. Die interne RTC ist dabei etwas genauer als die Version mit dem SysTick, wobei für die meisten Fälle die Genauigkeit der SysTick Version ausreicht.

Mit der Präzision gibt es also keine Probleme. Anderst sieht es aber mit der Initialisierung der RTC aus. Hier muss nach jedem Power-Up die Uhr neu gestellt werden. Neben dem zusätzlichen Hard- und Software Aufwand, den man benötigt, um die Uhr stellen zu können, stellt sich auch noch die Gefahr einer falschen Eingabe durch den Benutzer oder sogar ein Vergessen des Stellvorgangs.

Um diesen Problemen und Kosten aus dem Weg zu gehen bieten sich zwei Möglichkeiten.

1. Einsatz eines zusätzlichen Uhrenchips mit Akkupufferung. Hier erledigt sich das Stellen der Uhr von selbst bzw. es bleibt beim einmaligen Stellen. Allerdings sind dafür zumindest zusätzliche Portpins (6..8) bzw. bei einer I2C-Uhr wird ein weiterer Treiber gebraucht. Weiterhin ist langfristig mit einer Abweichung der Uhr zu rechnen, ganz abgesehen von Schaltjahr und Winter/Sommerzeit Umstellungen.
2. Einsatz eines DCF77 Empfängers. Hier gibt es absolut keine Stell- oder Genauigkeits Probleme. Einzig die erforderliche Setup-Zeit von bis zu 2 Minuten kann manchmal etwas stören. Je nach Umgebungsbedingungen kann u.U. der Platz bzw. Position der Ferrit Antenne etwas kritisch sein. Der DCF77 Treiber klinkt sich in den SysTick ein.

Wenn man die Wahl hat, sollte man der DCF77 Uhr den Vorzug geben. Bei einigermaßen erträglichen Empfangsbedingungen ist mit einem problemlosen Betrieb zu rechnen.

Die DCF77 Uhr ist jedoch keine Echtzeit Uhr im herkömmlichen Sinn. Sie muss immer in Verbindung mit einer anderen Uhr gesehen werden. Das AVRco System bietet hier drei Möglichkeiten:

1. DCF77 steuert die RTC einer AVR CPU. Hierzu ist der RTC Treiber zu importieren
2. DCF77 steuert die RTC, welche vom SysTick angetrieben wird.
3. DCF77 steuert einen externen Uhren Chip

Die Punkte 1 und 2 sind aus der Sicht des DCF77 Treibers identisch. Wird eine fehlerfreie Zeit empfangen, so wird bei jeder vollen Minute die RTC neu gestellt. Das geschieht automatisch ohne weiteres Zutun. Der DCF77 Treiber erkennt die RTC und handelt entsprechend.

Der Punkt 3 (Uhren Chip) wird durch eine **Callback** Prozedur unterstützt. Ist diese Prozedur vorhanden, wird sie bei jeder vollen Minute vom Treiber aufgerufen, wenn die Zeit fehlerfrei empfangen wurde. Die Callback Prozedur, die der Programmierer bereitstellen muss, kann jetzt das Uhrenchip stellen, indem die jeweiligen Speicherstellen der DCF77 in das Chip geschrieben wird. Diese Prozedur wird aus dem SysTick heraus aufgerufen. Hierbei ist die Register Rettung zu beachten, weiterhin sollte die Prozedur sehr schnell sein, um den hier gesperrten globalen Interrupt nicht zu lange zu sperren.

Der Uhrenchip ist aber ein Sonderfall. In den allermeisten Fällen genügt der RTC Treiber, um alle Aufgaben erledigen zu können.

### Import

Der Treiber muss, wie beim AVRco üblich, importiert werden.

```
Import SysTick, DCFclock, ..;
```

Der als DCF-Eingang verwendete Portpin sowie die Lage der DCF Uhren Register werden mit Define vorgegeben:

```
Define {mega103}
ProcClock      = 8000000;      {Hertz}
SysTick        = 10 , Timer2;  {msec}
DCFclock       = iData;
DCFport        = PinD, 2, negative; {Port, bitnummer, Polarität}
DCFfieldMode   = reset;      // decrement, optional "reset"
```



**Achtung:** Der SysTick muss bestimmte Bedingungen erfüllen. Er darf nicht kleiner 1msec oder grösser 100msec sein, er muss ganzzahlig sein und  $1000 / \text{SysTick}$  darf keinen Rest ergeben ( $1000 \bmod \text{SysTick} = 0$ ).

## DCF77-Funktionen/Prozeduren

Grundsätzliche und Implementations unabhängige Prozedure und Funktionen.

**Function** DCFready : boolean;

Das Ergebnis der Funktion wird wahr, wenn der DCF77 Treiber zum ersten mal ein komplettes Telegramm erhalten hat und im Falle der RTC-Implementation, die RTC komplett initialisiert hat. Ab diesem Moment kann man davon ausgehen, dass die Daten der RTC absolut stimmen. Dieser Vorgang dauert nach dem Einschalten mindestens 1 Minute und wenn der Empfang ok ist, maximal 2 Minuten, ansonsten kann es mehrere Minuten dauern.

**Function** DCFfield : byte;

Das Ergebnis der Funktion gibt einen ungefähren Wert für die Empfangsqualität wieder. Null ist kein Empfang, 255 ist optimaler Empfang. Es dauert mehrere Minuten bis sich dieser Wert stabilisiert hat.

Mit jedem gültigen Frame (1sec) wird der Wert um 1 inkrementiert. Bei fehlerhaftem Frame gibt es zwei Möglichkeiten:

Ist das Define **DCFfieldMode** auf „decrement“ gesetzt, wird bei einem Fehler der Wert um 1 dekrementiert.

Ist das Define **DCFfieldMode** auf „reset“ gesetzt, wird bei einem Fehler der Wert auf 0 gesetzt.

**Function** DCFDayLightSave : boolean;

Das Ergebnis der Funktion gibt die Sommer/Winterzeit zurück. Sommerzeit = true.

**Procedure** DCFupdate; // Callback Procedure

Findet der DCF77 Treiber den Import der RTC nicht, so ruft er jede Minute die Prozedur „DCFupdate“ auf, falls diese vorhanden ist. Diese Prozedur kann jetzt z.B. ein externes Uhrenchip beschreiben. Dazu können die Speicherstellen/Variablen

DCF\_SECOND, DCF\_MINUTE, DCF\_HOUR, DCF\_DAY, DCF\_MONTH, DCF\_YEAR, DCF\_WEEKDAY

benutzt werden. Hierbei ist unbedingt zu beachten, dass der Aufruf dieser Funktion aus dem SysTick heraus erfolgt. Evtl. benutzte Register retten. Möglichst sehr schnellen Assembler Code verwenden.

## Hardware

DCF77 Empfänger gibt es als fertige kleine Boards als auch als Wandmontage Gehäuse. Die Wandmontage ist vorzuziehen, da der Standort bzw. der Abstand von einer Störquelle, z.B. PC eminent wichtig ist. Ein gestörter Empfang kann dazu führen, dass der DCF77 Treiber viele Minuten braucht, bis er zum ersten mal ein fehler freies Telegramm empfangen hat. Bis dahin arbeitet die angebundene RTC mit Hausnummern.

Ein DCF77 Empfangsmodul schickt ein bitserielles Telegramm zur CPU. Bei der CPU genügt ein einziger Input Pin zum Empfang.

## Programm Beispiel

Ein Beispiel Programm ist unter `..\AVRco\Demos\DCF77` zu finden.