

DCF-RS1

Version 1.1

DCF-RS232 Wandler mit Uhr

- Decodiert automatisch DCF-Signal und gibt Uhrzeit digital aus
- Automatische Verifizierung der DCF-Uhrzeit
- Ausgabe erfolgt im TTL Rs232 Format (2400 Baud, 8 Bit, 1 Stop-Bit, None Parität)
- Ausgabe wahlweise als Text oder Binär
- Eingebaute Uhr und Kalender überbrückt DCF-Empfangsausfall, auch über Tage
- Ausgabe wahlweise im Minutentakt oder per sofort per Abruf
- Lediglich 1 Port für digitalen Zeitabruf notwendig
- Arbeitet mit allen gängigen Empfängern zusammen
- Erkennt automatisch ob DCF Signal normal oder invertiert ist, werdet beides aus
- Liefert negative Flanke, die bei manchen DCF-Empfängern zum aktivieren notwendig ist (PON)
- Entlastet Controller, da Uhrzeit jederzeit decodiert abrufbar ist und keinerlei Timer oder Interrupt notwendig ist
- Betriebsspannung 2,7 bis 5,5V / ca. 3 mA
- DIP-Bauform 8 polig

TOP VIEW

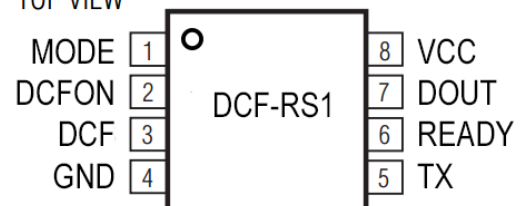


Abbildung 1: DIP-Bauform 8 polig



Bezug www.robotikhardware.de

Controllerschaltungen benötigen häufig eine genaue Uhrzeit, für diesen Zweck sind DCF-Funkuhren sehr hilfreich zumal sie sich nach einem Stromausfall automatisch wieder stellen und lebenslang genaue Zeitinformationen liefern. Die meisten Funkempfänger liefern jedoch nur ein DCF-Impulssignal welches vom Controller erst dekodiert und verifiziert werden muss. Die Verifizierung ist wichtig, denn fehlerhafte Übertragungen sind trotz Paritätsprüfung sehr oft der Fall und völlig normal. Diese Aufgabe belegt oft wichtige Ressourcen wie Speicherplatz, Timer oder Interrupt. Oft stört diese Dekodierung oder die notwendige Software-Uhr auch die Hauptaufgabe der Controller-Anwendung! Der [DCF-RS1-Chip](#) löst das Problem, er übernimmt die komplette Dekodierung & Verifizierung des DCF-Signales und arbeitet sogar noch als Uhrenbaustein falls das DCF-Signal einmal nicht empfangen werden kann. Die Zeit- und Datumsinformation kann jederzeit von dem Controller *über einen einzigen digitalen I/O Port* per RS232 (Pegel 2,7 bis 5,5V) abgerufen werden.

Pinbelegung

PIN		IN/OUT	Funktion	Zusammenfassung
1	Mode	Eingang	Der PIN legt fest ob die decodierte Uhrzeit als Text oder als binäre Zeichen ausgegeben werden sollen.	High=als Text ausgeben Low=Binär ausgeben unbeschaltet=als Text ausgeben
2	DCFON	Ausgang	Dieser Pin geht ca. 2 Sekunden nach dem die Spannung VCC/GND anliegt, auf LOW. Manche DCF-Empfänger benötigen eine LOW-Flanke zur Aktivierung, diese können dann an diesen Pin angeschlossen werden. Beispielsweise DCF-Empfänger von RobotikHardware.de, hier wird dann PON des Empfängers aufgeschaltet.	Nach 2 Sekunden immer LOW
3	DCF	Eingang	Hier wird ein handelsüblicher DCF-Empfänger angeschlossen. Es spielt keine Rolle ob das DCF Signal normal oder invertiert ist, der Schaltkreis analysiert dieses und erkennt es automatisch.	Signal vom DCF-Empfänger
4	GND		GND (Minus der Spannungsquelle)	
5	TX	Ausgang	Über diesen Pin wird im Abstand von 1 Minute die Uhrzeit digital ausgegeben. Die Uhrzeit wird im Rs232-Format (TTL-Pegel) mit 2400 Baud/8Bit/1Stopp-Bit7keien Parität ausgegeben. Der Pin 1 Mode bestimmt ob die Information als Text oder Binär ausgegeben werden.	RS232 Datenausgang
6	READY	Ausgang	Wurde die Zeitinformation korrekt empfangen als auch verifiziert, so geht dieser Pin auf LOW. Erst wenn dieser PIN das erste mal LOW Pegel besitzt, beginnt der Chip die Daten auszugeben bzw. kann die Zeitinformation abgerufen werden. Da die Zeit erst verifiziert wird, dauert es nach der Inbetriebnahme des Chips ca. 2-3 Minuten bis dies der Fall ist. Bei sehr schlechten Empfang kann dies auch länger dauern! Dieser Pin kann wahlweise vom Controller ausgewertet werden oder auch nur eine LED angeschlossen werden. Achtung: Erfolgt mindestens 15 Minuten lang keine Synchronisation mit dem DCF-Signal, dann geht <i>READY</i> wieder auf High Pegel (LED aus), der Datenabruf ist dann aber durch die eingebaute Uhr trotzdem normal möglich.	Geht auf LOW wenn Uhrzeit/Datum verifiziert wurde und geht auf High wenn mindestens 15 Minuten kein gültiger DCF-Abgleich erfolgt ist
7	DOUT	Ausgang & Eingang	Dieser Pin kann zum gezielten Abruf der Zeitinformation (Zeit und Datum) genutzt werden. Nach der Inbetriebnahme dieses Schaltkreises geht dieser Pin auf LOW ! Hat der Schaltkreis die Uhrzeit korrekt empfangen und verifiziert, dann geht dieser Pin auf High (Input mit Pullup). Der angeschlossene Controller kann nun durch ein kurzes Low-Signal die RS232 Ausgabe auf diesem Pin auslösen. Die Uhrzeit&Datum wird danach sofort im Rs232-Format (TTL-Pegel) mit 2400 Baud/8Bit/1Stopp-Bit7keine Parität auf dem gleichen Pin ausgegeben. Der Empfänger muss dadurch nicht lange auf die	Datenabruf

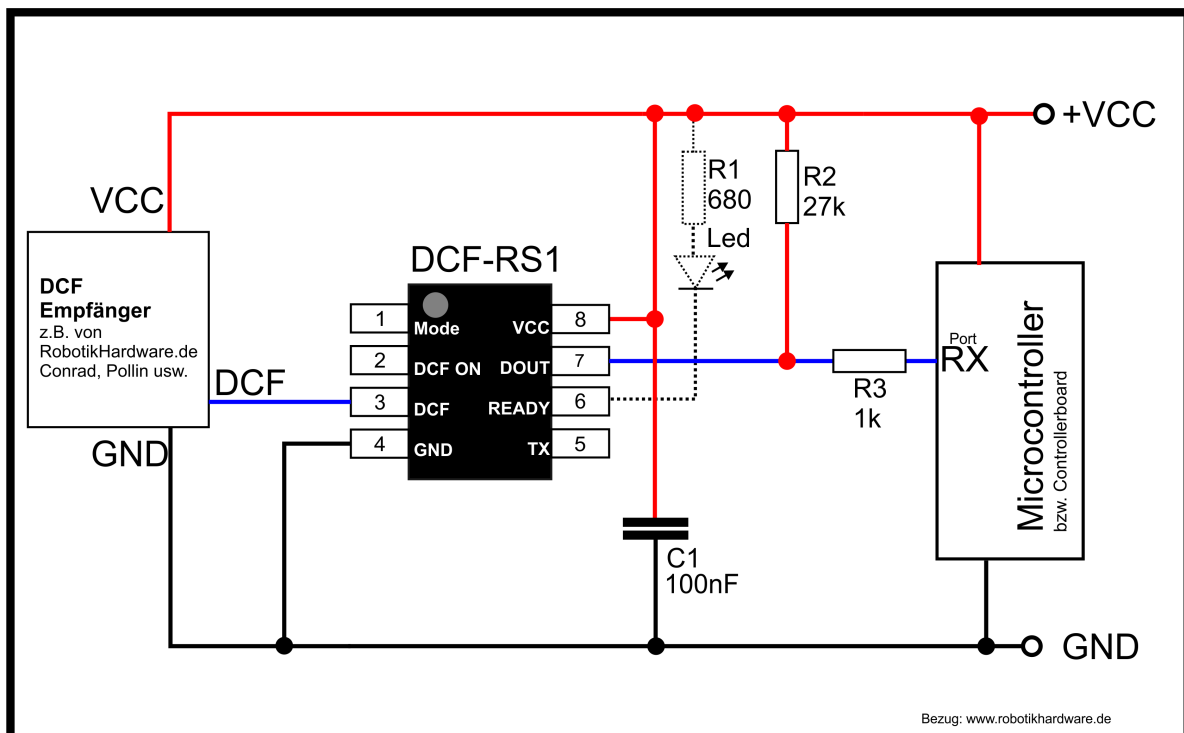
PIN		IN/OUT	Funktion	Zusammenfassung
			<p>Zeitinformation warten, da die interne Uhr (die ständig mit dem DCF-Signal synchronisiert wird) verwendet wird.</p> <p>Der Pin 1 Mode bestimmt ob die Information als Text oder Binär ausgegeben werden.</p> <p>Sobald die Ausgabe erfolgt ist, geht dieser Pin wieder auf High (Input).</p>	
8	VCC		Betriebsspannung von +2,7 bis 5,5V	+2,7 bis 5,5V

Beschaltung

Der [DCF-RS1 Chip](#) wurde so konzipiert, das die Beschaltung äußerst einfach ist. Externe Bauteile sind praktisch kaum notwendig.

Viele Pins müssen nicht belegt werden, es reicht wenn folgende Pin's angeschlossen werden:

- **Betriebsspannung** (VCC +3 bis maximal 5,5V) angelegt
- **DOUT** wird an einen beliebigen I/O Port eines Controllers angeschlossen (dieser kann dann eine Software RS232 bereitstellen) . Alternativ kann statt DOUT auch TX verwendet werden, dann erfolgt die Ausgabe aber nur periodisch im Minutentakt und kann nicht per Controller ausgelöst werden.
- **DCF** wird an einen herkömmlichen DCF-Empfänger angeschlossen werden. Gute Erfahrung haben wir mit dem Empfänger von Robotikhardware und Pollin. Noch etwas besser, da unempfindlicher, gegenüber Störungen, zeigte sich der DCF-Empfänger von Conrad-Elektronik. Da DCF-RS1 jedoch die Zeit verifiziert und eine eigene Uhr eingebaut hat, reicht es gewöhnlich wenn der DCF-Empfänger ab und zu die Zeit korrekt empfangen kann, es wird also kein besonderer Anspruch an die Empfangssicherheit erhoben.
- **READY** kann zum Anschluss einer LED benutzt werden, welche anzeigt wenn die Zeitinformation bereit steht. Der Pin muss aber nicht beschaltet werden, daher ist die LED gestrichelt eingezeichnet! *READY* geht auf Low sobald Zeit bereitsteht! Erfolgt mindestens 15 Minuten lang keine Synchronisation mit dem DCF-Signal, dann geht *READY* wieder auf High Pegel, der Datenabruf ist dann aber durch die eingebaute Uhr trotzdem möglich.



R3 zwischen DOUT und RX des Controllerboards dient zum Schutz des Controllers. Auf diese Weise kann bei falscher Programmierung ihres Controllerboards keine Beschädigung des Ports erfolgen. Zudem empfehlen wir den Kondensator C1 zwischen VCC und GND nahe am DCF-RS1 zu platzieren.

Falls Sie einen DCF-Empfänger rmit PON-Pin besitzen, welcher erst mit GND-Flanke in Betrieb geht, dann verbinden Sie einfach zusätzlich PON mit DCF ON.

Datenausgabe / Datenformat Text

Die Datenausgabe auf dem **Pin TX** als auch **DOUT** ist abhängig von der Belegung des PINs Mode. Wird dieser Pin nicht belegt oder wird er auf High gesetzt, so erfolgt die Ausgabe der Zeitinformation als Text.

Etwa 3 Minuten nach Inbetriebnahme des Schaltkreises (abhängig von der Qualität des Empfangs) geht die LED an (Ready geht auf LOW und DOUT geht auf HIGH) und die Ausgabe der Zeitinformation erfolgt im Minutentakt über den TX-PIN. Oder wie bereits geschildert durch einen LOW-Impuls auf DOUT auch auf DOUT selbst.

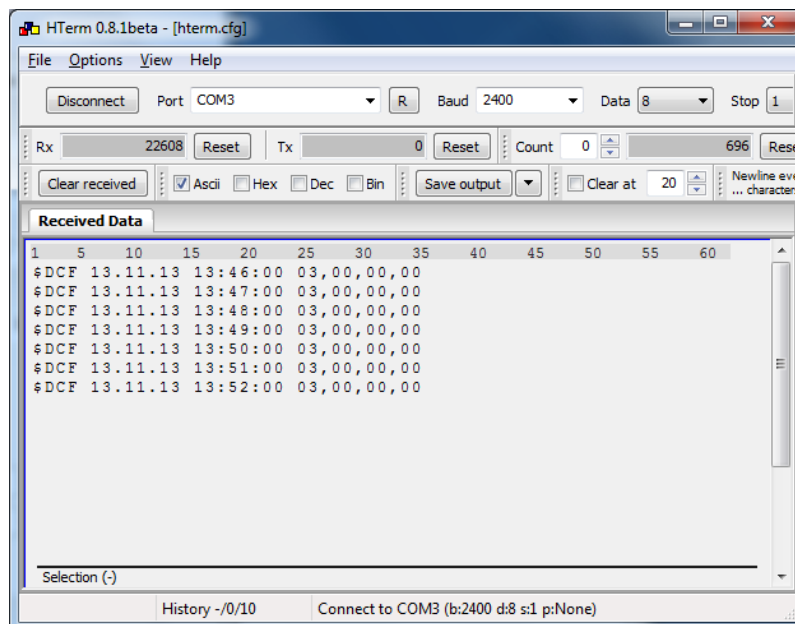
In beiden Fällen sieht die Ausgabe wie folgt aus:

```
$DCF 13.11.13 13:46:00 03,00,00,00
$DCF 13.11.13 13:47:00 03,00,00,00
$DCF 13.11.13 13:48:00 03,00,00,00
$DCF 13.11.13 13:49:00 03,00,00,00
$DCF 13.11.13 13:50:00 03,00,00,00
$DCF 13.11.13 13:51:00 03,00,00,00
$DCF 13.11.13 13:52:00 03,00,00,00
$DCF 13.11.13 13:53:00 03,00,00,00
```

...

USW.

In einem Terminalprogramm würde es in etwa so aussehen:



Im Minutenabstand empfängt also ihr Controller eine solche Zeile die mit \$DCF beginnt und mit einem Zeilenvorschub (Zeichenfolge 13/10) endet.

Die Interpretation ergibt sich eigentlich von selbst, aber hier noch mal aufgeschlüsselt in welcher Reihenfolge welche Information kommt:

Folge	Kürzel	Bedeutung
1	<i>\$DCF</i>	Kennzeichnet den Anfang der Datenzeile
2	<i>TAG.MONAT.JAHR</i>	Datum, z.B. 13.11.13
3	<i>STD:MIN:SEK</i>	Uhrzeit z.B. 13:51:00
4	<i>WOCHENTAG</i>	Wochentag von 1 bis 7, wobei 1 Montag und 7 Sonntag entspricht
5	<i>SOMMERZEIT</i>	Ist 1 wenn die Mitteleuropäische Sommerzeit aktiv ist, ansonsten 0
6	<i>ZEITUMSCHALTUNG</i>	Ist 1 wenn die Sommerzeit zur vollen Stunde umgeschaltet wird
7	<i>DCF_Synchronisation</i>	Gibt an, vor wieviel Minuten die interne Uhr des Chips mit der DCF Zeit synchronisiert wurde. Dieser Wert ist maximal 60 erreichen , auch wenn die Zeit darüber liegt.

Alle Angaben erfolgen immer formatiert mit führenden Nullen, dadurch ist die Ausgabezeile immer gleich lang, dies erleichtert die Auswertung bei Empfänger.

Der *Wochentag* geht im Wertebereich von 1 bis 7, wobei 1 dem Montag und 7 dem Sonntag entspricht!

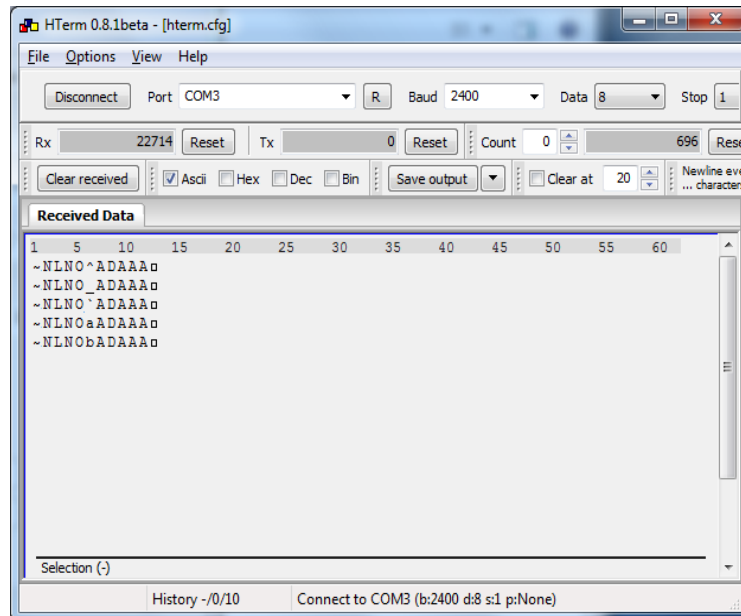
Der Wert *Zeitumschaltung* ist gewöhnlich immer 0, lediglich wenn zur vollen Stunde die Sommer/Winterzeit umgeschaltet wird, geht dieser Wert auf 1.

Der Wert *DCF_Synchronisation* liegt gewöhnlich unter 3 oft sogar bei 0. Ausnahmen sind schlechter DCF-Empfang oder starke Belastung durch viele schnell aufeinander folgende DOUT-Datenabrufe.

Datenausgabe / Datenformat Binär

Möchte man mit der Zeit im Controller rechnen oder weiterarbeiten, so kann man sich die Übernahme und das Interpretieren der DCF-Ausgabe vereinfachen indem man den MODE-Pin dauerhaft auf GND legt.

Dadurch werden alle als binäre Werte ausgegeben. Betrachtet man diese Ausgabe in einem Terminalprogramm, so sieht diese so aus:

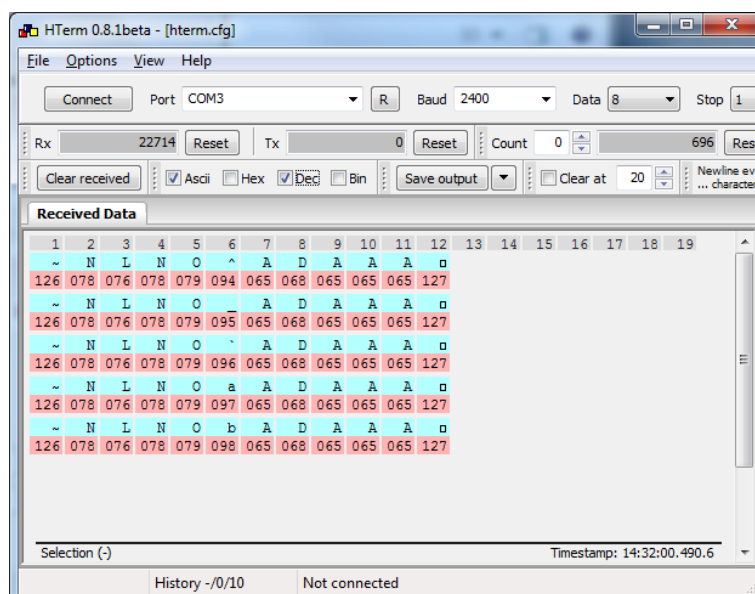


Auf den erst Blick etwas seltsam, aber das täuscht, die Interpretation ist ganz einfach. Jedes Zeichen entspricht einem binären Zahlenwert der mit 65 addiert wurde.

Der Aufbau ist wie folgt:

**126 ,TAG MONAT JAHR STD MIN SEK WOCHENTAG SOMMERZEIT
ZEITUMSCHALTUNG Synchronisation 127**

Schauen wir uns mal die binären Werte oberer Ausgabe an indem wir das Terminalprogramm auf Dezimal (rote Werte) umschalten:



Auf dem letzten Bild ist also deutlich erkennbar das nun alle Ausgabezeilen immer mit dem Zeichen 126 beginnen und mit dem Zeichen 127 enden. Dazwischen liegen die Werte addiert mit 65. Es werde also insgesamt immer **12 Bytes** + 2 Bytes für Zeilenvorschub übertragen.

Also die erste Zeile lautet: **~NLNO^ADAAA**

also Dezimal

126 078 076 078 079 094 065 068 065 065 065 127

Wir müssen also nur so rechnen:

Tag=078-65 also 13

Monat=076-65 also 11

Jahr=078-65 also 13

Stunde=079-65 also 14

Minute=094-65 also 29

Sekunde=065-65 also 0

Wochentag=068-65 also 3 Mittwoch

Sommerzeit=65-65 also 0 Derzeit ist keine Sommerzeit gültig

Zeitumschaltung=065-65 also 0 keine Sommerzeitumschaltung in den nächsten Stunde

Syncronisation=065-65 also 0 Uhrzeit wurde vor 0 Minuten (also jetzt) per Funk verifiziert

Also die abgerufene Zeit lautet :

13.11.13 14:29:00 Mittwoch

(keine Sommerzeitumschaltung in dieser Stunde und auch keine gültige Sommerzeit)

Das Einlesen dieser 12 Bytes und das zuweisen an Variablen ist also besonders einfach da nicht mit Stringfunktionen hantiert werden muss.

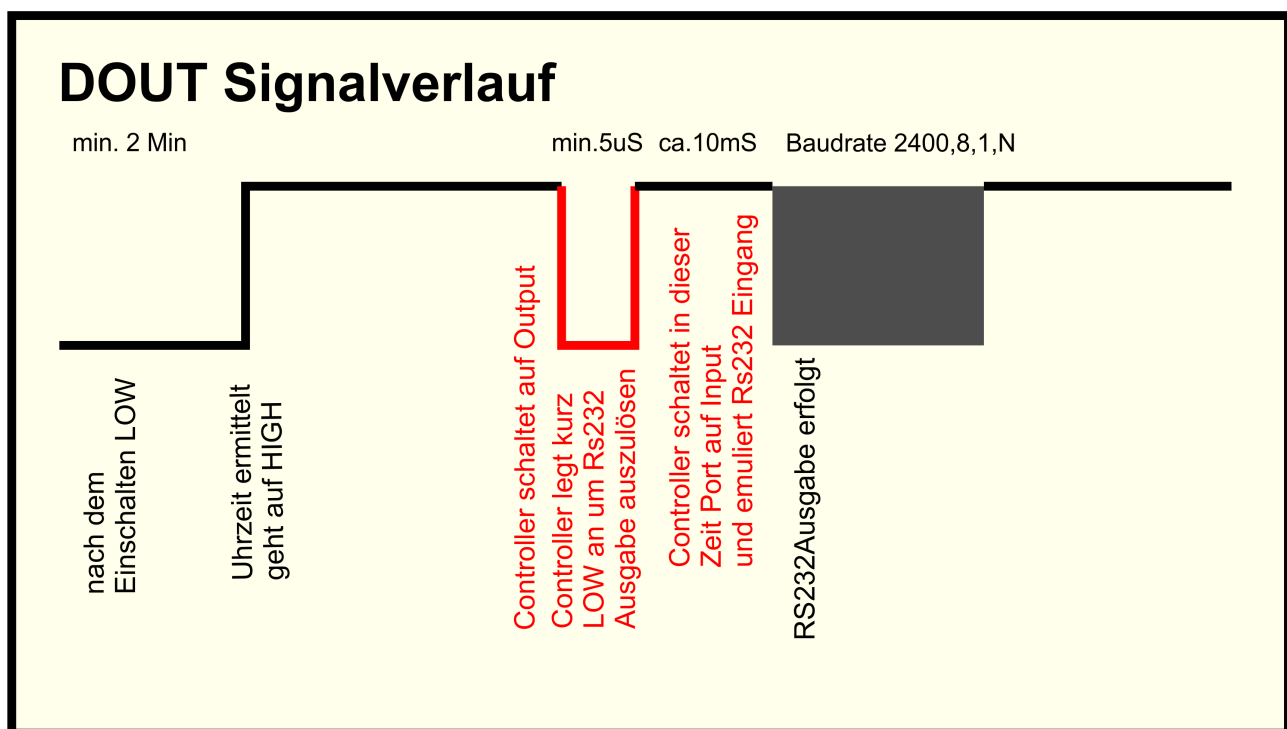
Benutzt man dann noch die den DOUT-Pin um die sofortige Zeitausgabe anzustoßen, so kann ein Controller jederzeit sehr schnell die genaue Zeit und das Datum abrufen ohne das sich eine merkliche Verzögerung ergibt.

DOUT – Signalverlauf

Damit Zeitinformationen vom Controller schnell abgerufen werden können, ist es sinnvoll den DOUT (statt TX) zu benutzen. Über diesen Pin kann der Controller durch eine kurze Low-Flanke einen Interrupt im DCFRS1 auslösen, welcher sofort die Daten auf dem gleichen Pin im RS232 Format übergibt.

Wichtig ist das der Controller vor dem LOW schalten prüft, ob das Signal überhaupt HIGH ist. Ist dies nicht der Fall, so hat der Chip noch keine gültige Uhrzeit, er muss vermutlich noch die DCF-Signale verifizieren. Ein schalten auf LOW würde in diesem Moment nichts bewirken, da der Port ja schon auf LOW ist.

Erst wenn das Signal auf High steht, kann der Controller seinen Port auf Ausgang schalten und es auf kurz auf LOW setzen. Anschließend sollte der Controller seinen Port sofort auf Eingang umschalten und eine serielle Schnittstelle per Software emulieren, denn gleich darauf sendet der DCFRS1 Chip seine Daten.



DCF-RS1 -Chip Robotikhardware.de

Das ganze kann man beliebig oft wiederholen, der Controller wird stets die aktuelle Uhrzeit und Datum vom DCFRS1 Chip erhalten. Diese Uhrzeit synchronisiert der Chip praktisch ständig sobald mal 2 Minuten lang kein Zeitabruf erfolgt. Dadurch geht die interne Uhr im Chip sehr genau. Lediglich wenn über Stunden kein DCF-Empfang möglich ist, können ganz leichte Zeitabweichungen die Folge sein.

Einfaches Beispielprogramm in Bascom Basic

Dieses Beispiel zeigt wie einfach sich die Zeitinformation aus dem Chip lesen lässt. Das Beispiel läuft so auch dem RN-AVR Universal Board, die Methode kann aber auf alle anderen Boards übertragen werden.

```
'#####  
'DCF_Decoder3.bas  
'  
'Das Programm liest die Uhrzeit aus dem digitalen DCF-Decoder aus  
'und gibt die Zeit über RS232 aus  
'Den digitalen DCF Decoder gibt es nur bei Robotikhardware.de  
'Es muss lediglich PIN PD7 mit dem DOUT Pin des Chips DCF-RS1 verbunden werden  
'Natürlich auch GND und VCC  
'  
'Wenn ein USB Modul an RN-AVR UNIVERSAL angesteckt oder eingelötet wurde  
'dann wird das Ergebnis über USB zu einem virtuellen COM Port übertragen  
'und kann auch mit allen Programmen oder Terminalprogramm empfangen werden  
'  
' (c) Frank roboternetz.de  
'Bezug: www.robotikhardware.de  
'Weitere Beispiele auf DVD oder im www.Roboternetz.de und rn-wissen.de  
'#####  
  
Declare Sub DCF_UhrzeitLesen()  
  
$programmer = 12                'MCS USB (Zeile weglassen wenn anderer Programmierer)  
$regfile = "m644pdef.dat"  
$framesize = 32  
$swstack = 32  
$hwstack = 64  
$crystal = 8000000              'Quarzfrequenz  
$baud = 9600                    'Baudrate (Übertragungsgeschwindigkeit)  
Baud = 9600  
Config Pind.6 = Output          'LED  
Led1 Alias Portd.6  
  
'Diese Variablen enthalten nach dem Abruf die Uhrzeit und Datum Informationen  
Dim uhr_sek As byte  
Dim uhr_min As byte  
Dim uhr_std As byte  
Dim uhr_tag As byte  
Dim uhr_monat As byte  
Dim uhr_jahr As byte  
Dim uhr_wochentag As byte  
Dim uhr_sommerzeit As byte  
Dim uhr_zeitumschaltung As byte  
Dim uhr_sync As byte  
dim bstart as byte 'Enthaelt Byte Zeilenanfang immer 126  
dim bende as byte 'Enthaelt Byte Zeilenanfang immer 127  
  
dim zeitgueltig as byte  
zeitgueltig=0  
wait 1  
Do  
    DCF_UhrzeitLesen  
    if zeitgueltig=1 then  
        Print "Heute haben wir den: "; uhr_tag; "."; uhr_monat; "."; uhr_jahr; " "; uhr_std; ":";  
uhr_min; ":"; uhr_sek; " Wochentag: "; uhr_wochentag  
        if uhr_sommerzeit=1 then print "Es gilt Sommerzeit!"  
        print "Die Uhr wurde von "; uhr_sync; " Minuten per Funk verglichen"  
        print  
    endif  
    Wait 3  
    toggle led1  
Loop
```

```
'Hilfsfunktion

'Diese Funktion ruft der Zeit im binären Modus ab
'und uebergibt diese den globalen Uhr Variablen
Sub DCF_UhrzeitLesen()
  Config Pind.7 = Input
  if Pind.7 =1 then 'DCF Uhrzeit bereits vorhanden?
    'Kurz auf GND setzen um DCF Decoderzeit ausgeben zu lassen
    Config PORTD.7 = output
    PORTD.7=0
    waitus 1
    Config Pind.7 = Input

    Open "comd.7:2400,8,n,1" For input As #1
    Inputbin #1,bstart,uhr_tag,uhr_monat,uhr_jahr,uhr_std,
uhr_min,uhr_sek,uhr_wochentag,uhr_sommerzeit,uhr_zeitumschaltung,uhr_sync,bende
    Close #1
    Config Pind.7 = Input
    uhr_tag=uhr_tag-65
    uhr_monat =uhr_monat-65
    uhr_jahr =uhr_jahr-65
    uhr_std=uhr_std-65
    uhr_min =uhr_min-65
    uhr_sek =uhr_sek-65
    uhr_wochentag = uhr_wochentag-65
    uhr_sommerzeit=uhr_sommerzeit-65
    uhr_zeitumschaltung=uhr_zeitumschaltung-65
    uhr_sync=uhr_sync-65
    zeitgueltig=1
  else
    Print "Zeit noch nicht verifiziert!"
    zeitgueltig=0
  endif
end sub
```

Das Programm findet man auf der Robotikhardware DVD

Hinweise

Bei Änderung des DCF-Signales von Invertiert in Normal und umgekehrt, zum Beispiel beim Wechsel des DCF-Empfängers, muss die Spannung zum Chip kurz unterbrochen werden um eine neue Analyse des Signales zu erzwingen.

Auch beim Wechsel des Ausgabemodi, also der Belegung des Pin's MODE, muss der Chip durch Neustart (Spannungsunterbrechung) darauf aufmerksam gemacht werden.

Die interne Uhr und der interne Kalender synchronisiert sich fortlaufend um höchste Genauigkeit zu erreichen. Die Synchronisation erfolgt immer dann, wenn zwischen den DOUT Datenabrufen mehr als 2 Minuten Pause ist. Wird jedoch fortlaufend die Uhrzeit benötigt, beispielsweise jede Minute ohne Unterbrechung, dann ist es empfehlenswert den TX Ausgang zu verwenden, hier wird minütlich die Uhrzeit synchronisiert. Würde ununterbrochen die Uhrzeit über DOUT abgerufen, ohne das eine Pause von 3 Minuten entsteht, so könnte sich der Schaltkreis nicht synchronisieren und es kommt zu Ungenauigkeiten der Uhrzeit. Diese Ungenauigkeit ist von Temperatur und Bauteiltoleranzen abhängig und könnte bis zu mehrere Sekunden pro Stunde betragen.

Daher synchronisiert sich DCFRS_1 in der Regel fortlaufend. Ein mehrstufiges Verifikationsverfahren sorgt in der Regel dafür das stets die korrekte Uhrzeit ausgegeben wird. Unter sehr ungünstigen Empfangsverhältnissen ist eine fehlerhafte Ausgabe der Uhrzeit nicht völlig auszuschließen, jedoch sehr selten. Sollte es doch einmal der Fall sein, so, korrigiert sich die Uhrzeit natürlich bei der nächsten Synchronisation von selbst.

Bei der Implementation des Schaltkreises in eigene Schaltungen, empfehlen wir für den Schaltkreis eine 8 polige IC Fassung. Dadurch können auch andere Pin-kompatible Schaltkreise, die eventuell noch angeboten werden, bei Bedarf im Austausch eingesetzt werden.

Die Beschriftung des Schaltkreises DCFRS_1 kann von der Abbildung in dieser Doku abweichen. Sie kann entweder vollständig fehlen oder andersartig lauten, die Funktion ist aber immer identisch.

Bezug des Chips oder passenden DCF-Empfängers:

derzeit nur über den Shop www.RobotikHardware.de

Wichtige Hinweise und Haftungsausschluss



Lesen Sie bitte diese Gebrauchsanleitung, bevor sie dieses Modul / Bauteil aufbauen bzw. in Betrieb nehmen.

Bestimmungsgemäße Verwendung: Dieses Modul ist nur Entwicklungsaufgaben, Forschung, Lehrzwecke und Unterricht und Prototypenbau konzipiert! Für die Einhaltung der technischen Vorschriften sind sie selbst verantwortlich.

Sicherheitshinweise

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden, insbesondere VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860.

Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, dass die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in den im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.

Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden ist, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muss das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist. Bei Einsatz von Bauelementen oder Baugruppen muss stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden. Wenn aus einer vorliegenden Beschreibung für den nicht gewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil oder eine Baugruppe gelten, wie eine externe Beschaltung durchzuführen ist oder welche externen Bauteile oder Zusatzgeräte angeschlossen werden dürfen und welche Anschlusswerte diese externen Komponenten haben dürfen, so muss stets ein Fachmann um Auskunft ersucht werden. Es ist vor der Inbetriebnahme eines Gerätes generell zu prüfen, ob dieses Gerät oder Baugruppe grundsätzlich für den Anwendungsfall, für den es verwendet werden soll, geeignet ist!

Im Zweifelsfalle sind unbedingt Rückfragen bei Fachleuten, Sachverständigen oder den Herstellern der verwendeten Baugruppen notwendig!

Bitte beachten Sie, dass Bedien- und Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen. Bausätze sollten bei Nichtfunktion mit einer genauen Fehlerbeschreibung (Angabe dessen, was nicht funktioniert...denn nur eine exakte Fehlerbeschreibung ermöglicht eine einwandfreie jedoch kostenpflichtige Reparatur!) und der zugehörigen Bauanleitung sowie ohne Gehäuse zurückgesandt werden. Zeitaufwendige Montagen oder Demontagen von Gehäusen müssen wir aus verständlichen Gründen zusätzlich berechnen. Bereits aufgebaute Bausätze sind vom Umtausch ausgeschlossen. Bei Installationen und beim Umgang mit Netzspannung sind unbedingt die VDE-Vorschriften zu beachten. Geräte, die an einer Spannung über 35 V betrieben werden, dürfen nur vom Fachmann angeschlossen werden. In jedem Fall ist zu prüfen, ob der Bausatz oder die Platine für den jeweiligen Anwendungsfall und Einsatzort geeignet ist bzw. eingesetzt werden kann.

Derjenige, der einen Bausatz fertigstellt oder eine Baugruppe durch Erweiterung bzw. Gehäuseeinbau betriebsbereit macht, gilt nach DIN VDE 0869 als Hersteller und ist verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch seinen Namen und Anschrift anzugeben. Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und/und Anschlussfehler außerhalb unseres Einflusses liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.

Ihr Shop www.Robotikhardware.de