R2T2
Anleitung
und
Schnittstellen

Version 0.2
24.06.16

Allgemeines

Der R2T2 ist ein DDC/DUC-SDR-Transceiver. Die aktuelle Konfiguration stellt 8 unabhängige Empfänger bereit. Auf dem R2T2 wird zur Zeit eine modifizierte QtRadio-Version (http://napan.ca/ghpsdr3/index.php/Main_Page) als SDR-Software verwendet.

Grundüberlegungen zum Schaltungsdesign des R2T2

Der R2T2 ist ein kompletter SDR Transceiver, der durch seinen Aufbau den aktuellen Stand der Technik widerspiegelt und durch seine umfangreiche Peripherie und modularen Aufbau auch zukünftig keine Wünsche offen lassen dürfte. Zum Betrieb ist keinerlei zusätzlicher PC notwendig, an den R2T2 können optional auch direkt ein Monitor (HDMI) und eine USB-Tastatur bzw. eine USB-Maus (noch besser: Wireless USB Desktop)angeschlossen werden.

Das Konzept:

Da sich das Direktwandlerprinzip („Antenna-to-Bits") in vielen hundert HiQSDR-Modulen weltweit bewährt hat, legten wir dies auch als Basis für einen Muli-User WEDSDR für den DARC fest.

Dabei stellte sich sehr schnell heraus, dass für ein zukunftsfähiges Konzept mehr als ein „aufgebohrter" DDC/DUC - SDR notwendig ist. Ein SDR-Multiuser-System, bei dem alle Anwender gleichberechtigt einen „eigenen" kompletten SDR in voller Auflösung und mit allen Optionen über das Internet zur Verfügung haben erfordert nicht nur Softwaremäßig entsprechend umfangreiche Vorkehrungen sondern auch die Hardware muss entsprechend leistungsfähig sein.

Die Software eines modernen SDR-Konzeptes lässt sich nachträglich erweitern und ändern, bei der
Hardware ist dies nicht so einfach möglich. Aus diesem Grund haben wir sehr viel Augenmerk auf modernste Bauelemente und Erweiterbarkeit (und Verfügbarkeit!) der Hardware gelegt.
Das aus diesen Anforderungen entstandene Konzept des R2T2 lässt viel Luft für die Zukunft und ermöglichtes es, die Baugruppe wahlweise als Multiuser-WEBSDR oder als kompletten, erweiterbaren voll digitalen standalone Multiband-Transceiver zu verwenden.

Die ADC/DAC Wandler:

Bei der Wahl der ADC/DAC wurden hochwertige, schnelle dual-14-Bit-Wandler von Analog Devices vorgesehen.

16 Bit Wandler sind hier unseren Erachtens nur ein Werbeargument und technisch sinnlos, da die letzten beiden Bits in der Praxis weder vernünftig auflösbar noch nutzbar sind. Bereits echte 14 Bit stellen extrem hohe Anforderungen, u.a. an den Hauptoszillator! (der Jitter muss kleiner als 115 Femto-Sekunden(!) sein) von restlichen Systemanforderungen (Stromversorgung, Einstrahlung) sowie externen Einflüssen wie Antennenrauschen, QRN usw. ganz zu schweigen.

Wichtiger als marktschreierische ADC-Auflösungen erschienen uns praxistaugliche, hoch aussteuerbare Eingangsverstärker (PGA) und exakt Software-steuerbare Abschwächer.
Wir verwenden für den Hauptoszillator eine spezielle Multi-Chain-PLL mit integrierten programmierbaren Teilern (SI5344) der es zusätzlich gestattet, das gesamte System mit einer extern zugeführten Frequenznormal (z.B.10 MHz)zu synchronisieren.
So haben wir auch der Anforderungen von Betriebsarten mit extremer Frequenzgenauigkeit und -Stabilität Rechnung getragen.

Ein stabiler TXCO erlaubt aber auch ohne externe Synchronisierung Betriebsarten wie WSJT oder EME obwohl auch diese Betriebsarten besondere Anforderungen an die Frequenzgenauigkeit haben.

Das Controllerboard: ein kompletter PC auf einem Chip:

Schnell stand fest, dass neben modernen doppelten ADC/DAC auch das Prozessorsystem auf dem Board sein sollte, um die entstehenden I/Q-Daten direkt verarbeiten zu können und nicht erst über ein (langsames) Interface zu einem andern PC leiten zu müssen.
Hier fiel unsere Wahl auf das modernste was momentan verfügbar ist, ein komplettes System-on-Chip ZC7020 (ZYNQ) der neben einem großen FPGA auch zwei ARM9-Cores inklusiver kompletter Peripherie direkt auf einem Chip enthält.

Peripherie & Schnittstellen

Ebenfalls viel Wert haben wir auf die Schnittstellen zum Anwender gelegt.
Hierfür sind neben einem Gigabit-Netzwerkanschluss zwei USB2 (OGT und 2x Seriell-Emulation), SD-Karteninterface usw. vorhanden. Die Audioaufbereitung übernimmt der programmierbare Audio-DSP ADAU-1791, für die Audioausgabe (Monitoring) ist ein 2W Verstärker mit Kopfhörer- und Lautsprecheranschluss vorhanden.

Um Lieferschwierigkeiten einiger besonderer Peripherie für den ZYNQ zu umgehen (1.8V-Gigabit-PHY, DDR3L-Chips) haben wir uns entschlossen, den FPGA/Prozessorteil mit Speicher und Interfaces auf ein separates Modul auszulagern. Dies ermöglicht neben späterem „upgrading" auf noch schnellere und größere ZYNQ-SoCs den R2T2 komplett auf einer Leiterplatte mit „nur" 6 Lagen unterzubringen.

Besonderes Problem war dabei, die durch die vielen Optionen notwendigen Steckverbinder an sinnvollen Stellen an der Platine unterzubringen.
Die Leiterplatte des R2T2 ist 200x160mm groß und kann direkt in ein BOPLA-Alu-Stranggußgehäuse
ABP 2070-0200 eingeschoben werden, die Abwärme wird durch unter der Leiterplatte montierten Wärmeverteilerplatten direkt an das Gehäuse abgegeben.
Für den ZYNQ-SoC ist optional ein leiser, temperaturgesteuerter Lüfter vorhanden.
An der Rückseite sind alle Anschlüsse vorhanden, die für den Betrieb notwendige sind.

An der Frontplatte kann optional ein LCD und div. Encoder (für den Transceiverbetrieb) angeschlossen werden. Da der R2T2 für den standalone und für Remotebetrieb konzipiert ist, kann die Stromversorgung sowie PTT und KEY über galvanisch getrennte Leitung angesprochen werden wobei alle Spannungen, Ströme und Betriebszustände über das Webinterface kontrolliert werden können.
Auf dem R2T2-Board sind Steckverbinder für Zusatzmodule wie VHF/UHF/SHF-Erweiterung sowie einem neuartigen Preselektor vorhanden. Dieses neuartige Konzept eines MultiUser-Preselektor erlaubt es mehreren Anwendern oder mehreren Software definierten Empfängern gleichzeitig, verschiedene Bänder zu nutzen.

Einsatz des R2T2 als DARC Web-Gerät

In der ersten Ausbaustufe des DARC Webservers ist auf logistischen, organisatorischen und rechtlichen Gründen nur ein Betrieb als Empfänger vorgesehen. Nach Klärung aller mit dem Sendebetrieb stehenden Vorgaben kann dieser freigeschaltet werden. Momentan ist aus Sicherheitsgründen auf allen ausgelieferten R2T2 DARC-Webservergeräten der Sendebetrieb Hardwaremäßig gesperrt.

Bitte beachten Sie auch dass möglichst alle Nutzer alle Möglichkeiten des Gerätes nutzen wollen. Bitte teilen Sie Ihren Mitgliedern mit, dass sie kooperativ nach Beendigung er eigenen Nutzung QTRadio wieder schließen und freigeben.

Bei der Nutzung und Antennenplanung sollten Sie daran denken, möglichst Breitbandantennen einzusetzen. Nur so kann auf vielen Bändern gleichzeitig gearbeitet werden. Bitte achten Sie dabei aber auch auf ausreichenden Blitzschutz / Überspannungsschutz der Antennen! Defekte durch Überspannung / Blitzschutz sind naturgemäß von keiner Garantie gedeckt!

Inbetriebnahme

Zum Betrieb des R2T2 ist die beiliegende SD-Karte einzusetzen, die Spannungsversorgung ist mit einem 12V-Netzteil (mindestens 2A) zu verbinden und das Netzwerkkabel sollte mit dem PC oder dem Switch verbunden werden.

Die SD-Karte enthält zwei Partitionen. Die erste Partition ist enthält ein Windows-Dateisystem mit dem Bootloader, der FPGA Software und dem Linux-Kernel. Auf der zweiten Partition liegt das Linux-Dateisystem mit den R2T2-Daten.

Nach dem Einschalten bootet der R2T2 das auf der SD-Karte liegende Linux. Der Bootvorgang kann bis zu einer Minute dauern. Anschließend ist das Gerät über die IP-Adresse 192.168.1.99 über ssh zu erreichen. Der Login-Name und das Passwort ist 'r2t2'.

Konfiguration

Der R2T2 benutzt ein Arch-Linux-System (https://www.archlinux.de). Alle Fragen, zur allgemeinen Konfiguration werden ausführlich auf der gut gepflegten Wiki-Seite ('https://wiki.archlinux.de') beantwortet.

Zur Konfiguration das R2T2 sind einige Dateien zu bearbeiten. Installierte Editoren sind 'nano' oder für fortgeschrittene Nutzer auch der 'vi'.

Zunächst sollte die gewünschte Netzwerkadresse festgelegt werden, dazu ist eth0 in /etc/netctl zu editieren:

sudo nano /etc/netctl/eth0

Hier sollten die Einträge Addresse, Gateway und DNA an das eigene Netzwerk angepasst werden. Anschließend kann der R2T2 neu gestartet werden, damit die neuen Einstellungen wirksam werden:

reboot

Nach ca. einer Minute sollte man sich über die neue IP-Adresse neu einloggen können.

Hinweis: Falls man sich ausgesperrt hat, kann man die SD-Karte auch direkt an einem Linux-Rechner editieren. Dazu die SD-Karte aus dem R2T2 entnehmen, in den Kartenleser des PCs

einstecken.

sudo mount /dev/mmcblk0p2 /mnt

cd /mnt

sudo nano /etc/netctl/eth0

umount /mnt

Anschießend ist noch die dspserver-Konfiguration zu berarbeiten:

sudo nano /root/dspserver.conf

Hier ist mindestens das Rufzeichen (call) einzutragen, der Ort (location), die Antenne (ant) sind optional zu setzen. Wenn der R2T2 über das Internet bereitgestellt werden soll, ist 'share' auf 'yes' zu setzen.

Die DSP-Server sind nun mit

sudo killall dspserver

zu beenden. Sie werden nach einigen Sekunden automatisch neu gestartet.

Der R2T2 ist nun einsatzbereit.

Damit die R2T2-Empfänger aus dem Internet erreicht werden können, müssen noch die TCP-Ports 8000 bis 8007 auf dem Router freigegeben werden.

Software

FSBL-Bootloader

Nach dem Reset des Prozessors wird der initiale Bootloader (First Stage BootLoader) (FSBL) vom Zynq Z7020-Soc auf verschiedenen Bootmedien gesucht. Der FSBL ist ein Teil der boot.img-Datei auf der ersten Partition der SD-Karte.

Hinweis: Auf der SD-Karte liegen zwei boot.img-Dateien, *boot.img.hdmi* und *boot.img.nohdmi.* Die Datei *boot.img.nohdmi* enthält einen FSBL, der das HDMI-Core des FPGAs nicht initialisiert. Damit hat man keine Bildschirmausgabe über den HDMI-Anschluss. In dieser Konfiguration steht die volle DMA-Bandbreite zwischen FPGA und CPU für den SDR-Transceiver zur Verfügung. Bei aktivierter HDMI-Ausgabe wird die DMA-Bandbreite zwischen SDR und HDMI aufgeteilt und es kann zu Störungen im SDR-Empfang kommen.

Um die HDMI-Ausgabe zu aktivieren, ist das boot.img durch boot.img.hdmi zu ersetzen.

U-Boot

Der FSBL startet den eigentlichen Bootloader (Second Stage Bootloader). Der R2T2 verwendet den U-Boot. Dieser ist Teil der boot.img-Datei auf der ersten Partition der SD-Karte.

FPGA-Image

Wesentliche DSP-Funktionalität der SDR-Empfänger und des Senders werden im FPGA des Zync-Soc realisiert. Das FPGA-Image wird vom FSBL in das FPGA geladen. Auch das FPGA-Image ist Teil der boot.img.

Linux-Kernel

Der U-Booter lädt den Linux-Kernel in den Speicher und startet diesen. Es wird derzeit ein modifizierte Kernel der Version 4.0.0 verwendet.

Linux

Das Arch-Linx-System auf der zweiten SD-Partition stellt für den SDR-Transeiver die grundlegenden Funktionen, wie Netzwerk, TCP/IP, diverse Systemserver, Webinterface etc. bereit.

R2T2

Das R2T2-Programm dient zu Test der FPGA-Schnittstellen und ist zum normalen Betrieb nicht notwendig.

usage: r2t2

-c rx	*select rx [0..8]*
-r rxfreq	*set rx freq*
-a rx rate	*set rx rate*
-t rxfreq	*set tx freq*
-b tx rate	*set rx rate*
-s samplerate	*use sample rate*
-u	*init clock chip*
-p	*init adc and dac*
-g gain (-9 ..32)	*rx 1 gain*
-k gain (-9 ..32)	*rx 2 gain*
-i att (0..31)	*rx 1 att*
-j att (0..31)	*rx 2 att*
-e	*read rx data to stdout*
-f	*write 2 tone signal to tx data*
-d reg	*select reg*
-l cnt	*read from selected reg to reg+cnt*
-w val	*write val to selected reg*
-h	*this usage*

Alle Zahlen können Dezimal, als Float oder Hex eingegeben werden.

Beispiel:

Frequenz von Receiver 0 auf 7.1 MHz einstellen:

r2t2 -c 0 -r 7.1e6

Sollten die DSP-Server stören (z.B. bei der -e oder -f Option), können sie gestoppt werden. Für die Deaktivierung der DSP-Server ist zunächst die Überwachung der DSP-Server abzuschalten, anschießend sind die Server zu beenden:

systemctrl stop qtradiowatch.service
systemctrl stop qtradio.service

Die Server können später wieder neu gestartet werden, indem die Überwachung der DSP-Server wieder akiviert wird:

systemctrl start qtradiowatch.service

QtRadio

QtRadio verwendet ein Client-Server-Konzept. Auf den R2T2-Linux laufen acht DSP-Server die auf einen gemeinsamen R2T2-Server zugreifen. Der R2T2-Server ist das Interface zur Hardware das R2T2 und stellt die Schnittstelle zum FPGA dar. Die DSP-Server verarbeiten die (T)RX-Signale vom R2T2-Server und stellen für die QtRadio-Software den demodulierten Audiostrom und die FFT-Wasserfall-Daten bereit. Die QtRadio-Software sendet Kommandos, wie das Einstellen der Empfangsfreqenz an den DSP-Server. Eine grundlegende Einführung findet man auf der Webseite des Projekts unter http://napan.ca/ghpsdr3/index.php/Main_Page .

Auf jeden der acht DSP-Server kann ein oder mehrere Clienten mit der QtRadio-Software vom entfernten PC zugreifen. Nur der erste Client kann den genutzten Empfänger bedienen, alle anderen können nur mithören.

Die aktuelle QtRadio-Software kann nur 4 unterschiedliche Empfänger auswählen, daher sollte die beiliegende angepasste Software verwendet werden. Unter Receiver → Configure → Receiver lassen sich dort die Empfänger von 0 bis 7 auswählen.

Wir d fortgesetzt, bitte beachten Sie Die Wiki—Seite des Projektes unter

www.r2t2.de für aktuelle informationen!