

60. UKW-Tagung 2015

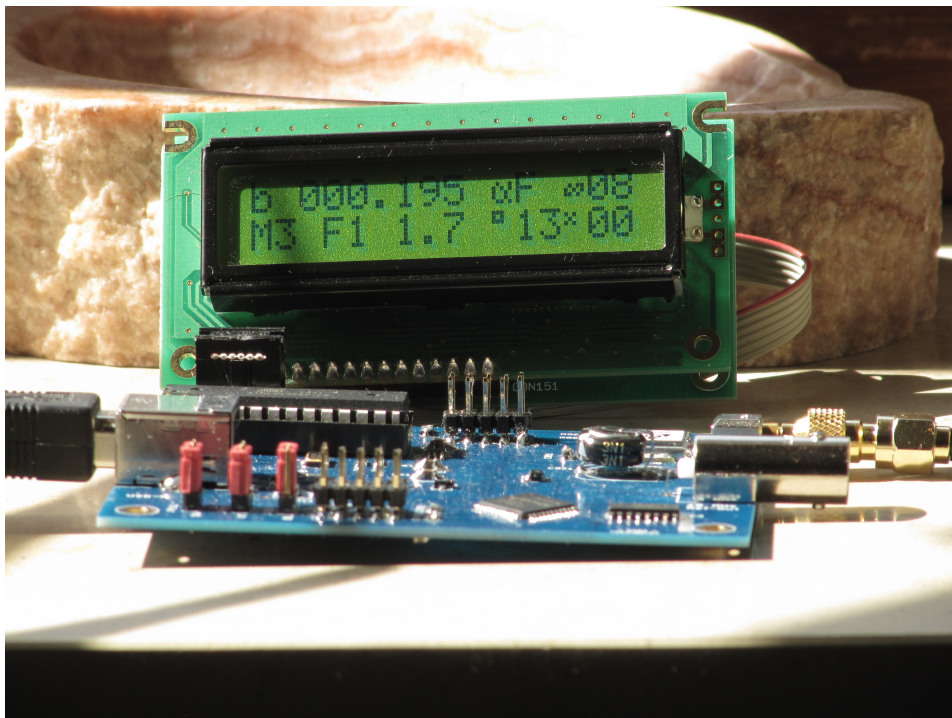
Ein GPS nachgeführtes Frequenznormal für 10 MHz auf der Basis eines TCXOs

DF4IAH, Ulrich Habel

Ladenburg

DF4IAH - 10 MHz-Ref.-Osc. ein GPS nachgeführtes Frequenznormal für 10 MHz auf Basis eines VCTCXOs und eines ATmega 328P Prozessors

Mehr und mehr gehen die Hersteller von Halbleiterbauteilen dazu über, ihre integrierte Schaltkreise als SMD-Bauteile (Surface Mounted Device = Oberflächenmontage) vorzusehen. Als Amateur ist es sehr wohl möglich, sich an die geänderten Anforderungen der Leiterplattenherstellung anzupassen. Diese Schaltung vereint die Möglichkeit des Nachbaus mit der einer aktuellen Schaltung mit modernen Schaltkreisen. Mit Hilfe der GPS Positions- und Zeitbestimmung kann eine hohe zeitliche Auflösung von besser als ± 15 ns erreicht werden.



1 GPS nachgeführtes Frequenznormal für 10 MHz auf der Basis eines VCTCXOs und eines ATmega 328P Prozessors

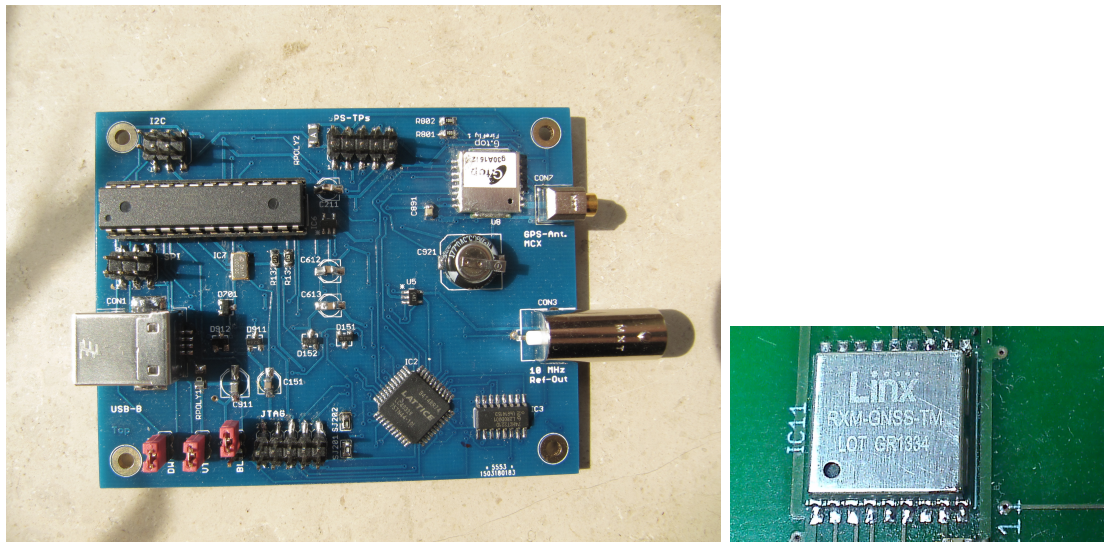


Abbildung 1: Draufsicht auf das Frequenznormal - GPS-Receiver-Chip basierend auf MT3333

Bei der vorliegenden Schaltung handelt es sich um einen prozessorgesteuerten 10 MHz Taktgenerator mit einer zeitsymmetrischen Rechteckspannung (50% / 50%) von $3,3 V_{SS}$. Auf dem Board befindet sich neben einem Atmel ATmega 328P Prozessor auch ein GPS-Receiver Chip. Dieser erhält seine GPS-relevante Information über den GPS-Hochfrequenz-Eingang, der mit einer Gleichspannung beaufschlagt ist, um damit eine aktive GPS-Antenne zu versorgen. Das GPS-Modul bestimmt autark seine Zeit- und Positionsinformationen und kommuniziert mit Hilfe von NMEA 0183 mit dem Atmel-Prozessor. Zusätzlich verfügt dieser Chip, welcher auf dem MediaTek MT3333-Chipsatz basiert, einen 1 PPS-Ausgang. Das ist ein Sprungsignal von low nach high, welches den Beginn einer jeden neuen Sekunde im GPS-System anzeigt. Dieses Signal erfolgt weltweit zum selben Zeitpunkt und ist synchron mit den Atomuhren auf dieser Erde und auf den Satellitenbahnen um uns herum. Es wurde bereits viel über GPS als Zeitbasis geschrieben, daher möchte ich nicht näher auf dieses Thema eingehen.

Die vorliegende Schaltung hat einen VCTCXO (spannungsgesteuerter und temperaturkompensierter Oszillator) mit einer Frequenz von 20 MHz. Der Prozessor, ein Atmel ATmega 328P, stellt mittels Taktzähler-Zeitstempel den Low-High-Übergang des 1 PPS-Signals vom GPS-Modul fest und zählt nun seine eigene 20 Millionen Taktimpulse zwischen zwei steigenden Flanken des 1 PPS-Signals. Daraus lässt sich eine Frequenzabweichung berechnen.

1.1 AFC - Automatic Frequency Correction

Die Frequenzabweichung wird nun durch ein Optimierungsverfahrens verringert, indem der VCTCXO mit einer angepassten Ziehspannung an die 20,000 MHz Vorgabe herangezogen wird. Mit Hilfe eines PWM-Ausgangs des Atmel-Prozessors wird diese Ziehspannung für den VCTCXO zur Verfügung gestellt, indem das Taktverhältnis entsprechend eingestellt wird.

1.1.1 PWM - Ziehspannung des VCTCXO

Der ATmega 328P besitzt drei Zeit- und Zählregister, davon eines mit 16 Bit Breite, die anderen zwei haben 8 Bit Breite. Der 16 Bit breite Zähler wird bereits für die Zeiterfassung benötigt, so

kann für die PWM nur noch eine 8 Bit Auflösung genutzt werden. Das hat zwar den Nachteil, dass die Auflösung der Schrittweite darunter leidet; aber im Gegenzug ist der PWM-Takt recht hochfrequent:

$$f_{PWM} = \frac{20 MHz}{2^8} = 78,125 kHz$$

Das führte dann zu der Idee, den PWM-Zähler zusätzlich mit einer 8 Bit breiten Nachkomma-Einheit auszustatten. Dabei läuft ein Sub-PWM Zähler mit und der Prozessor entscheidet, ob der Integeranteil des PWM Werts inkrementiert wird, so lange die Nachkomma-Einheit noch unvollständig ist. Das hat zur Folge, dass der ausgegebene Integer-PWM-Wert im Takt von $f_{PWM_{Sub}} = \frac{78125 Hz}{2^8} = 305,176 Hz$ inkrementell springt. Das nachgeschaltete Tiefpassfilter muss entsprechend eine niedrige Grenzfrequenz aufweisen, diesen einzelnen Schritt auszumitteln. Diese daraus resultierende Ziehspannung wird dann dem VCTCXO zugeführt. Ein Spannungshub von 1 V bewirkt eine Frequenzdrift von rund 2,5 ppm, oder hier als Beispiel berechnet: $2,5 \cdot 10^{-6} \cdot 20 MHz = 50 Hz$. Der Ausgangstakt wird durch eine 2:1 Teilung des 20 MHz Muttertaktes gewonnen, weshalb am 10 MHz-Taktausgang bei einem Volt Spannungshub nur noch eine Frequenzabweichung von 25 Hz wäre.

1.2 APC - Automatic Phase Correction

Viele Bausätze für ein 10 MHz-Frequenznormal begnügen sich mit einer Frequenzzählung und -korrektur. Generell lässt sich zwar die Frequenz recht genau einregeln - hier beim vorliegenden Fall wären das ≈ 1 Zählertakte auf 20 Millionen Takte, entsprechend $5 \cdot 10^{-8}$ oder 50 pbb (parts per billion) - und mit Sicherheit sind über eine Langzeitmittelung auch noch bessere Werte erzielbar. **Um jedoch an der Langzeitgenauigkeit eines Atomuhren-Systems teilzunehmen ist es nötig, die Phase auszuregeln und dafür sicherzustellen, dass diese nicht übersprungen wird.** Somit würde zwar die kurzfristige zeitliche Schwankung der Phasenausregelung typisch für eine Betriebssoftware sein, die langfristige Genauigkeit wäre davon jedoch nicht berührt und einzig der GPS-Zeitbasis und deren Genauigkeit unterworfen.

Im vorliegenden Fall wird die Phasenauswertung freigegeben, sobald die AFC eine Genauigkeit von 100 pbb erreicht und die Qualifizierungszeit überstanden hat. Ab diesem Zeitpunkt prüft die Phasenregelung nach einem passenden Zeitfenster und übernimmt dann letztendlich die Ausregelung. Die Vorgabefrequenz für die AFC wird absichtlich um $\frac{1}{20} Hz$ verstimmt (also ein Takt mehr als die 20 Mio. Takte pro Sekunde), damit das Phasen-Fenster nach maximal 20 Sekunden gefunden wird.

Mit jeder Sekunde wird dann der Phasenfehler gegenüber der 1 PPS-Flanke bestimmt und die PWM-Einheit mit dem Offset-Wert einmal pro Sekunde angepasst. Ein Summationstopf dieser Phasenfehler bestimmt die detektierte Frequenzdrift, die dann vom Prozessor ausgewertet wird, um den PWM-Ausgangswert langsam nachzuziehen. Dieser Algorithmus ist es, der darüber entscheidet, ob das Frequenznormal ab und zu ausrastet, oder der Impulsvorgabe nicht ausreichend folgen kann, oder aber ein brauchbarer Kompromiss gefunden ist. Nach einer Einlaufzeit von rund 1 Stunde sind GPS-Chip und VCTCXO so frequenzstabil, dass der Phasenfehler keine Drift-Tendenz mehr aufweist. Vielmehr sind dann nur noch Phasensprünge vorhanden, die von der Empfangssituation des GPS-Receiver herrührt. Auf Grund der sich im Orbit befindenden Satelliten, verändern sich die Laufzeiten zu diesen. Zusätzlich wählt der Receiver aus diversen Satelliten aus, um daraus einen Positions- und Zeit-Fix zu bestimmen. Somit ist die Zeitbasis um einige zehn Nanosekunden variant, was leicht mit mehreren GPS-Systemen am Oszilloskop beobachtet werden kann. Mit Fingerspitzengefühl muss hier ein Optimum gefunden werden, den Regel-Algorithmus entsprechend auszugestalten und zu parametrisieren.

Für die Schaltung wird ein CPLD von Lattice genutzt, der alle Aufgaben der TTL-Logik und der verstreuten „Glue-Logik“ in sich vereint. Der Monoflop ist jedoch nicht abbildbar und muss explizit in der Schaltung integriert werden.

1.3 CPLD als Logik-Sammelstelle

Ein CPLD reduziert nicht nur die benötigten TTL-Gatter, Schieberegister und -Zähler um sich herum, in dem es diese in sich vereint, sondern als angenehmer Nebeneffekt ist auch der Energiebedarf geringer, die Durchlaufzeiten der Gatter z.T. erheblich kürzer (höhere Grenzfrequenz) und in einem gewissen Rahmen kann die abgebildete „Schaltung“ auch noch nachträglich mit einer veränderten Konfiguration angepasst oder optimiert werden. Bei der Erstellung dieser CPLD-Konfiguration kommt VHDL als Hochsprache zum Einsatz. Diese ist normiert und ermöglicht somit, dass diese Schaltungs- und / oder die Verhaltensvorschrift auf eine andere Architektur eines anderen Herstellers übernommen werden kann. Die vorliegende Schaltung hat daher zu dem CPLD Chip auch noch einen Programmier-Stecker, damit die CPLD Konfiguration nachträglich geändert werden kann. Das CPLD wird mit 3,3 V betrieben, ist allerdings an seinen digitalen I/O-Eingängen 5 V resistent und somit auch als Pegelumsetzer von 5 V nach 3,3 V geeignet.

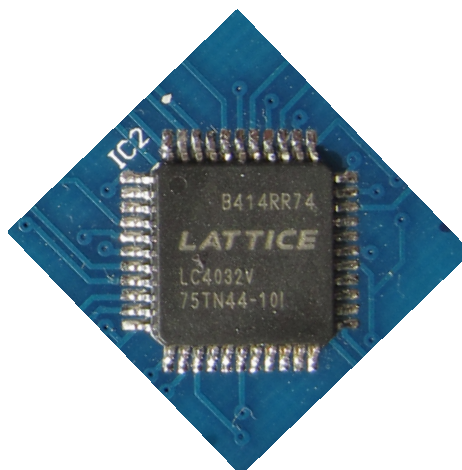


Abbildung 3: Lattice LC4032V mit 44 Pins

Nachfolgend die auf RTL abgebildete Schaltung, die mittels VHDL in dem CPLD abgebildet wurde:

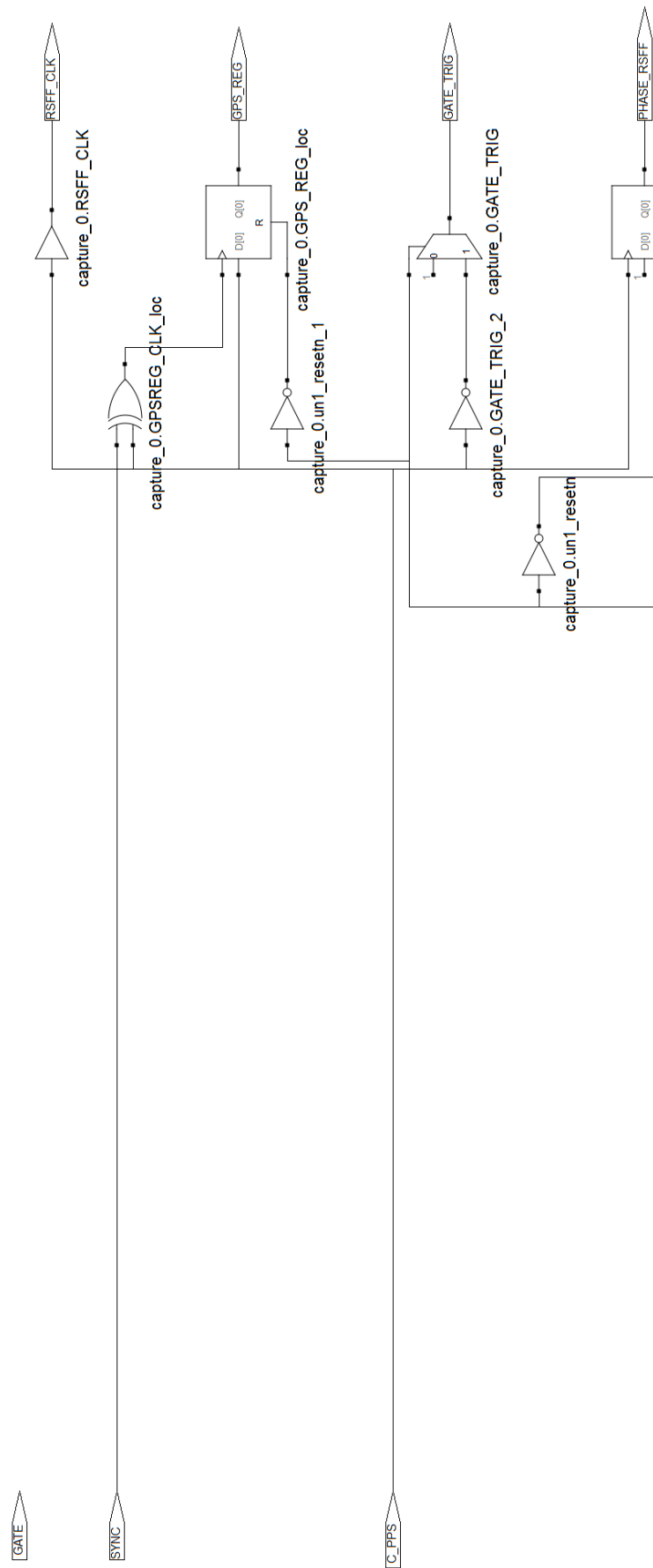


Abbildung 4: CPLD Kombinatorik - Teil 1 von 4

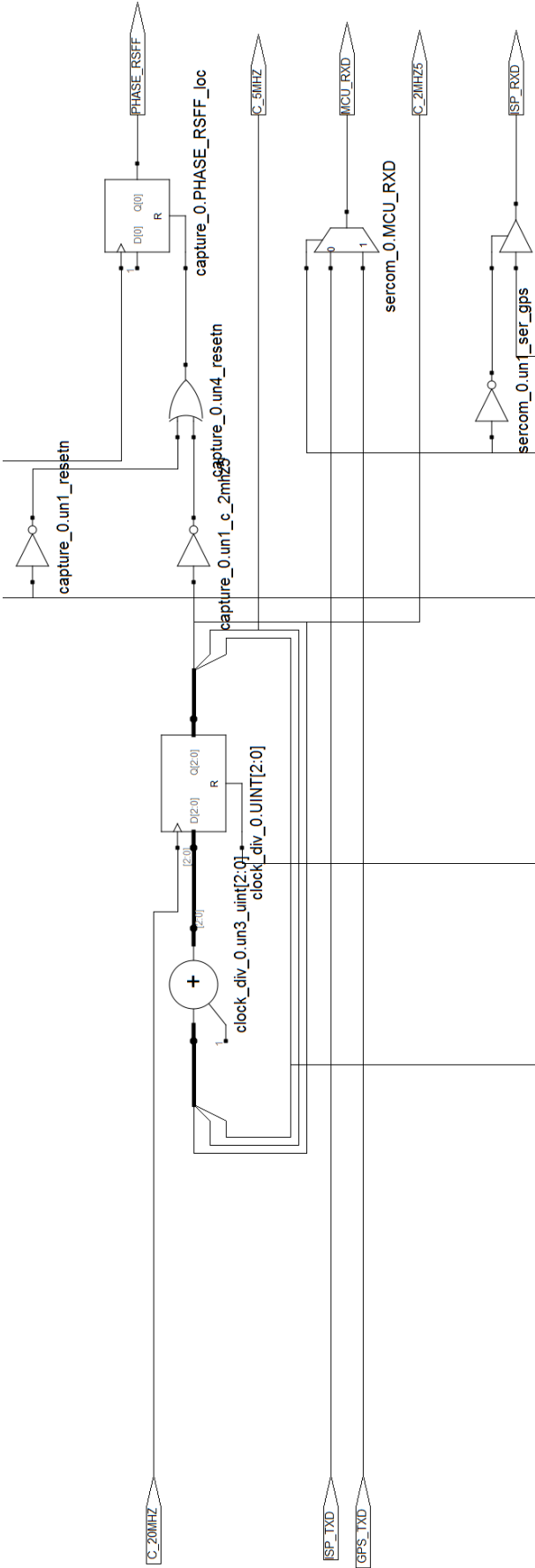


Abbildung 5: CPLD Kombinatorik - Teil 2 von 4

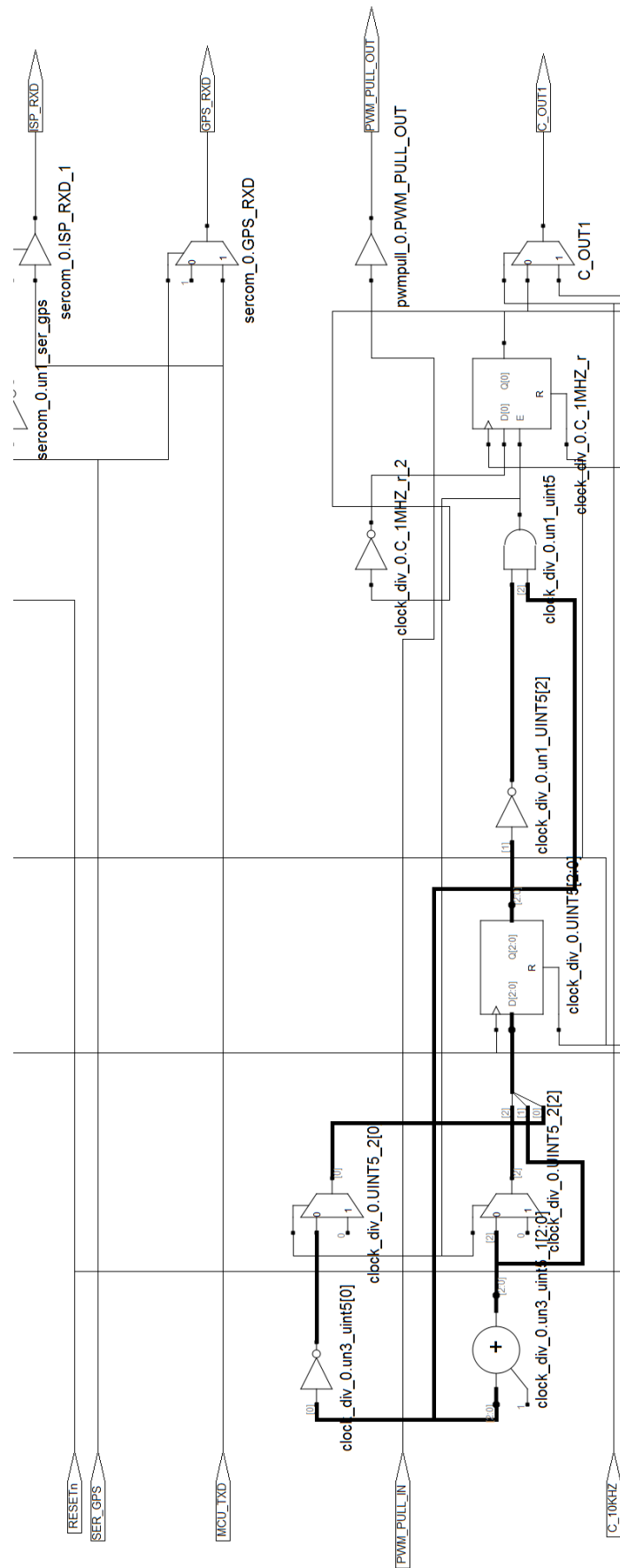


Abbildung 6: CPLD Kombinatorik - Teil 3 von 4

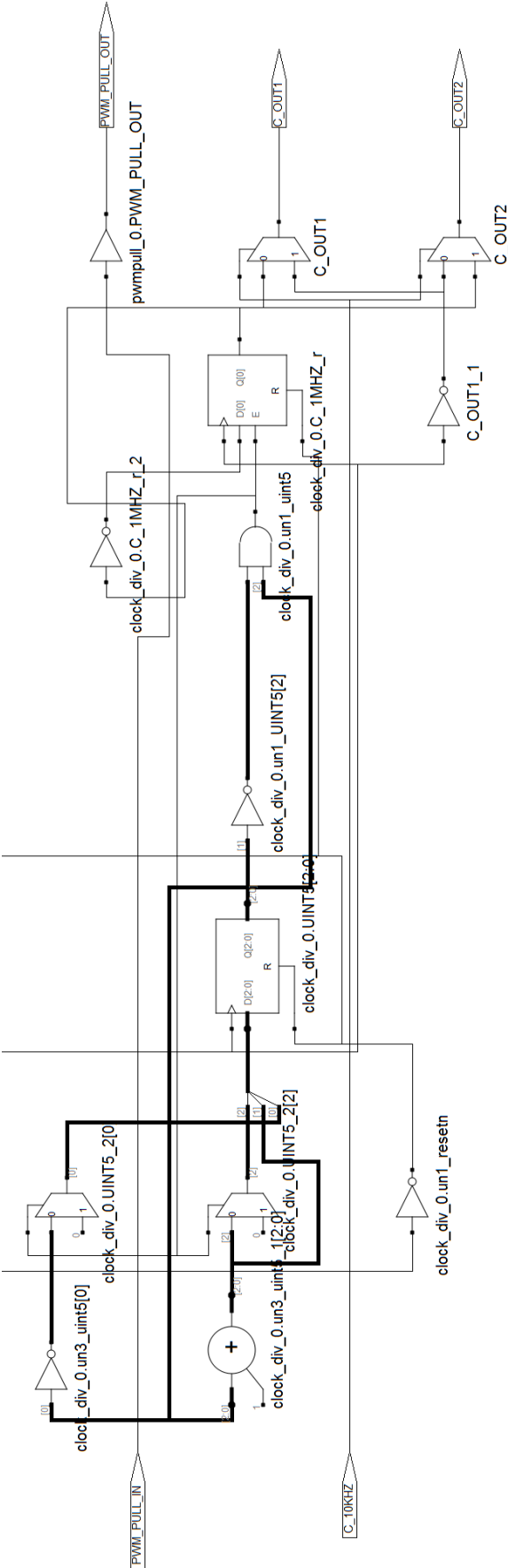


Abbildung 7: CPLD Kombinatorik - Teil 4 von 4

1.4 Atmel ATmega 328P

Der Prozessor ist eine Atmel MCU, und in der Tat wäre der Referenz-Generator noch sehr viel kleiner ausgefallen, wäre keine Phasenbeziehung berücksichtigt worden. Denn alle Aufgaben kann der ATmega bereits von Hause erledigen oder konnte durch eine Interrupt-Programmierung dazu erweitert werden:

- 20 MHz Takt erzeugen, am Clock-Eingang liegt das niedrig gepegelte VCTCXO-Signal an und wird am Pin 14 verstärkt ausgegeben
- mit Hilfe eines PWM-Ausgangs lässt sich eine sehr genau Breite des Signals erzeugen (8 Bit Integer / 8 Bit Nachkomma), in dem nachfolgenden Tiefpass wird daraus die Ziehspannung des VCTCXOs generiert
- die USB-Kommunikation ist zwar von Hause aus nicht für diesen Chip vorgesehen, dank der „VUSB“-Umsetzung in C und Assembler-Code ist dies in einem begrenzten Umfang jedoch möglich. Das daraus resultierende Prozessor-System wird durch die Interrupts stark beansprucht. Es ist Vieles zu berücksichtigen um alle anderen Programmteile daran anzupassen, dass neuer Programm-Code die bereits etablierte USB-Verbindung nicht stört
- zwei Signale wurden zuerst reserviert und dann doch noch aktiviert: der I²C-Bus. Dieser ermöglicht nun ein angeschlossenes LCD-Modul, den aktuellen Zustand des Referenz-Oszillators darzustellen
- über die A/D-Wandler-Kanäle kommen die Ziehspannung und die Phasen-Diskriminator Spannung an. Die Chip-Temperatur wird ebenfalls per internen A/D-Kanal ermittelt
- eine serielle Schnittstelle zum GPS-Modul ermöglicht die Auswertung der aktuellen Empfänger-Daten, ebenso aber auch die Steuerung des GPS-Empfängers
- natürlich dürfen die Zeit-Impuls-Informationen nicht fehlen, diese gelangen ebenfalls über Timer-Kanäle zur CPU
- letztendlich steht dann noch eine ISP-Programmier-Buchse zur Verfügung
- durch die Bootloader-Firmware ist ein Programmieren der Firmware direkt am USB-Bus möglich. Das Programmieren ist allerdings nur möglich, wenn der Jumper „BL“ gesetzt ist und somit der Bootloader anläuft

Weiterhin ist der Prozessor mit einer Taktrate von 20 MHz recht flott und der Befehlssatz ist trotz der RISC-Architektur nicht unnötig verkrüppelt, der viele zusätzliche Programmschritte zur Folge hätte, um das gleiche Ziel eines CISC-Prozessor zu erreichen. Die RISC-Befehle werden mit wenigen Arbeitstakten abgearbeitet, so entsteht ein sehr performantes System bei dennoch sehr geringer Leistungsaufnahme.

Der gemessene Strom bezüglich 5 V, wie sie von einem USB-Port entnommen wird:

	Stromaufnahme bei 5 V
ohne LCD-Modul	65 mA
mit LCD-Modul, Backlight aus	70 mA
mit LCD-Modul, Backlight ein	180 mA

Tabelle 1: Stromaufnahme mit und ohne Display

Mit einer USB-Powerbank lässt sich dieses Frequenznormal auch über Stunden mit einer Akku-Ladung betreiben.

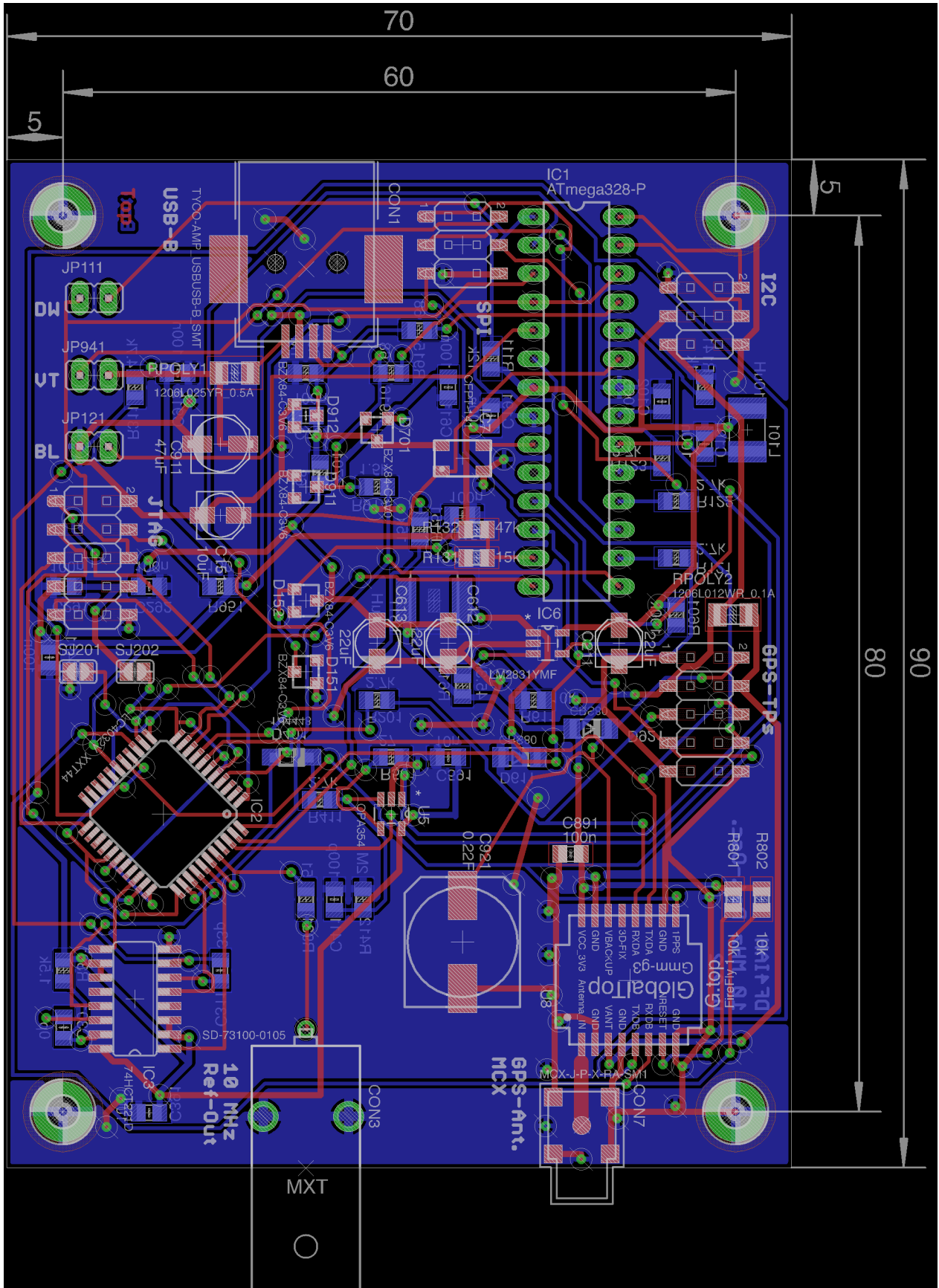


Abbildung 8: Schaltungslayout der V2.2

2 LCD-Modul Statusinformationen

An den I²C-Bus lässt sich ein LCD-Modul anschließen, welches aktuelle Status-Informationen darstellt. Nachfolgend sehen wir die drei Bildschirmseiten, die automatisch im drei Sekundentakt wechseln:



Abbildung 9: LCD-Module - Datums-Information

Zuerst wird das Datum und die UTC-Zeit dargestellt. Die Zeile darüber wechselt nicht und zeigt die derzeitige Oszillator-Genauigkeit in 10^{-9} (parts per billion) an. Weiterhin die Genauigkeitsklasse („0xF“) und die zur Positions- und Zeitbestimmung genutzten Satelliten, hier sieben.

Der zweite Bildschirm zeigt im unteren Feld den derzeitigen Satelliten-Status an. Derzeit „navigiere“ der GPS-Chip 3D und nutze Differential-GPS zur Genauigkeitserhöhung. Weiterhin seien zwölf westliche Satelliten derzeit über dem Horizont, die östlichen Satelliten waren zu dieser Zeit noch nicht vom System erfasst.

Der letzte Informationsschirm zeigt die aktuelle PWM-Einstellung an: 106,773 von 256 Zeitan teilen wird die Spannung auf „high“ gehalten. Das führt dann zu der nötigen Ziehspannung des VCTCXOs, der somit dem 1 PPS-Zeitimpuls synchron folgt. Die am Tiefpass liegende Ziehspannung ergibt sich dann zu 1.394 Volt.

Das LCD-Modul ist ebenfalls bei DF4IAH erhältlich, der Aufbau flugs vollbracht. Dieses Modul basiert auf dem I²C-Expander-Baustein MCP23017 und stellt 16 I/O-Kanäle für das leicht erhältliche LC-Display zur Verfügung. Zusätzlich kann die Hintergrundbeleuchtung zugeschaltet werden.

Weiterhin empfehle ich die Dokumentation „Von der Idee zur Platine“ [?], die den Werdegang dieser Schaltung beschreibt. Dieser ursprünglich vorgesehene Text für den Tagungsband war jedoch zu umfangreich, um abgedruckt zu werden.

Vy 73

Uli, DF4IAH

Weiterführende Literatur

Literatur

- [1] Ulrich Habel, DF4IAH. *Von der Idee zur fertigen Platine - ein Werdegang des DF4IAH 10 MHz Ref-Osc. V2 Normfrequenz-Oszillators.* HTTP-Server: http://dl0wh.de/wp-content/uploads/2015/02/DF4IAH-10MHzRefOsc_V2_Beamer_20150910.pdf
- [2] Ulrich Habel, DF4IAH. *Technischer Aufbau des DF4IAH 10 MHz Ref-Osc. V2 Normfrequenz-Oszillators.* HTTP-Server: [.../2015/02/DF4IAH-10MHzRefOsc_V2_UKW60-Skriptum_20150901.pdf](http://dl0wh.de/wp-content/uploads/2015/02/DF4IAH-10MHzRefOsc_V2_UKW60-Skriptum_20150901.pdf)
- [3] Ulrich Habel, DF4IAH. *Von der Idee zur fertigen Platine - ein Werdegang des DF4IAH 10 MHz Ref-Osc. V2 Normfrequenz-Oszillators.* HTTP-Server: http://dl0wh.de/wp-content/uploads/2015/05/DF4IAH-10MHzRefOsc_V2.pdf
- [4] Ulrich Habel, DF4IAH. *Aufbauanleitung für den DF4IAH 10 MHz Ref-Osc. V2 Normfrequenz-Oszillator.* HTTP-Server: http://dl0wh.de/wp-content/uploads/2015/05/DF4IAH_10MHzRefOsc_V2_Aufbauanleitung.pdf
- [5] Ulrich Habel, DF4IAH. *Aufbauanleitung für die DF4IAH LCD-Module. V1 Anzeige-Einheit.* HTTP-Server: http://dl0wh.de/wp-content/uploads/2015/05/DF4IAH_LCD-Module-1V0_Aufbauanleitung.pdf
- [6] Ulrich Habel, DF4IAH. *Eclipse-AVR Projekt: DF4IAH 10 MHz Ref-Osc. V2 Sourcen für eine Atmel ATmega 328P MCU.* SVN-Server Leserechte: auf Anfrage.
- [7] Ulrich Habel, DF4IAH. *VHDL-Sourcen für das CPLD Lattice LC4032V - DF4IAH 10 MHz Ref-Osc. V2.* SVN-Server Leserechte: auf Anfrage.
- [8] Ulrich Habel, DF4IAH. *Eagle-Layout-Dateien für eine doppelseitige Leiterplatte - DF4IAH 10 MHz Ref-Osc. V2.* SVN-Server Leserechte: auf Anfrage.