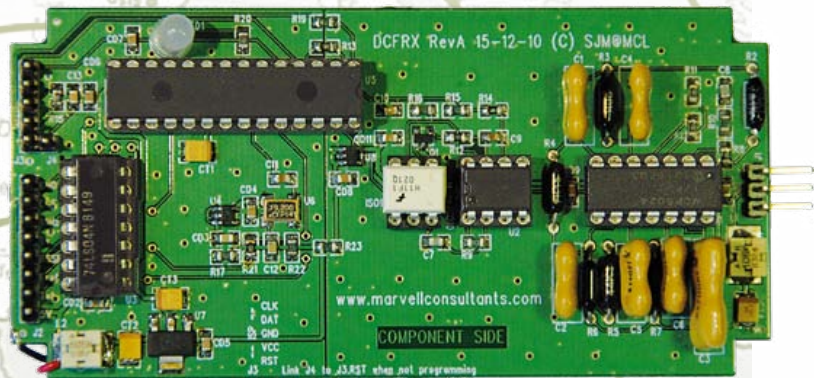


Ultra-präziser DSP-basierter DCF77-Zeitsignal-Empfänger

Besser als kommerzielle DCF-Empfänger...

Um die höchstmögliche Genauigkeit aus dem Zeitsignal des Mainflinger DCF77-Senders herauszukitzeln, nutzt dieses Projekt DSP-Algorithmen in einem preiswerten dsPIC33-Controller. So wird das amplituden- und phasenmodulierte Signal gefiltert und demoduliert. Das Ergebnis ist ein sehr stabiles 10-Hz-Taktsignal.



Von Steve Marchant (UK)

Kommerzielle DCF77-Receiver/Demodulatoren, zum Beispiel von Galleon Systems oder Conrad Elektronik, funktionieren zwar sehr gut und produzieren zuverlässige Zeitsignal-Impulse, doch die exakte Position dieser Impulse unterliegt im Vergleich mit einem „richtigen“ 1-Hz-Takt doch einem erheblichen Jitter. Über 10 ms Ungenauigkeit sind da nicht selten. Der Grund hierfür liegt im typischen Einsatz von Quarzen zur Extraktion der Trägerfrequenz. Ihre besonders schmale Bandbreite macht sie scheinbar ideal für diese Aufgabe, hat aber Timing-Probleme in der Demodulationsstufe zur Folge. Die preiswerten kommerziellen

DCF77-Empfänger stellen außerdem am Ausgang weder einen stabilen, mit dem Trägersignal gekoppelten Takt zur Verfügung, noch decodieren sie die phasenmodulierte Zufallszahlenfolge, die ein sehr viel genaueres Timing ermöglicht und gerade bei geringerer Feldstärke deutlich zuverlässiger ist. Mit dem hier beschriebenen Verfahren kann man hingegen ein sehr präzises Taktsignal generieren.

Hardware

Der Empfänger besteht aus zwei Teilen: Einer Platine mit dem eigentlichen Empfänger und einer aktiven Antenne. Auf diese Weise kann man die Antenne via Koaxkabel weit vom Empfänger

Leserprojekte sind Beiträge von Elektor-Lesern für experimentelle Zwecke oder zur Anregung für andere Leser. Die in dieser Rubrik vorgestellten Schaltungen wurden vom Elektor-Labor nicht auf Reproduzierbarkeit und Funktion getestet.

entfernt anordnen, wodurch der Empfang deutlich weniger von Störstrahlungen anderer Geräte beeinflusst wird. Die Antenne besteht aus einem fertigen Ferritstab mit Spule und einem parallel geschalteten Kondensator, der eine Abstimmung auf 77,5 kHz erlaubt. Das Antennensignal wird von zwei phantomgespeisten Opamps verstärkt (siehe Bild 1), wodurch das Verbindungskabel zum Empfänger recht lang ausfallen kann. Die erste Opamp-Stufe des Empfängers (Bild 2) sorgt für eine weitere Signalverstärkung und treibt eine Kette von drei weiteren

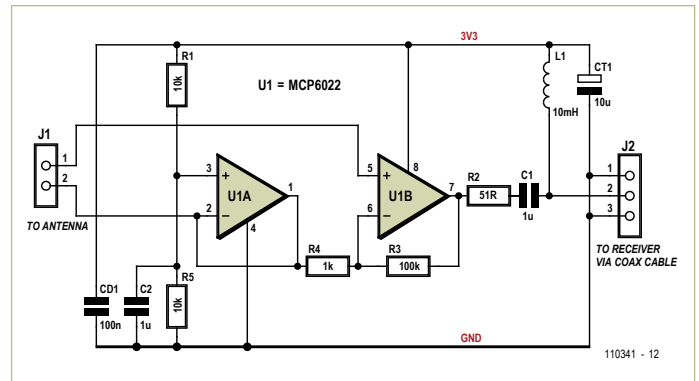


Bild 1. Schaltung der aktiven Antenne für DCF77. Die Elektronik wird phantomgespeist.

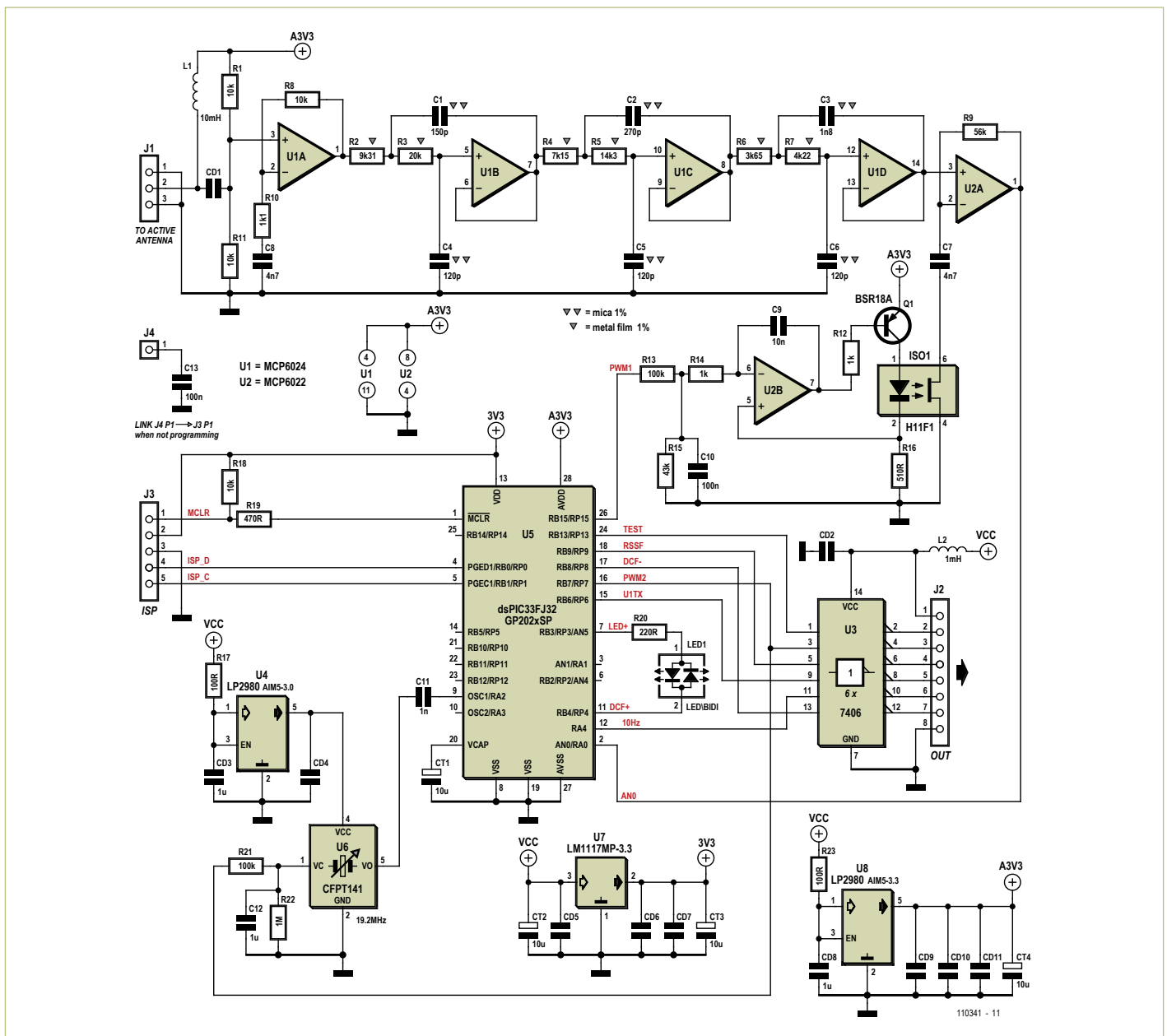


Bild 2. Die Schaltung des eigentlichen DCF-Empfängers enthält analoge und digitale Schaltungsteile.

| Anschlüsse | |
|--|--|
| J1: Antennenanschluss (3-polig) | |
| J1.1 | GND |
| J1.2 | HF-Signal |
| J1.3 | GND |
| J2: Betriebsspannung und Ausgangssignale (8-polig) | |
| J2.1 | 5 V / 100 mA |
| J2.2 | TEST-Ausgang, ändert sich, wenn die Hauptschleife zu viel Zeit braucht |
| J2.3 | VCXO-PWM-Signal für den Taktgenerator |
| J2.4 | unbenutzt |
| J2.5 | Seriellles TTL-Signal mit 9600 Baud |
| J2.6 | 10-Hz-Referenz-Signal |
| J2.7 | DCF-Impuls-Ausgang, führende negative Flanke |
| J2.8 | GND |
| <i>Hinweis: Alle Ausgänge sind bei Verwendung eines 74LS06 vom Typ Open-Collector.</i> | |
| <i>Wird für IC3 ein 74HCT04 eingesetzt, erhält man Totem-Pole-Ausgänge.</i> | |
| J3: PIC-ISP (5-polig) | |
| J3.1 | MCLR |
| J3.2 | 3,3 V |
| J3.3 | GND |
| J3.4 | ISP Data / GPS-Referenz-Impuls-Eingang |
| J3.5 | ISP Clock |

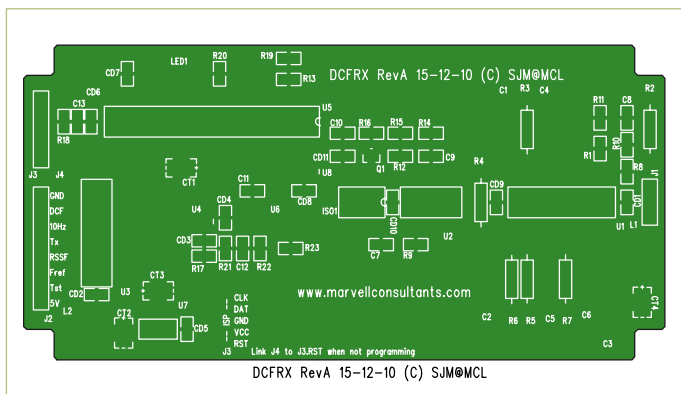


Bild 3. Bestückungsplan des Empfängers. Die Layout-Dateien sind kostenlos von [1] erhältlich.

Stufen, die zusammen als Tiefpass sechster Ordnung (Grenzfrequenz 78 KHz bei -1 dB und Dämpfung von -50 dB bei 232,5 kHz) die Anti-Aliasing-Filterung übernehmen. Nachfolgend wird das Signal mit einer Sample-Rate von 310 kHz ($f_s = 4f_c$) abgetastet und dann von einem mittels DSP realisierten 77,5-kHz-Kerbfiler behandelt, sodass man sich erst um die dritte Oberwelle bei 232,5 kHz kümmern muss. Die analogen Filter wurden mit Hilfe der kostenlosen FilterLab-Software von Microchip erstellt. Die indizierten Bauteile sollen eine Toleranz von 1% aufweisen. Der fünfte Opamp ist als einstellbarer Wechselspannungsverstärker geschaltet. Alle Opamps von IC1a bis einschließlich IC2a haben eine Gleichspannungsverstärkung von 1, weshalb sich die mit R1 und R11 am Anfang eingestellte halbe Betriebsspannung bis zum Ausgang von IC2a durchzieht. Am Ende ist direkt der Mikrocontroller mittels ADC-Eingang angeschlossen.

Die Verstärkung des Antennensignals wird von der Software automatisch (AGC) so eingestellt, dass am ADC-Eingang ein Pegel von etwa $3 V_{SS}$ anliegt. Nach Demodulation durch die Software wird das DCF-Zeit-Signal an einen Open-Collector-Ausgang gelegt. Die einstellbare Verstärkerstufe um IC2a basiert auf dem speziellen Optokoppler H11F1 mit bidirektionalem FET, wodurch sich ein optisch gesteuerter Widerstand im Rückkopplungszweig von IC2a ergibt. Je heller die LED, desto niedriger der Kanalwiderstand des FETs und umso höher die Verstärkung. Die LED des Optokopplers wird vom U/I-Konverter um IC2b und T1 angesteuert. Am Eingang von IC2b liegt die Spannung eines PWM-Ausgangs des Controllers. Über dessen Tastverhältnis kann die Software die Verstärkung einstellen. Um einen an die DCF-Trägerfrequenz gekoppelten Takt für den Mikrocontroller zu erzeugen, wurde als Taktquelle mit IC6 ein spannungsgesteuerter Quarzoszillator verwendet. Auch hier wird die Regelung über das Tastverhältnis eines PWM-Ausgangs realisiert.

Software

Ein dsPIC33 enthält einen 16-bit-Rechenkern mit gut 40 MIPS und neben RAM, Flash-Speicher und einiger Peripherie auch einen 12-bit-ADC mit einer Sample-Rate von bis zu 500 kS/s, der das HF-Signal problemlos mit 310 kHz Samplefrequenz abtasten kann. Der ADC-Teil des Controllers ist sehr gut konstruiert, denn sein Buffer enthält zwei Pages mit bis zu acht Samples. Er schaltet zwischen diesen automatisch um und setzt entsprechende Flags, um der Software mitzuteilen, dass neue Samples eingetroffen sind. Die Software bearbeitet vier Samples auf einmal in einer Endlosschleife, bei der jeder Durchlauf in weniger als $13 \mu s$ abgeschlossen sein muss – der Periodendauer des DCF77-Signals. Dazu reichen die verfügbaren 40 MIPS aber gerade nicht ganz aus, weshalb der Controller hier übertaktet wird.

Nach dem Einlesen der vier Samples (= 1 Periode) besteht die erste Aufgabe in der Kreuzkorrelation, der Multiplikation mit dem Sinus und Cosinus der zu extrahierenden Frequenz f_c . Digital besteht das lediglich aus $\{0, 1, 0, -1\}$ und $\{1, 0, -1, 0\}$ – die Multiplikationen sind also eher trivial. Alle vier Sinus- und Cosinus-Ergebnisse werden jeweils addiert und damit zwei Ring-Puffer mit einer Länge von je 120 Elementen gespeist. Die Gesamtsummen der beiden Ring-Puffer werden so vektoriell addiert:

$$\sqrt{(\sin^2 + \cos^2)}$$

und ergeben ein Amplituden-Resultat. Bei diesem gefilterten Amplitudensignal wird in den nächsten zwei Perioden Minimum und Maximum bestimmt. Von diesen Werten wird jeweils ein Schwellwert abgeleitet, mit dessen Hilfe eine Umwandlung in ein Binärsignal erfolgen kann. Dieses binäre Signal ist dann schon das rohe Zeitsignal und könnte so als Taktsignal weiterverwendet werden. Doch das geht noch besser.

Zuvor gilt es weitere Funktionsgruppen zu beleuchten. Zunächst der AGC-Teil: Die Software überprüft die vier ADC-Samples darauf, ob einige die untere oder obere Schwelle für den ADC-Eingang über- oder unterschreiten. Die Software ist bestrebt, den Pegel

Debugging

Der Autor hat seinen Empfänger im rund 850 km vom DCF77-Sender entfernt gelegenen West Yorkshire (mitten in England) entwickelt. Auf Grund der dadurch bedingten geringen Signalstärke war eine hohe Verstärkung erforderlich. Wenn man deutlich näher an Mainflingen wohnt, könnte weniger Verstärkung besser sein, was sich durch einen geringeren Wert für R8 oder einen größeren Wert für R10 erreichen lässt. Bei geringer Feldstärke spielt auch die Antennenposition und ihre Ausrichtung eine entscheidende Rolle. Die breite Seite des Ferrit-Stabs sollte in Richtung Mainflingen (50° 2' N, 9° 2' O) ausgerichtet werden. Außerdem sollte man die Antenne von Interferenzquellen wie Computern, Monitoren und Schaltnetzteilen fernhalten. Besonders „elektronische“ Niedervolttrafos machen sich

hier negativ bemerkbar. Sogar ein angeschlossener ICD2-Debugger beeinträchtigt die Signalqualität. Der serielle Debug-Ausgang liefert nichts, solange der Phasendecoder nicht eingerastet ist. Beim Einschalten passiert das dann, wenn die Takt-PLL einrastet und die LED von rot auf grün wechselt, was mindestens 10 s dauert. Dann muss ein brauchbares AM-Signal anliegen. An der LED kann man gut sehen, ob das Antennensignal verrauscht ist. Wenn die seriellen Daten fließen, stehen jede Menge Debug-Informationen zur Verfügung. Für einen vernünftigen Empfang sollte die Signalstärke >40 betragen. Dabei sollte die AGC nicht gegen die Extremwerte 0 oder 1.023 laufen. Außerdem sollte die Steuerspannung für den System-Taktoszillator nicht stark fluktuieren oder gegen die Extremwerte laufen.

am oberen Limit zu halten. Wenn keine oder nur wenige Samples die obere Schwelle überschreiten, dann wird die Verstärkung ein Stück erhöht. Wenn es zu viele Samples sind, wird sie entsprechend verringert. Da die Verstärkung letztlich vom Tastverhältnis eines PWM-Ausgangs abhängt, ändert die Software hierzu schlicht das zugehörige PWM-Register.

Ähnlich funktioniert auch der abstimmbare Taktgenerator: Ein fest an die Trägerfrequenz gekoppelter Takt (PLL) vereinfacht die Demodulation von phasenmodulierten Signalen nämlich deutlich. Mit Hilfe der Sinus- und Cosinus-Werte der Kreuzkorrelation ist es recht einfach, die Phase des Signals relativ zur Abtastfrequenz zu bestimmen. Und wenn man erst einen Wert für die Phasendifferenz hat, dann ist es einfach, eine PLL zu realisieren. Sowohl Sinus- als auch Cosinus-Daten dividiert durch die Amplitude sind ein Maß für die Phase. Dieser Wert wird direkt und über ein Software-Filter zur Steuerung eines weiteren PWM-Ausgangs benutzt. Nach einem RC-Tiefpass kommt das Signal auf den Steuerspannungseingang von IC6, womit die Regelschleife geschlossen ist. Der Takt steht nun in einem festen Verhältnis zur Trägerfrequenz, weshalb davon ein stabiles 10-Hz-Taktsignal zur beliebigen Verwendung abgeleitet werden kann. Die Software überwacht nun, ob die PLL „in lock“ ist. Andernfalls wird der 10-Hz-Takt solange ausgetastet, bis die PLL wieder einrastet.

Nun kann man endlich die phasenmodulierte Zufallszahlenfolge des DCF-Signals angehen. So eine Folge besteht aus 512 bit, von denen jedes für 120 Trägerperioden dazu verwendet wird, die Phase um $\pm 13^\circ$ zu verschieben. Die Modulation beginnt 200 ms nach der ansteigenden Flanke des AM-Zeit-Codes und dauert annähernd 793 ms. Da gleich viele Nullen und Einsen vorhanden sind, bleibt die Phase des Trägers „über alles“ von der Modulation unbeeindruckt. Die Software hat schon ein Maß für die Phase (stabilisiert durch die PLL), weshalb nun klar wird, weshalb hier eine Puffer-Größe von 120 Einträgen gewählt wurde. Der Code enthält noch einen weiteren Kreuz-Korrelator. Eine im Befehlsspeicher abgelegte Referenz-Bitsequenz wird mit der Träger-Phasemessung alle 120 Perioden multipliziert und über 512 bit aufaddiert. Das Resultat hiervon ist ein Maß für die Korrelation zwischen der Referenz und der übermittelten Bitfolge. Sie hängt stark davon ab, wie gut die beiden Sequenzen übereinstimmen.

Bei hoher Übereinstimmung der Referenz-Sequenzen ergibt sich ein gutes Timing der Sekunden-Impulse. Tatsächlich ist im DCF-Signal ein Bit der Zeit-Code-Daten entweder normal oder als Komplement in der Sequenz codiert. Als Folge erhält man abhängig vom codierten Bit entweder eine positive oder negative Korrelation.

Doch wie erreicht man eine möglichst gute Übereinstimmung? In einer perfekten Welt mit unbegrenzter Rechenleistung könnte man alle Phasendaten für zum Beispiel eine Sekunde zwischenspeichern (77.500 Werte) und nach der höchsten Korrelation zur Referenz suchen, ausgehend von unterschiedlichen Startpositionen innerhalb der Daten. In der realen Welt reicht die Rechenleistung nur dazu aus, einen Satz von Korrelationen pro Sekunde zu berechnen. Von daher muss man einen Startpunkt festlegen und damit leben. Außerdem benötigt man eine Methode, um zu bestimmen, ob die Position in Ordnung geht oder nicht, sodass man den Startpunkt für die nächste Sekunde verändern kann.

Zur Gewinnung dieser Information wird eine weitere Kreuzkorrelation gerechnet, die um eine halbe Bit-Zeit versetzt ist und eine differentielle Bit-Sequenz nutzt, die aus der Eingangssequenz abgeleitet wurde. Man erhält dann ein Signal, das nach Korrektur bezüglich der Polarität der encodierten Daten dann Null wird, wenn die Übereinstimmung perfekt ist. Der Wert wird bei zu frühem Startpunkt negativ und bei zu spätem Startpunkt positiv. Mit diesem Verfahren wird permanent der nächste Startpunkt optimiert und eine gute Übereinstimmung erzielt. Ist Letzteres erreicht, dann können die Sekunden-Impulse mit einer sehr großen Präzision der führenden Flanke generiert werden. Ein Problem ist die Wahl des initialen Startpunkts, da die Korrelationsergebnisse nur für einen Wertebereich innerhalb der Zeit von ± 1 bit beziehungsweise 120 Träger-Perioden oder 1,5 ms entfernt vom Optimum funktionieren. Für den initialen Startpunkt muss man sich daher auf das amplitudenmodulierte Signal verlassen. Die Software sucht nach einer sauberen negativen Flanke und setzt diese als Startpunkt. Bei gutem Empfang ist das kein Problem. Doch wenn der Phasendecoder nicht einrastet, muss man es nochmals versuchen. Hat er einmal eingerastet, dann ist diese Kopplung sehr robust und bleibt auch erhalten, wenn der Pegel so gering ist, dass sich keine amplitudenmodulierten Daten mehr extrahieren lassen. Source- und Object-Code-Dateien für dieses Projekt sind wie immer kostenlos von [1] erhältlich.

Serieller Datenausgang

Die Software stellt die phasenmodulierten Zeitsignal-Daten und einige zusätzliche Debug-Informationen an einer seriellen Schnittstelle mit 9600 Baud zur Verfügung. Jede Sekunde wird ein Datenpaket ausgegeben, wobei das Start-Bit des ersten Zeichens mit dem Sekunden-Impuls synchronisiert ist. Der serielle Ausgang liefert nur Daten, wenn der Phasen-Decoder läuft. Das Datenpaket kann eines von zwei Formaten haben:

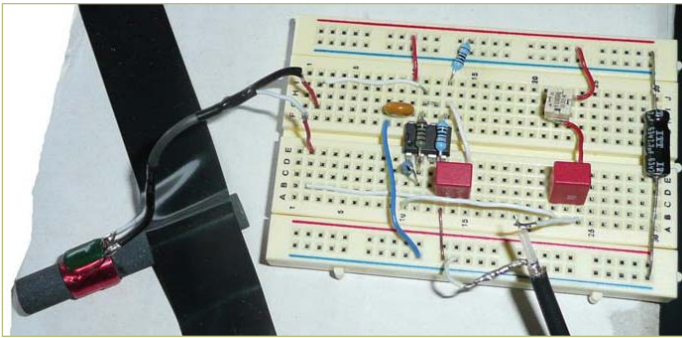


Bild 4. Die aktive Antenne wurde auf einem Steckbrett aufgebaut.

„!LDsseeeddddpppaayy“ oder
 „:LDsseeeddddpppaayyoocrrrrzzzz“

wobei

! = Übergang des Phasen-Decoders zwischen eingerastet und nicht eingerastet.

: = der Phasen-Decoder ist stabil.

L = Einrast-Status der Takt-PLL (0 oder 1).

D = Ausgang des Phasen-Decoders; Zeitsignal-Bit der vorherigen Sekunde.

ss = Signalstärke; <40 = schlecht, >100 = sehr gut, Maximum = 160.

eeee = Ausgang des Pseudo-Zufalls-Phasen-Korrelators.*

ddd = differentieller Ausgang des Pseudo-Zufalls-Phasen-Korrelators.*

ppp = gefilterter PWM-Wert des VCXO (PLL-Steuerspannung 0...1.023).

aaa = aktueller AGC-PWM-Wert (Steuerspannung 0...1.023).

yy = Signalamplitude am Ende der Phasenmodulation (0...160).

oo = Zustand des Phasen-Korrelators; -1 = Fehler, 0 = schlecht, Maximum = 60.*

cc = letzte Anpassung des Startpunkts des Phasen-Korrelators.*

rrrr = absolute Phase des GPS-Referenz-Eingangs, falls vorhanden (0...77.499).

zzzz = absolute Phase der Sekunden-Impulse (0...77.499).

Alle Felder werden hexadezimal angegeben. * bedeutet, dass der Wert negativ sein kann; bei gesetztem MSB gilt der Komplementärwert; man addiere eine 1 und interpretiere den Wert als negative Zahl.

Der DCF-Ausgang der Platine entspricht dem eines gewöhnlichen AM-Empfängers, allerdings mit sehr viel höherer Präzision der führenden Impulsflanke. Der optimierte Impuls dauert 50 ms –

danach übernimmt die Kurvenform des AM-decodierten Impulses bis 200 ms erreicht wurden. In der Zeit von 50...200 ms können bei schlechtem Empfang also durchaus Störungen auftreten. Das gilt auch für den Impuls der 59. Sekunde, der ja im AM-Signal fehlt. Der 50-ms-Impuls aus dem Phasen-Signal ist allerdings vorhanden. Wenn der Phasen-Decoder für Pseudo-Zufallszahlen nicht eingerastet ist, erscheinen keine Impulse am Ausgang. Interessant ist auch, dass die phasencodierten Daten für die Sekunden 59...9 zur Zeit alle als 1 codiert sind (im Gegensatz zu den AM-Daten), was für bestimmte Zwecke ausgenutzt werden kann.

Falls richtig angeschlossen, zeigt die LED bei eingerasteter Takt-PLL die AM-Impulse in grün an. Bei nicht eingerasteter PLL oder beim Start blinkt die LED rot.

Aufbau

Die Platine des eigentlichen Empfängers ist doppelseitig und mit einer Mischung aus bedrahteten und SMD-Bauteilen bestückt. Bild 3 zeigt den Bestückungsplan. Die Layout-Dateien des Autors können kostenlos von [1] herunter geladen werden. Bild 4 zeigt den experimentellen Aufbau der aktiven Antenne. Hierfür gibt es (noch) keine Platine. Wenn Sie einen Entwurf hierfür zur Verfügung stellen wollen, können Sie das gerne im Elektor-Forum tun.

Signalqualität

Beim DCF77-Signal gibt es zwei Ausbreitungswege: Die Bodenwelle ergibt im Radius von bis zu 1.000 km rund um Mainflingen brauchbare Pegel. Die von der Ionosphäre reflektierten Raumwellen variieren je nach Tages- und Jahreszeit erheblich. Normalerweise ist der Pegel nachts und im Winter stärker. Unglücklicherweise ist aufgrund der unterschiedlichen Laufzeiten auch die Phase relativ zur Bodenwelle variabel, was zu Fading mit Aussetzern und zu Phasenverschiebungen führt. Von daher wird es mit zunehmender Entfernung von Mainflingen immer schwieriger, nachts ein stabiles Signal zu bekommen. Tagsüber ist es zwischen 10:00 und 14:00 Uhr im Winter erfahrungsgemäß meistens möglich, ein Signal mit einer Abweichung von weniger als $\pm 250 \mu\text{s}$ zu erhalten.

Genauigkeit

Zur Überprüfung der Genauigkeit des Mustersaufbaus wurde ein Eingang für einen GPS-Timing-Impuls (positiver Impuls mit 3,3 V Amplitude an J3.4) vorgesehen. Die Phase dieses Impulses und das

Weitere Software-Entwicklung

Wer die Signallaufzeit kompensieren oder gar tiefer in den Code eintauchen möchte, der kann sich gerne den Source-Code herunterladen und damit experimentieren. Doch ein Anliegen hat der Autor: Bitte machen Sie eventuelle Bug-Fixes oder Verbesserungen für alle verfügbar! An Werkzeugen braucht man Microchips MPLab als IDE und den C30-Compiler, welche beide kostenlos von der C30-Webseite verfügbar sind.

Zur Übertragung der Firmware in den dsPIC33 benötigt man einen passenden ISP-Programmer. Die preiswerteste von MPLab unterstützte Variante ist PICKit3. Der Autor setzte einen ICD2 ein. Zum Anschluss an den fünfpoligen Port J3 muss man sich dann noch ein passendes Kabel anfertigen. Nochmals die Warnung: Bei angeschlossenem Programmer verschlechtert sich das DCF77-Signal!

der decodierten Sekunden-Marke sind Teil des seriellen Debug-Datenstroms. Beim Autor ergibt sich an einem guten Tag eine absolute GPS-Phase von typisch 32.547, die Phase des decodierten Sekunden-impulses liegt bei 32.768 (siehe Oszillogramm in Bild 5). Die Einheiten sind Träger-Perioden, sodass 221 Perioden gut mit der Signal-Laufzeit von 858 km aus Mainflingen übereinstimmen - rechnerisch wären dies nämlich 2,86 ms oder 221,8 Perioden. Langzeitdaten von GPS-basierten Messungen des Mustergeräts sind auf der Webseite des Autors zu sehen.

Für eine optimale absolute Genauigkeit ist eine Kalibrierung entweder des Taktgenerators oder des Empfängers erforderlich, um die Signallaufzeit der Bodenwelle auszugleichen. Aus diesem Grund ist ein Link zu www.daftlogic.com angegeben. Auf dieser Webseite kann man aus der Entfernung zu Mainflingen die zugehörige Signallaufzeit berechnen, wobei die Lichtgeschwindigkeit mit 300 km/ms oder $3,33 \mu\text{s}/\text{km}$ oder 0,2585 Trägerperioden/km angegeben wird. Die resultierende Zahl an Trägerperioden rundet man auf die nächste Integer-Zahl und gibt sie in den Source-Code ein, wenn sie kleiner als 378 ausfällt (siehe den Abschnitt „Propagation delay adjustment“ im Code). Größere Werte als 377 würden die Erzeugung von Sekunden-Impulsen behindern. Wenn die geänderte Firmware neu kompiliert ist, erreicht man damit am eigenen Wohnort eine absolute Präzision von $\pm 250 \mu\text{s}$ am Tag – siehe Bild 6. Die Standard-Version der Firmware wurde ohne diese Korrektur kompiliert.

(110341)

Kontakt zum Autor

steve@marvellconsultants.com

Projekt-Webseite

[1] www.elektor.de/110341

Weitere Webseiten

1. Webseite des Autors: www.marvellconsultants.com/DCF
2. Die Software für das digitale Sperrfilter mit Kreuzkorrelation ist inspiriert von „THE SCIENTIST & ENGINEER'S GUIDE TO DIGITAL SIGNAL PROCESSING“ von Steven W. Smith, erhältlich von www.analog.com: <http://bit.ly/faQrb2>. Das Original findet sich unter: www.analog.com/en/embedded-processing-dsp/adsp-21xx/processors/design-handbooks/scientist_engineers_guide/resources/fca.html
3. Entwickeln in C30 für dsPIC-Controller: www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=81
4. In-System-Programming für PIC-Mikrocontroller: www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2519¶m=en534451&page=wwwdevMPLABEmulatorDebuggers

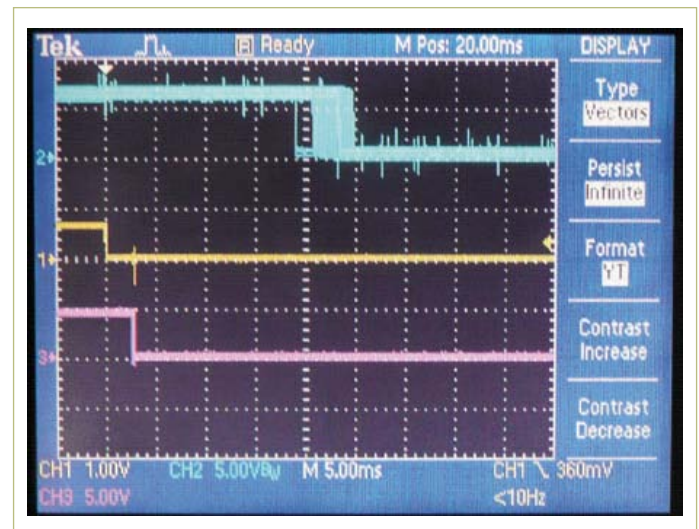


Bild 5. Ein Oszillogramm zeigt die Unterschiede: Oben (blau) = Galleon MSF Rxvr; mittlere Kurve (gelb) = GPS-Referenz; unten (pink) = DCF-Rx. Zwei Stunden = 7200 s Aufzeichnung von 14:30 bis 16:30 am 24.05.2011. Zuhause bei HX7 UK beträgt die Signallaufzeit 2,86 ms und die MSF-Verzögerung 0,3 ms.

5. Software FilterLab von Microchip: www.microchip.com/filterlab
6. DCF77-Details: www.ptb.de/en/org/4/44/442/dcf77_1_e.htm
http://hw-server.com/design/pcf8574/komponenta_pcf8574.html
<http://elektronik-engineering.ch/microchip/datasheets/dcf77/dcf77.html>
7. Google-Maps-Entfernungsrechner: www.daftlogic.com/projects-google-maps-distance-calculator.htm

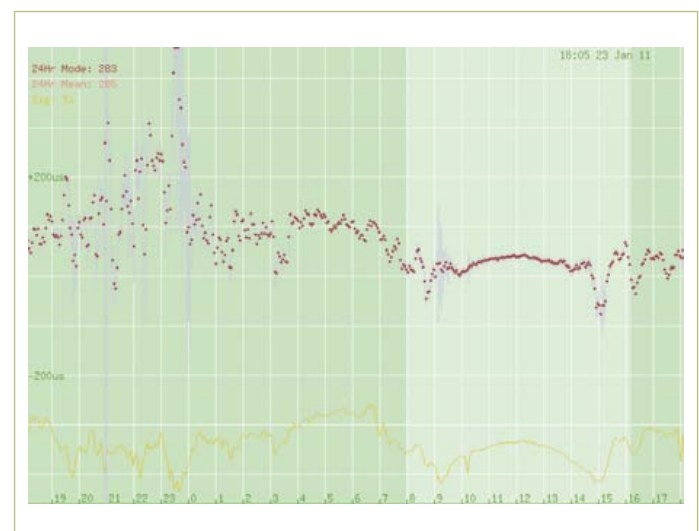


Bild 6. Kurzzeitüberwachung des DCF77-Empfängers. Die roten Punkte zeigen die mittlere Timing-Genauigkeit während einer Periode von 200 s, während die grauen Linien die Standard-Abweichung während dieser Periode angeben. Die vertikale Achse ist $100 \mu\text{s}$ pro Skalenteil. Ein ideales Ergebnis wäre eine flache Linie in der Mitte. Die gelbe Linie zeigt die Signalstärke. Man beachte den Unterschied zwischen Tag (10 Uhr bis 14 Uhr) und Nacht.