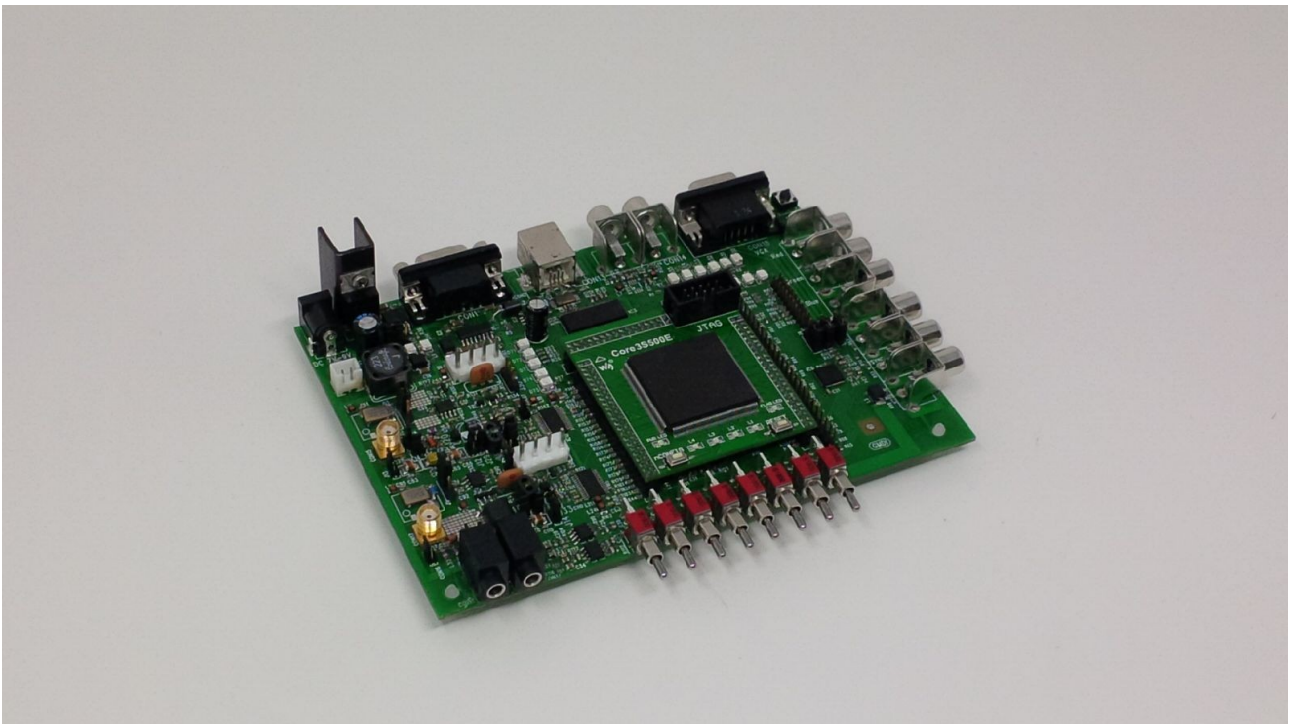


# User's Manual

SDR-500 Rev. 3  
Software Defined Radio Evaluation System



# Inhaltsverzeichnis

1	Einleitung.....	3
2	Erläuterung der Teilbaugruppen.....	4
2.1	FPGA.....	4
2.2	GPIO.....	5
2.3	Switches + LEDs.....	6
2.4	AUX I/O High-Speed.....	6
2.5	3-fach 8 Bit DAC (Video).....	6
2.6	USB-Controller.....	7
2.7	RS232 / UART.....	8
2.8	Audio.....	8
2.9	HF-Frontend.....	8
3	Inbetriebnahme.....	9
4	Beispiele.....	10
4.1	Serielle Schnittstelle.....	10
4.2	USB.....	11
4.2.1	Installation am PC.....	12
4.2.1.1	USB-Treiber.....	12
4.2.1.2	USB-Applikation und FPGA-Designsoftware.....	15
4.2.2	Programmierung der Firmware des USB-Controllers.....	16
4.2.3	FPGA-Konfiguration laden.....	19
4.2.4	Test.....	19
4.3	VGA.....	20
4.4	Audio.....	21
4.5	HF RX/TX.....	21
Anhang.....		24
Schaltplan.....		24
Bestückungsdruck.....		27
Schaltbild FPGA-Konfiguration.....		28
Abbildungsverzeichnis.....		29
Literaturverzeichnis.....		30

# 1 Einleitung

Das SDR-500 ist ein Software-Defined-Radio Evaluation-Board mit RF-Frontend.

Mögliche Einsatzgebiete sind:

- Software-Defined-Radio
- Spektralanalyse
- MPX (De-)Coder (Stereo, Video, ... )
- Video-Patterngenerator
- Arbitrary Waveform Generator

Folgende Teilbaugruppen sind im SDR-500 enthalten:

- FPGA Spartan 3-500 (optional Spartan-6)
- USB Controller
- Konfigurierbares HF RX/TX-Frontend mit Antialias-Filter
- 3-Kanal 8-Bit DAC (z.B. Video RGB)
- Audio I/O - Analog Stereo Codec
- GPIO
- RS-232, USB 2.0 Highspeed
- Switchpad, LEDs, AUX Output
- RS-232

Abbildung 1 zeigt den Aufbau mit den Teilkomponenten.

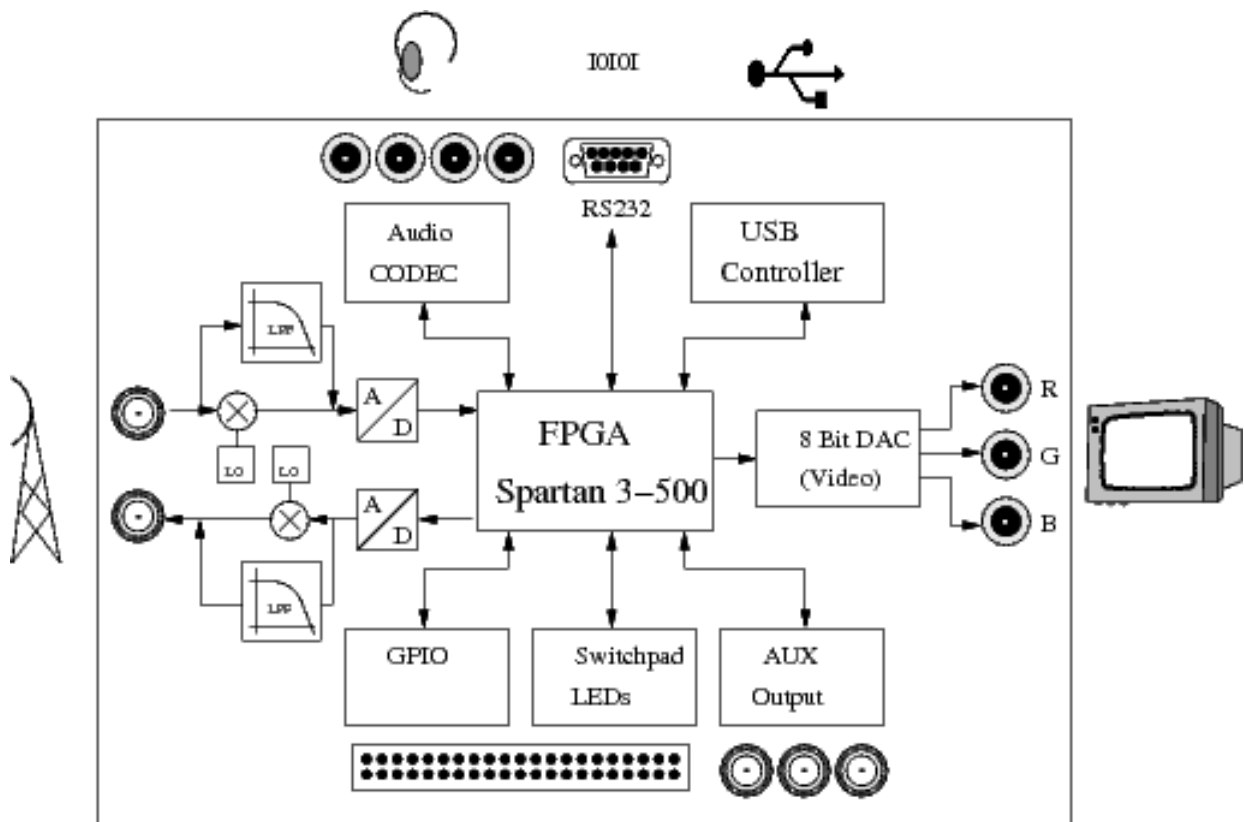


Abbildung 1: Blockdiagramm SDR-500

## 2 Erläuterung der Teilbaugruppen

Abbildung 2 zeigt die Anordnung der Teilbaugruppen auf der Platine, sowie die Position der Buchsen.

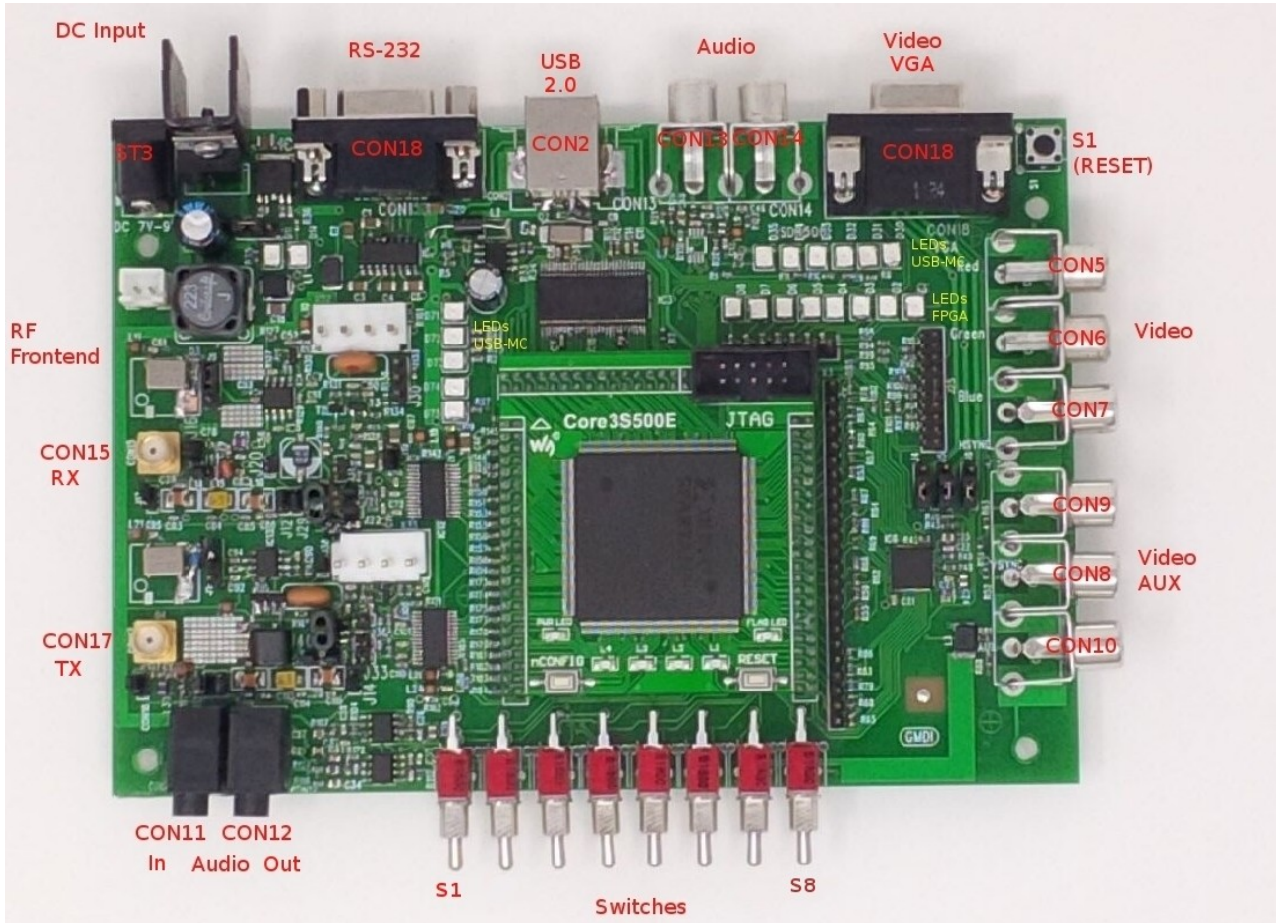


Abbildung 2: Anordnung der Teilbaugruppen

Die Position der Jumper finden Sie im Bestückungsdruck im Anhang.

### 2.1 FPGA

Als FPGA wird ein XILINX Spartan 3S500e verwendet, der auf einem Aufsteckboard sitzt. Dadurch besteht die Möglichkeit, künftige Aufsteckboards mit leistungsstärkeren FPGAs (z.B. Spartan-6) zu verwenden.

Die I/O-Pins des FPGAs sind über die Pinheader X1, X2 und X4 auf das Basisboard herausgeführt.

Die Betriebsspannungen 5 V und 3,3 V werden über das Basisboard bereitgestellt.

Neben dem 50 MHz-Takt (an I/O184) auf dem Aufsteckboard kann alternativ ein Taktsignal mit 54 MHz Frequenz an I/O177 des FPGA verwendet - oder extern (durch J34, J35) eingespeist werden.

Darüberhinaus befinden sich auf dem Aufsteckboard noch 4 LEDs, eine Power-

LED, die FLAG-LED, der RESET- und der nCONFIG-Taster.

## 2.2 GPIO

Als GPIO sind bestimmte Pins des FPGAs an X3 und X5 (nur Ausgänge) verfügbar.

Doppelbelegung X5:

X5 Pin	Signal Video	FPGA Pin
1	nicht belegt	nicht belegt
2	ROT0	I/O160
3	ROT1	I/O152
4	ROT2	I/O150
5	ROT3	I/O146
6	ROT4	I/O144
7	ROT5	I/O139
8	ROT6	I/O137
9	ROT7	I/O134
10	nicht belegt	nicht belegt
11	GRUEN0	I/O132
12	GRUEN1	I/O128
13	GRUEN2	I/O126
14	GRUEN3	I/O122
15	GRUEN4	I/O119
16	GRUEN5	I/O115
17	GRUEN6	I/O112
18	GRUEN7	I/O108
19	BLAU0	I/O106
20	BLAU1	I/O100
21	BLAU2	I/O98
22	BLAU3	I/O96
23	BLAU4	I/O93

Die Signale am Pinheader CON21 werden auch für die LEDs verwendet.

Doppelbelegung CON21:

X3 Pin	LED	FPGA Pin
1	LED0	I/O153
2	LED1	I/O151
3	LED2	I/O147
4	LED3	I/O145
5	LED4	I/O140
6	LED5	I/O138

7	LED6	I/O135
8	LED7	I/O133

### 2.3 Switches + LEDs

Es gibt 8 Switches und 8 LEDs, die am FPGA über Pins angeschlossen sind. Folgende Tabelle zeigt das Routing zum FPGA:

Switch	Signal	FPGA Pin
S1	SWTAST0	I/O123
S2	SWTAST1	I/O120
S3	SWTAST2	I/O116
S4	SWTAST3	I/O113
S5	SWTAST4	I/O109
S6	SWTAST5	I/O107
S7	SWTAST6	I/O102
S8	SWTAST7	I/O99
LED0	LED0	I/O153
LED1	LED1	I/O151
LED2	LED2	I/O147
LED3	LED3	I/O145
LED4	LED4	I/O140
LED5	LED5	I/O138
LED6	LED6	I/O135
LED7	LED7	I/O133

### 2.4 AUX I/O High-Speed

Die Chinch-Buchsen CON10, CON11 und CON12 sind an 50 Ohm angepasste I/O-Pins des FPGAs. Damit können hochfrequente digitale Signale an Koaxialleitungen herausgeführt werden.

Die Buchsen CON10 CON11 sind mit den VGA-Synchronsignalen verbunden.

Das Routing ist folgendermaßen:

Chinch-Buchse	Signal Video	FPGA Pin
CON10	HSYNC	I/O36
CON11	VSYNC	I/O40
CON12	AUX	I/O129

### 2.5 3-fach 8 Bit DAC (Video)

Der 3-fach DAC kann u.a. zur Ausgabe von VGA-Signalen verwendet werden.

Als DAC wird ein ADV7125 von Analog Devices verwendet.

Die analogen RGB-Signale liegen an der VGA-Buchse CON6 und an den Chinch-Buchsen CON7 ... CON11 an.

Das Routing ist folgendermaßen:

SUB-D Buchse Pin	Chinch-Buchse	Signal	FPGA Pin	Typ
1	CON7	ROT (R)	-	Analog
2	CON8	GRUEN (G)	-	Analog
3	CON9	BLAU (B)	-	Analog
	CON10	HSYNC	I/O36	Digital
13	CON11	VSYNC	I/O40	Digital

## 2.6 USB-Controller

Als USB-Controller wird ein Cypress CY7C68013A verwendet. Dieser erlaubt eine Datenübertragung nach USB High-Speed Standard. Die Übertragung zum FPGA erfolgt parallel über einen 16-Bit Datenbus (FD0 ... FD 15) über folgende Schnittstellensignale:

FPGA Pin	Signal / CY7C68013A
I/O4	FD0
I/O2	FD1
I/O203	FD2
I/O200	FD3
I/O197	FD4
I/O193	FD5
I/O190	FD6
I/O187	FD7
I/O178	FD8
I/O172	FD9
I/O168	FD10
I/O165	FD11
I/O164	FD12
I/O161	FD13
I/O162	FD14
I/O163	FD15
I/O5	FIFOADR0
I/O3	FIFOADR1
I/O205	FLAGA
I/O199	FLAGB
I/O202	FLAGC
I/O196	SLOE
I/O192	SLRD

I/O189	SLWR
I/O185	IF_CLK

CON3 ist als USB Typ-B Buchse ausgeführt.

## 2.7 RS232 / UART

Die Signale RX/TX an der SUBD-9 Buchse CON0 sind über einen Pegelwandler (MAX232A) mit dem FPGA verbunden.

Die Signale am FPGA sind:

FPGA Pin	Signal
I/O9	RX0
I/O12	TX0

## 2.8 Audio

Als Audio-Schnittstelle wird der 2-Kanal Delta-Sigma Codec PCM3060 von Texas Instruments verwendet. Er besitzt 24 Bit Datenwortbreite, bis 192 kS/s Abtastrate und je zwei Ein- und Ausgangskanäle.

Der Anschluss am FPGA erfolgt über eine I<sup>2</sup>S-Schnittstelle für die Audio-Daten mit folgendem Routing:

FPGA Pin	Signal
I/O25	DOUT
I/O31	LRCK
I/O34	MCLK
I/O29	SCLK
I/O55	DIN

Die Analogsignale werden über Operationsverstärker Tiefpass-gefiltert und gelangen an die 3,5 mm Klinkenbuchsen CON13 (Input) und CON14 (Output) nach außen.

## 2.9 HF-Frontend

Das HF-Frontend besteht aus einem Empfangs- (RX) und einem Sendepfad (TX), die gleichzeitig betrieben werden können. Die Signalwege sind mit Jumper konfigurierbar.

Die Signalwandlung Analog-Digital erfolgt mit IC12, AD9235A bzw. IC14, AD9762B.

Sowohl der RX- als auch der TX-Pfad sind mit separatem Quarz-Lokaloszillatoren bestückt. Über Jumper können auch externe LO-Signale eingespeist werden.



## **RX-Pfad:**

Es sind die Betriebsarten Basisband, Unterabtastung und Down-Converter möglich.

### Basisbandbetrieb:

Für den Basisbandbetrieb wird durch J22 und J21 ein Tiefpassfilter eingeschleift. Anschließend kann optional mit den Jumper J16, J13, J17/J18 ein Symmetrieübertrager zur besseren Signalperformance bei höheren Frequenzen eingeschleift werden.

### Unterabtastung:

(Optional ohne Bandpass)

Durch J20 und J19 ist ein Bandpassfilter einschleifbar, womit höherfrequente Signalbänder mit Unterabtastung selektiert werden können.

### Down-Converter:

Mittels J14 und J13 kann ein Down-Converter - bestehend aus dem Mischer IC11 SA612A und einem Bandpassfilter - eingeschleift werden. Das ZF-Signal (10,7 MHz) gelangt über einen Verstärker zum ADC.

Der Lokaloszillator schwingt auf einer Frequenz von 80 MHz. Die Empfangsfrequenz liegt damit 90,7 MHz. Alternativ kann auch ein externes Lokaloszillator-Signal eingespeist werden.

## **TX-Pfad:**

Es sind die Betriebsarten Basisband, Überabtastung und Up-Converter möglich.

### Basisbandbetrieb:

Für den Basisbandbetrieb wird durch J33, J31 ein Tiefpassfilter eingeschleift. Optional kann mit den Jumper J41, J31, J36 ein Symmetrieübertrager zur besseren Signalperformance bei höheren Frequenzen eingeschleift werden.

### Überabtastung:

(Optional ohne Bandpass)

Durch J17, CON16 ist ein Bandpassfilter einschleifbar. Das Signal wird mittels Überabtastung auf höherfrequente Signalbänder umgesetzt.

### Up-Converter:

Mittels J26 kann ein Up-Converter - bestehend aus dem Mischer IC13 SA612A und einem Bandpassfilter - eingeschleift werden. Das Ausgangssignal ergibt durch Umsetzung von 10,7 MHz und dem Lokaloszillator das Sendesignal.

Der bestückte Lokaloszillator besitzt eine Frequenz von 80 MHz. Die Sendefrequenz liegt damit bei 90,7 MHz. Alternativ kann auch extern ein Lokaloszillator-Signal eingespeist werden.

## **3 Inbetriebnahme**

Sie benötigen:

- FPGA-Entwicklungsumgebung Xilinx *ISE Design Suite*
- JTAG-Probe (z.B. Xilinx *Platform Cable USB II*)
- RS-232 Kabel

Das SDR-500 wird über das mitgelieferte Steckernetzteil mit Strom versorgt. Stecken Sie dieses in eine Steckdose und stecken Sie den Hohlstecker in die Buchse CON3. Als Kontrolle der Betriebsspannungen dienen die beiden LEDs D11 (+5 V, grün) und D14 (+3,3 V, grün).

Die Programmierung des FPGAs erfolgt über den am FPGA-Board mit "JTAG" gekennzeichneten Steckverbinder mit dem "XILINX Platform Cable USB" oder einer kompatiblen JTAG-Probe. Die Konfiguration des FPGA erfolgt mit der Entwicklungsumgebung "XILINX ISE". Für die folgenden Beispiele ist die frei verfügbare und kostenlose "Web Edition" ausreichend.

## 4 Beispiele

Die mitgelieferten Beispiele testen und erklären die Grundfunktionen der Teilbaugruppen des SDR-500 und dienen als Ausgangsbasis für eigene Projekte.

Laden Sie hierzu das Projekt "template.xise" in die Xilinx ISE Entwicklungsumgebung.

Die Beispiele sind in grafischer Eingabe und VHDL erstellt. Für das Mapping der FPGA-Pins ist ein User Constraint File (UCF) mitgeliefert. Abbildung 7 zeigt das Schaltbild *Top.sch* der FPGA-Konfiguration.

Das Kompilat kann mit dem Programm *iMPACT* auf den FPGA geladen werden.<sup>1</sup>



Installieren Sie die kostenlose Software *ISE Design Suite* von Xilinx (Download von [www.xilinx.com](http://www.xilinx.com), Suche nach *ISE*).

Zur Anwendung von ISE und *iMPACT* lesen Sie bitte die entsprechenden Dokumentation von Xilinx.

Die Position der in den Beispielbeschreibungen referenzierten LEDs, Schalter, Buchsen, etc. finden Sie farblich markiert in Abbildung 6.

### 4.1 Serielle Schnittstelle

In diesem Beispiel ist eine einfache, bidirektionale Kommunikation zwischen dem FPGA und einem RS232-Terminal implementiert.

Die Funktion ist im Block *serial* umgesetzt.

Beschreibung:

Schaltet man Schalter S4 von Stellung OFF auf ON, wird der Text "SDR-500",

---

<sup>1</sup> Aufrufbar über Tools -> *iMPACT*

gefolgt von einem Ton <BEL> und <CR><LF> ausgegeben.

Die Eingaben der Zeichen 0 bis 7 im Terminal lassen die entsprechenden LEDs 0 bis 7 aufleuchten.

☞ Stecken Sie ein RS-232-Kabel zwischen dem Terminal (oder PC mit Terminal-Emulation) und der Buchse CON0 des SDR-500 an.

Stellen Sie die Parameter des Terminals auf 9600 Baud, 8 Bit, No Parity.

Bringen Sie Schalter S4 in Stellung *OFF*.

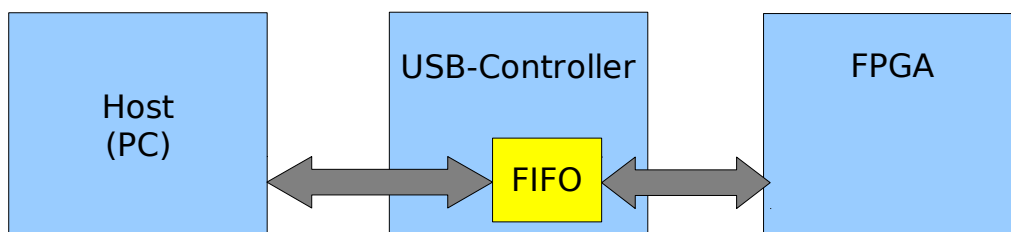
Stecken Sie das SDR-500 an.

☞ Bringen Sie Schalter S4 in Stellung *ON*.

Beobachten Sie die Ausgabe im Terminal. Es wird die Zeichenkette "SDR-500" ausgegeben. Geben Sie nun im Terminal eine beliebige Zahl 0 - 7 ein. Beobachten Sie das Aufleuchten der entsprechenden LED 1 - 8 auf dem SDR-500.

## 4.2 USB

Das USB-Beispiel ist mit dem USB-Controller CY7C68013A und dem FPGA umgesetzt. Beim USB-Controller wird dabei nur das FIFO verwendet. Die CPU selbst dient nur der Konfiguration. Die Datenübertragung erfolgt mit USB High-Speed.



Beschreibung:

Das Byte, das vom Host (z.B. PC) in den Endpoint 2 geschrieben wird, durchläuft das FIFO und steht dem FPGA am Datenbus zur Verfügung. Die Logik im FPGA liest dieses Byte und gibt die Bits auf die LEDs 0 bis 7 aus. Auf dem USB-Controller läuft das Programm *SlaveFIFO*, das das integrierte FIFO konfiguriert.

Die Kommunikation zwischen USB-Controller und FPGA erfolgt durch die Daten- und Steuersignale des FIFOs.

Die Kommunikation erfolgt gemäß der Beschreibung in Kapitel "Slave FIFO Asynchronous Read" - siehe [DATASHEET\_CY], Seite 47.

Die Fluss-Steuerung verwendet die Signale

- FLAGA (Signalisierung *empty* des FIFOs an den FPGA)

- SLRD (Lesanforderung des FPGA an das FIFO)
- SLOE (Aufforderung des FPGA an das FIFO, den Datenbus zu treiben)

Die Steuerung des FIFOs im USB-Controllers gibt diese Signale aus, um den Zustand *Empty* zu signalisieren und die Ausleseanforderungen des FPGAs entsprechend zu starten oder zu stoppen.

## 4.2.1 Installation am PC<sup>2</sup>

### 4.2.1.1 USB-Treiber

Um vom PC aus USB-Geräte anzusteuern, muss zuerst der passende Treiber für die Ansteuerung des USB-Controllers installiert sein. Darüberhinaus benötigen Sie die Software *CyConsole* zur Ansteuerung des USB-Controllers, sowie die kostenlose Entwicklungsumgebung *ISE Design Suite* von Xilinx.



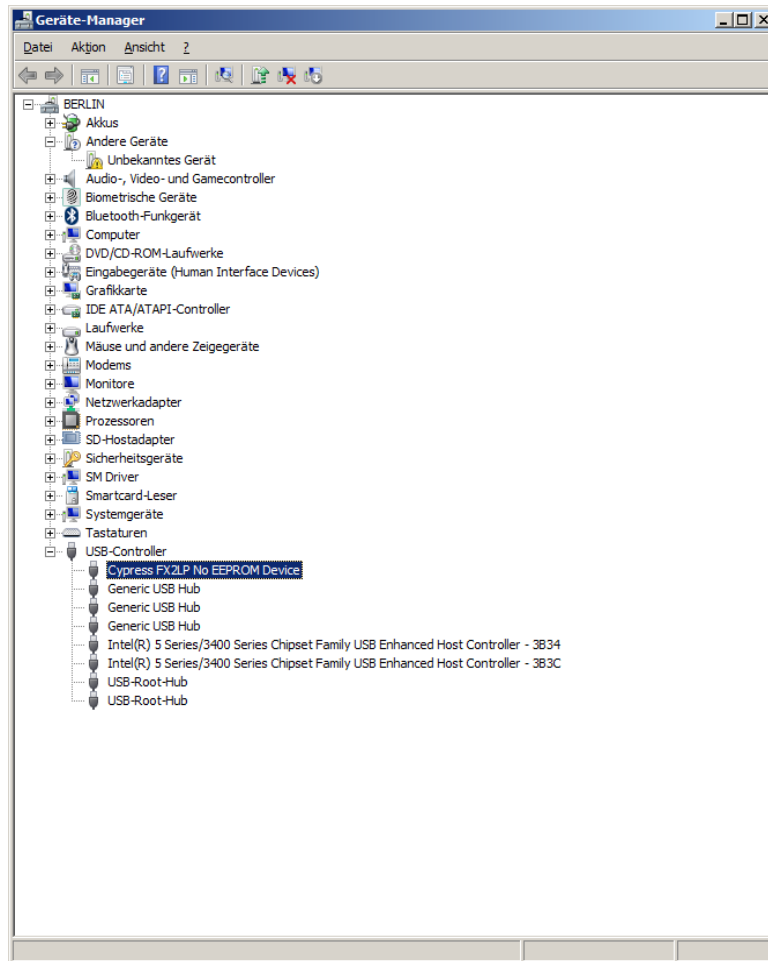
Installieren Sie dazu den im mitgelieferten Software-Paket enthaltenen Gerätetreiber *cyusb3*. Die Installation erfolgt im Gerätemanager manuell und wird im folgenden Abschnitt erklärt:

Grundsätzlich meldet sich der USB-Controller CY7C68013A nach RESET an einem Host mit den USB-Geräteeigenschaften (VID=0x04B4, PID=0x8613, Product=EZ-USB FX2, usw.). Lädt man eine Firmware ins RAM, so können die USB-Geräteeigenschaften beliebig **geändert** werden. Erstere gehört zur Gruppe der registrierten Hersteller. Daher versucht Windows bei der Erstanmeldung des USB-Geräts, einen passenden USB-Treiber im Internet zu suchen und automatisch zu installieren. Um den USB-Treiber mit den **geänderten** USB-Geräteeigenschaften auch zu installieren, muss die Installation manuell durchgeführt werden.

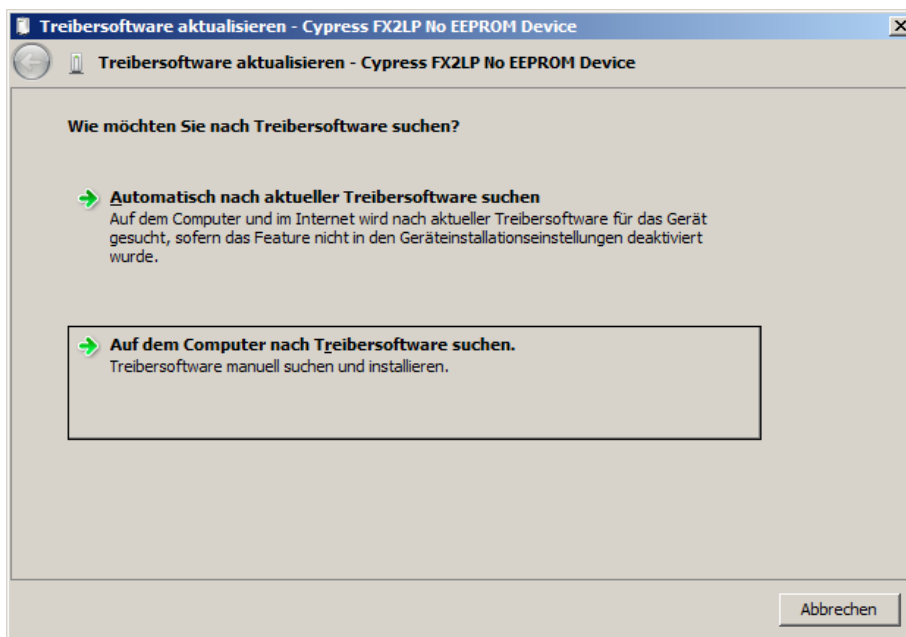
Der Treiber enthält die Dateien *cyusb3.sys*, *cyusb3.inf* und *cyusb.cat*, welche in der mitgelieferten Software enthalten sind. Schalten Sie das SDR-500 ein und lassen Sie eine automatische Installation des USB-Treibers durchführen - unabhängig davon, ob erfolgreich oder nicht. Öffnen Sie in den Systemeinstellungen den Gerätemanager und identifizieren das neue USB-Gerät.

---

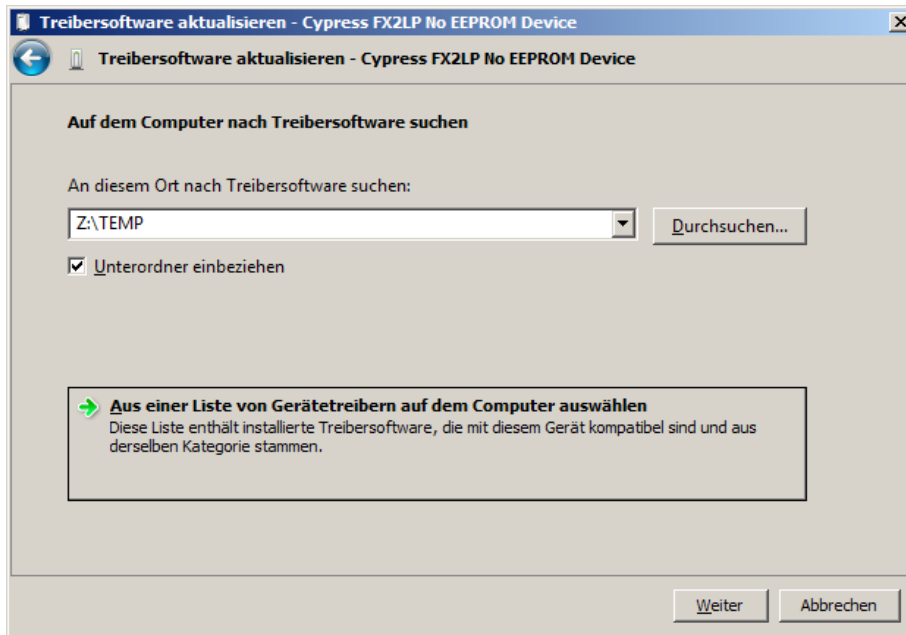
<sup>2</sup> Das Beispiel ist unter Microsoft Windows ab Version XP lauffähig. Andere Betriebssysteme werden von Cypress ebenfalls unterstützt.



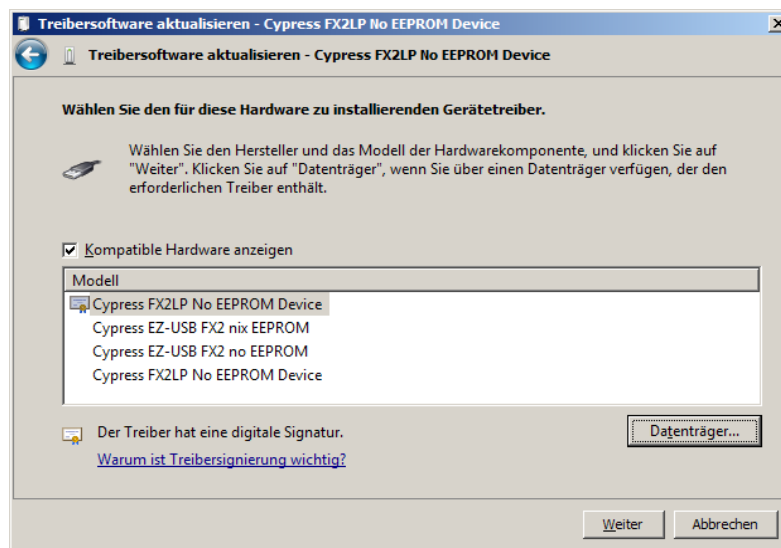
Öffnen Sie mit Rechtsklick das Kontextmenü und wählen "Treibersoftware aktualisieren". Im neuen Fenster wählen Sie "Auf dem Computer nach Treibersoftware suchen".



Im weiteren Fenster wählen Sie "Aus einer Liste von Gerätetreibern auf dem Computer auswählen".



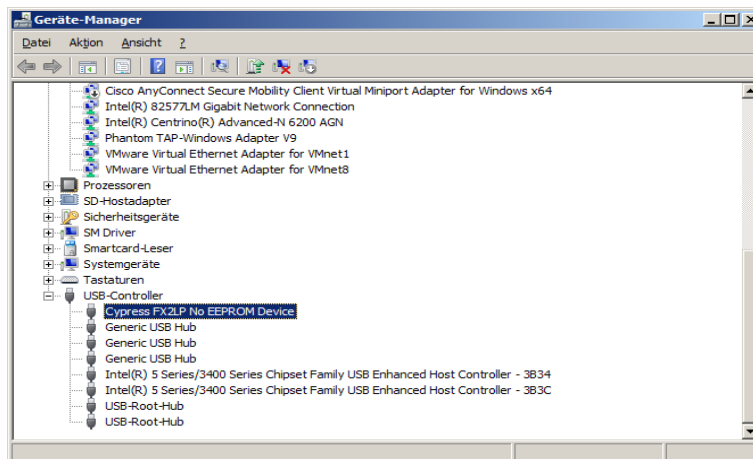
Im folgenden Fenster drücken Sie den Button "Datenträger..." .



In der sich öffnenden Auswahl suchen Sie nun in den mitgelieferten Software-Paket das Verzeichnis, in dem sich die für dieses Beispiel geänderten Dateien "cyusb3.\*" befinden. Wählen Sie passend für Ihre Betriebssystemversion die Datei "cyusb3.inf" und drücken anschließend den Button "OK". Im neuen Fenster wählen Sie das Modell "Cypress FX2LP No EEPROM Device" und drücken den Button *Weiter*.



Die Treibersoftware wird nun installiert. Eine Warnung, die besagt, dass der Treiber nicht signiert ist, quittieren Sie ggf. mit "Treiber trotzdem installieren".<sup>3</sup> Windows erfordert jetzt einen Neustart, den Sie jetzt durchführen müssen. Im Gerätemanager muss das installierte USB-Gerät nun fehlerfrei angezeigt werden.<sup>4</sup>



#### 4.2.1.2 USB-Applikation und FPGA-Designsoftware

Als USB-Applikation wird das Programm *CyConsole* von Cypress verwendet. Es erlaubt u.a. Firmware-Download zum RAM und EEPROM, sowie Datenübertragungen. Es befindet sich im mitgelieferten Software-Paket.



Starten Sie das ebenfalls mitgelieferte Software *CyConsole* zur Ansteuerung des USB-Controllers.

- 3 Grund: In der Datei cyusb.inf sind die zum USB-Beispiel zugehörige USB-Geräteeigenschaften VID=0x0547, PID=0x1002 eingetragen, welche nicht zertifiziert sind. Das ist nur dann von Belang, wenn das System als Gerät öffentlich verkauft wird. Für die Beispiele ist das unerheblich.
- 4 Ein Ausrufezeichen am Symbol deutet auf einen Fehler hin. Der Gerätetreiber wird in diesem Fall nicht funktionieren. Wiederholen Sie in diesem Fall die Treiberinstallation und exakt nach Beschreibung vor.



Starten Sie Xilinx ISE.

#### **4.2.2 Programmierung der Firmware des USB-Controllers**

Der USB-Controller CY7C68013A enthält ein RAM, worin die Firmware geladen werden muss. Die Daten sind flüchtig, d.h. nach RESET gehen alle Daten verloren und die Firmware muss erneut geladen werden. Alternativ kann diese auch in das auf dem Board befindliche EEPROM gespeichert werden. Der USB-Controller erkennt ein gültiges Programm im EEPROM und bootet dieses nach RESET.

Die Firmware führt eine Initialisierung des FIFOs durch.

Die Firmware ist mit einem Software-Projekt unter der Entwicklungsumgebung *KEIL C51* erstellt worden und kann beliebig verändert bzw. erweitert werden. Für dieses Beispiel ist aber lediglich die Binärdatei *Slave\_FIFO.hex* in den USB-Controller zu laden.

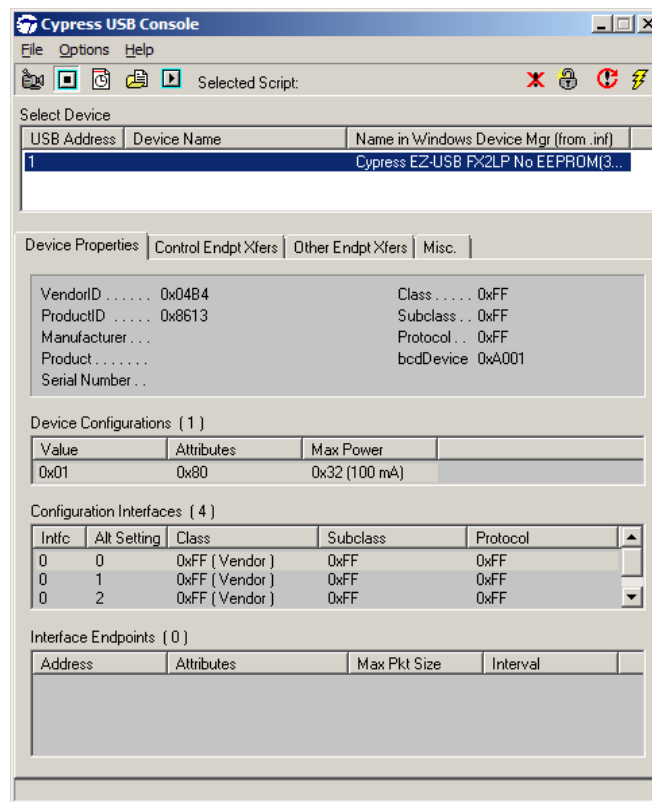


Starten Sie das Programm *CyConsole*





Verbinden Sie das SDR-500 mit dem PC durch das mitgelieferte USB-Kabel und schalten Sie das SDR-500 ein

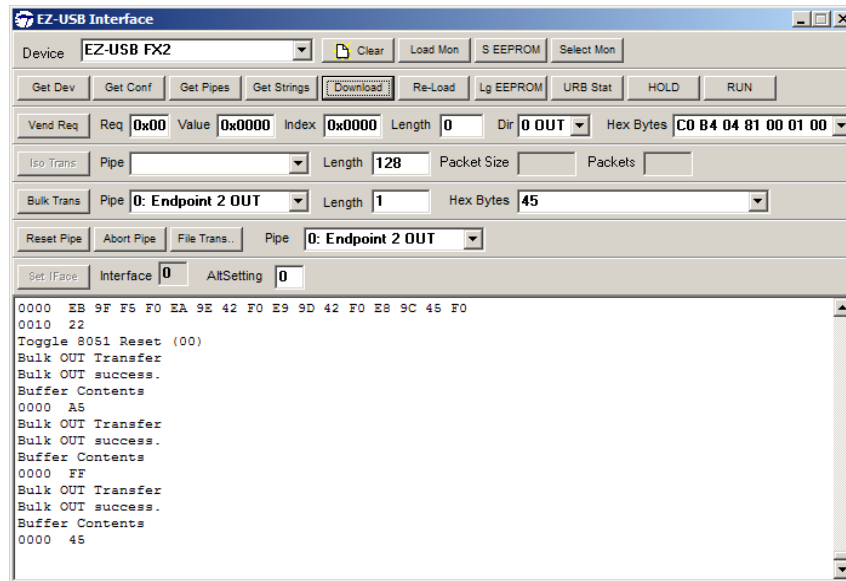


Die Daten im Reiter *Device Properties* (VendorID, ProductID, Manufacturer, ... ) müssen wie im Bild gezeigt erscheinen.



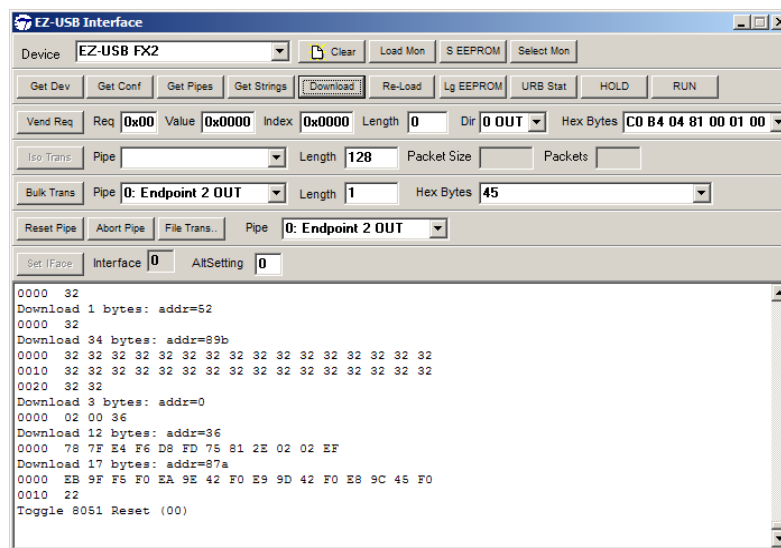
Clicken in der *CyConsole* auf *Options* -> *EZ-USB Interface*

Es öffnet sich das gezeigte Fenster.

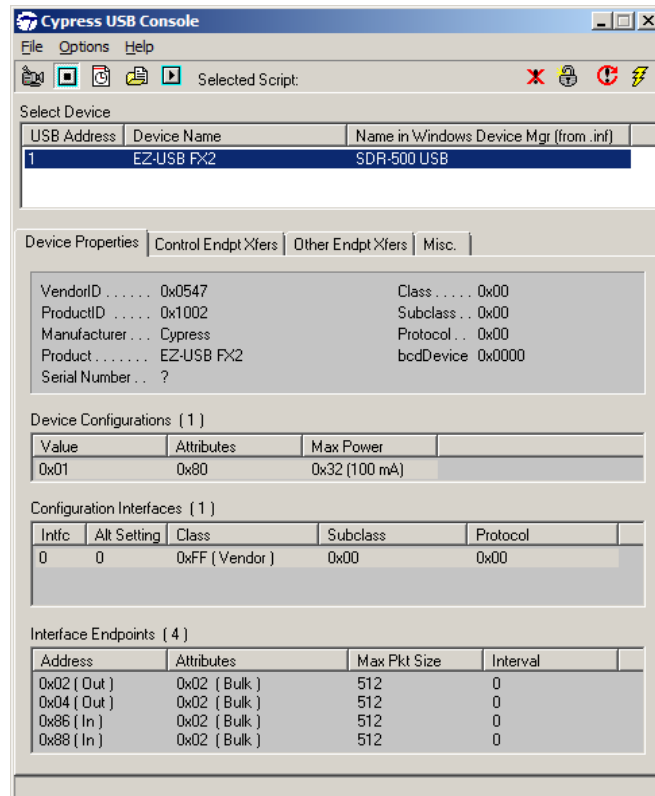


Drücken Sie den Button *Download* und wählen die Datei *Slave-FIFO.hex*. Für ein Programmieren des EEPROM drücken Sie *Lg EEPROM* und wählen die Datei *Slave-FIFO.iic*.

Im Ausgabefeld sollten die Programmdateien ausgegeben werden. Das Fenster sieht danach folgendermaßen aus:



Im USB-Controller läuft nun das geladene Programm. Es erfolgt nun eine Neuanmeldung des USB-Devices unter mit neuen Geräteparametern. Im Gerätemanager wechselt der Name zu "SDR-500", wie im folgenden Bild gezeigt.



### 4.2.3 FPGA-Konfiguration laden

Im Beispiel wird folgende Einheit konfiguriert:

Die Funktion ist im Block *USB\_LEDdemo* umgesetzt.

Eingang *FLAGA* signalisiert, dass das FIFO des USB-Controllers leer ist. Schreibt der Host ein Byte zum USB-Controller, so wird *FLAGA* inaktiv. Ein Prozess im FPGA führt dann das Auslesen des FIFO durch, indem die Fifo-Adresse (*FIFO-ADR*) auf 0b00 gesetzt wird und *SLOE* und *SLRD* aktiviert werden. Das vom Host geschriebene Byte liegt nun am Datenbus *FIFODATA* an und es werden die LSBs 3:0 an die LEDs (*led*) ausgegeben. Nach Rücksetzen der Steuersignale *SLRD* und *SLOE* wird *FLAGA* wieder gesetzt und der Host kann ein neues Byte schreiben.

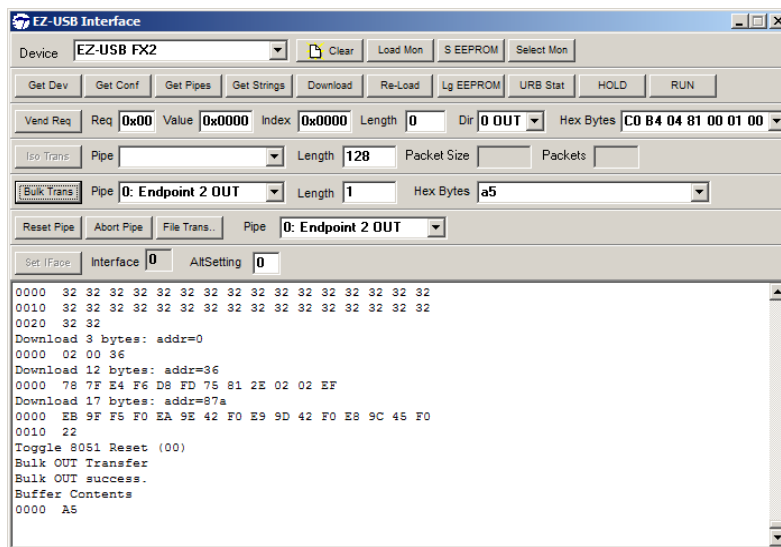
### 4.2.4 Test



Starten Sie Xilinx *ISE Design Suite* und laden Sie das FPGA-Projekt



Übertragen Sie nun in der *CyConsole* ein Byte und kontrollieren Sie die LEDs auf dem SDR-500. Stellen Sie in der Zeile neben dem Button *Bulk-Trans Pipe 0: Endpoint 2 OUT, Length 1 Hex Bytes 01 (bzw. 02, 03, 04 oder 05)* und drücken den Button *Bulk-Trans*. Im Textfenster im unteren Bereich muss das übertragene Byte wie folgt gezeigt werden:



Die LEDs auf dem FPGA-Board L1 ... L4 müssen entsprechend aufleuchten:

Hex Bytes	LED Anzeige
01	L1 leuchtet
02	L2 leuchtet
03	L3 leuchtet
04	L4 leuchtet
05	Lauflicht L1 ... L4

### 4.3 VGA

Dieses Beispiel erzeugt ein Videosignal, welches über den Video-DAC an der VGA-Buchse CON6 ausgegeben wird. Das Testbild kann über einen PC-Monitor oder Fernsehgerät mit VGA-Buchse betrachtet werden. Als Testbild gibt es folgende Varianten:

- Oszilloskop
- Farbbalken

Die Funktion ist in den Blöcken *oszi* und der mit *Audio\_Source* bezeichneten Baugruppe umgesetzt.

Schalter S7 bestimmt die Funktion Oszilloskops (OFF) oder Farbbalken (ON).

Schalter S2 bestimmt die Signalquelle für das Oszilloskop. Es gibt die Auswahl zwischen einem internen Sinusgenerator und der Eingangsbuchse CON13.

Der Trigger des Oszilloskops ist auf positiven Nulldurchgang fest eingestellt.



Stecken Sie ein VGA-Kabel zwischen Buchse CON6 und einem Monitor an.

Bringen Sie Schalter S7 auf Stellung ON.

Stecken Sie das SDR-500 an.

Auf dem Monitor erscheint das Farbbalken-Testbild.



Bringen Sie Schalter S7 auf Stellung OFF.

Bringen Sie Schalter S2 auf Stellung ON.

Das Monitorbild wechselt nun in die Oszilloskop-Darstellung. Es wird das interne Sinussignal angezeigt.



Verbinden Sie eine Signalquelle mit CON13.

Bringen Sie Schalter S2 auf Stellung OFF.

Die Anzeige wechselt jetzt auf das Signal an Buchse CON13.

#### **4.4 Audio**

Dieses Beispiel erzeugt ein Duoton-Signal mit 1 kHz und 2 kHz und routet dieses oder wahlweise das Eingangssignal an Buchse CON13 an die Ausgangsbuchse CON14 durch.

Die Funktion ist in dem mit *Audio\_Source* bezeichneten Modul umgesetzt.

Schalter S1 schaltet zwischen dem internen Signalgenerator und dem Eingangssignal (Durchschleifen) um.

Schalter S2 schaltet den internen Sinusgenerator oder das eingespeiste Audio-Signal an den Ausgang.



Schließen Sie ein Audio-Wiedergabegerät (z.B. Aktivlautsprecher, Kopfhörer, PC) an die Ausgangsbuchse CON12 an.

Bringen Sie Schalter S1 auf Stellung OFF.

Bringen Sie Schalter S2 auf Stellung ON.

Stecken Sie das SDR-500 an.

Sie hören ein Duoton-Signal mit 1 kHz im linken - und 2 kHz im rechten Kanal.



Stecken Sie ein Audio-Kabel zwischen einer Audio-Quelle und der Eingangsbuchse CON11 an.

Bringen Sie Schalter S1 auf Stellung OFF.

Bringen Sie Schalter S2 auf Stellung OFF.

Stecken Sie das SDR-500 an.

Sie hören das unveränderte Signal der Audio-Quelle.

#### **4.5 HF RX/TX**

Dieses Beispiel realisiert einen FM-Sender im UKW-Frequenzbereich. Im FPGA

wird ein Trägersignal mit einem Audiosignal FM-moduliert und am ADC ausgegeben. An der TX-Buchse *CON15* liegt das hochgesetzte HF-Signal an.

FPGA-Teil:

Die Funktion ist in den Modulen *FM-Modulator* und *Audio-Source* realisiert und wird über einen Multiplexer an den DAC-Ausgang geführt.

Der Multiplexer befindet sich am Ausgang und kann zwischen dem FM-modulierten Signal, einem 1 MHz Sinussignal und dem Eingangssignal vom ADC (Durchschleifen) umschalten.

Der FM-Modulator moduliert das Audio-Signal auf eine Trägerfrequenz von 10,7 MHz.

Als Eingangssignal des Modulators ist ein 1 kHz Sinussignal oder das Signal von der Audio-Signalquelle umschaltbar.

Schalter S2 wählt zwischen dem internen Sinusgenerator oder dem extern eingespeisten Audio-Signal als Quelle.

Hardware-Teil:

Nach der D/A-Wandlung durch IC14 gelangt das Signal zum Up-Converter, der das Signal mit der LO-Frequenz von 80 MHz auf 90,7 MHz mischt. Das Bandpass-gefilterte HF-Signal ist an Buchse *CON17* herausgeführt.



Konfigurieren Sie folgende Jumper:

J24 --> geschlossen

J26 --> geschlossen

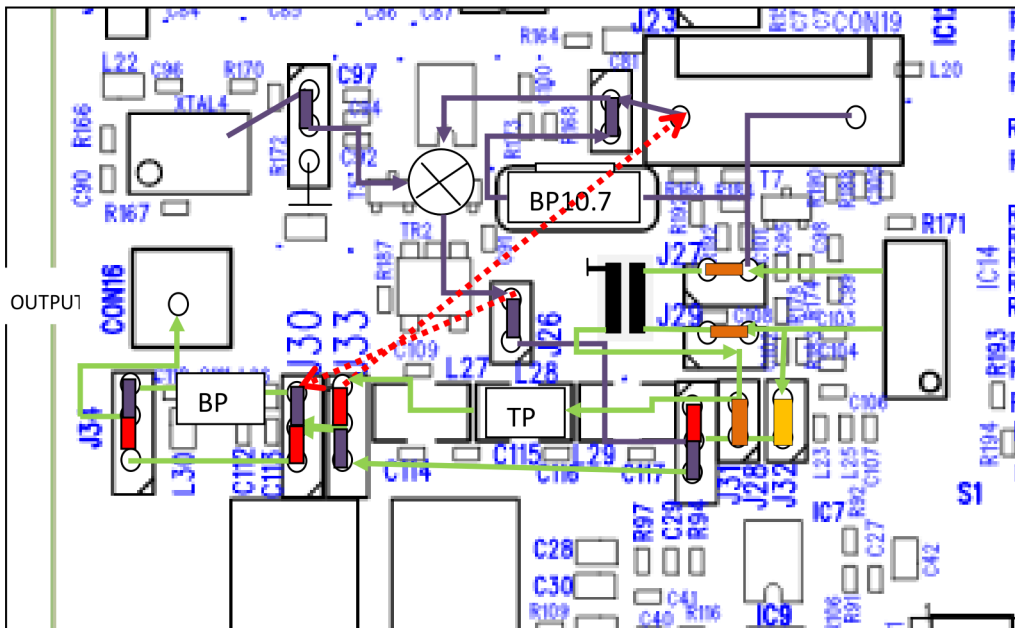
J25 --> 1-2 verbunden

J28, J32 --> offen

J31, J33 --> 2-3 verbunden

J30, J34 --> 1-2 verbunden

Folgende Grafik hilft bei der Orientierung auf der Platine. Verbinden Sie in Farbe Magenta ■ markierten Jumper.



Schließen Sie ein Audio-Wiedergabegerät (z.B. Aktivlautsprecher, PC) an die Ausgangsbuchse CON12 an.

Bringen Sie Schalter S2 auf Stellung ON.

Schließen Sie an der HF TX-Buchse CON17 ein Empfangsgerät (z.B. Radioempfänger, Spectrum Analyzer oder Antenne) an. Schalten Sie dieses auf FM und wählen Sie als Mittenfrequenz 90,7 MHz.

Stecken Sie das SDR-500 an.

Sie hören im Empfangsgerät ein Signal mit 1 kHz.



Bringen Sie Schalter S2 auf Stellung OFF.

Sie hören im Empfangsgerät das Signal der eingespeisten Audio-Signalquelle.

# Anhang

## Schaltplan

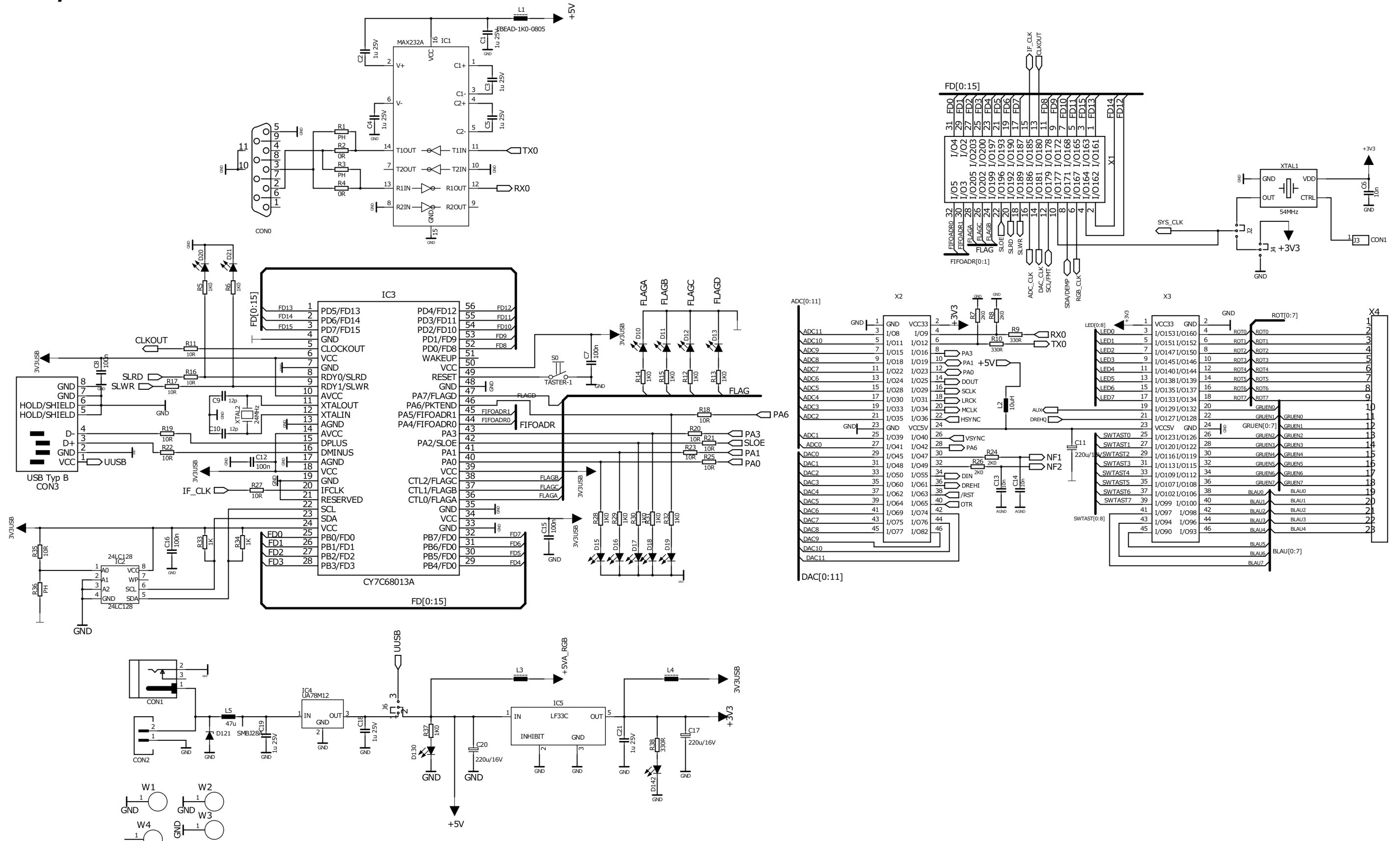


Abbildung 3: Schaltplan SDR-500 Rev. 3, S. 1/3



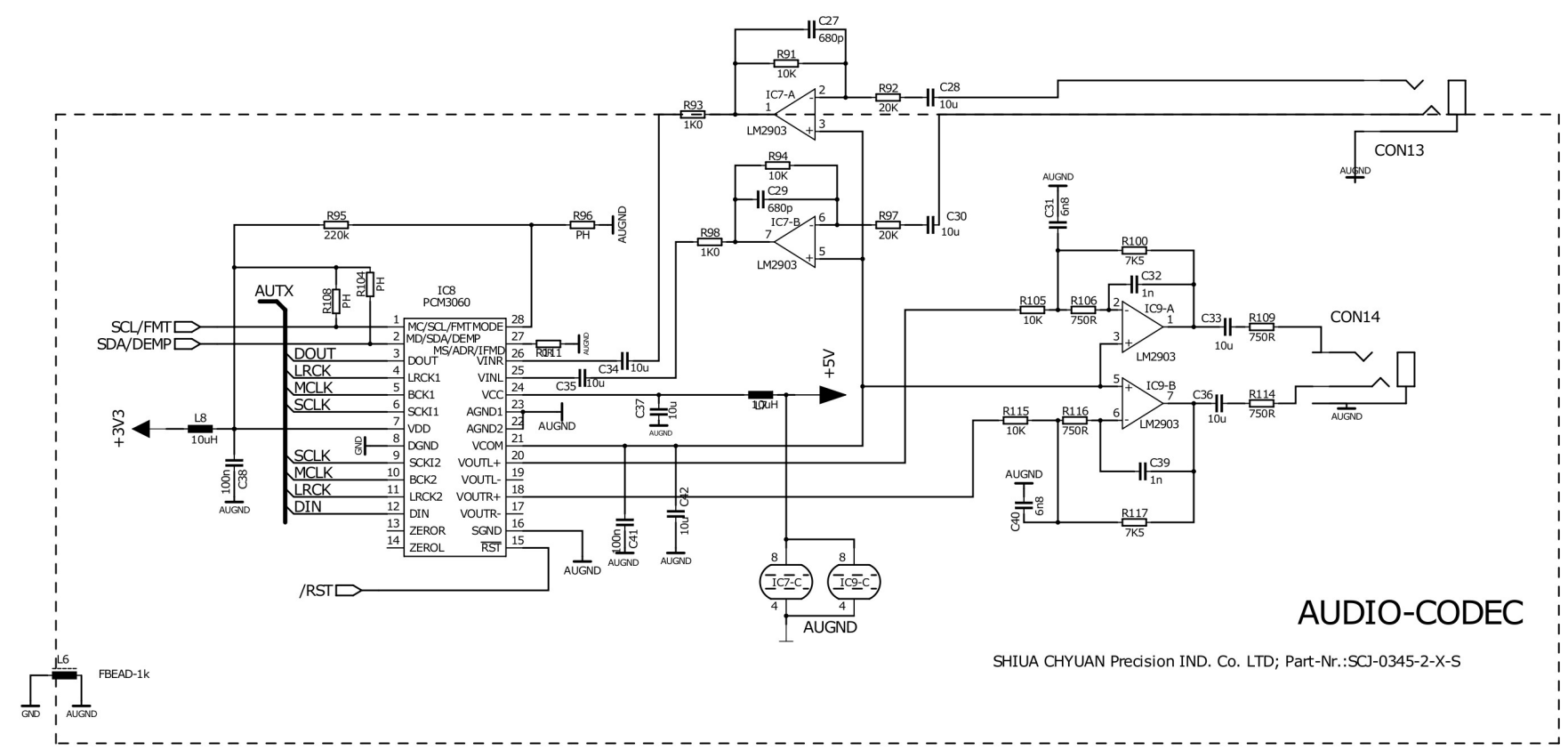
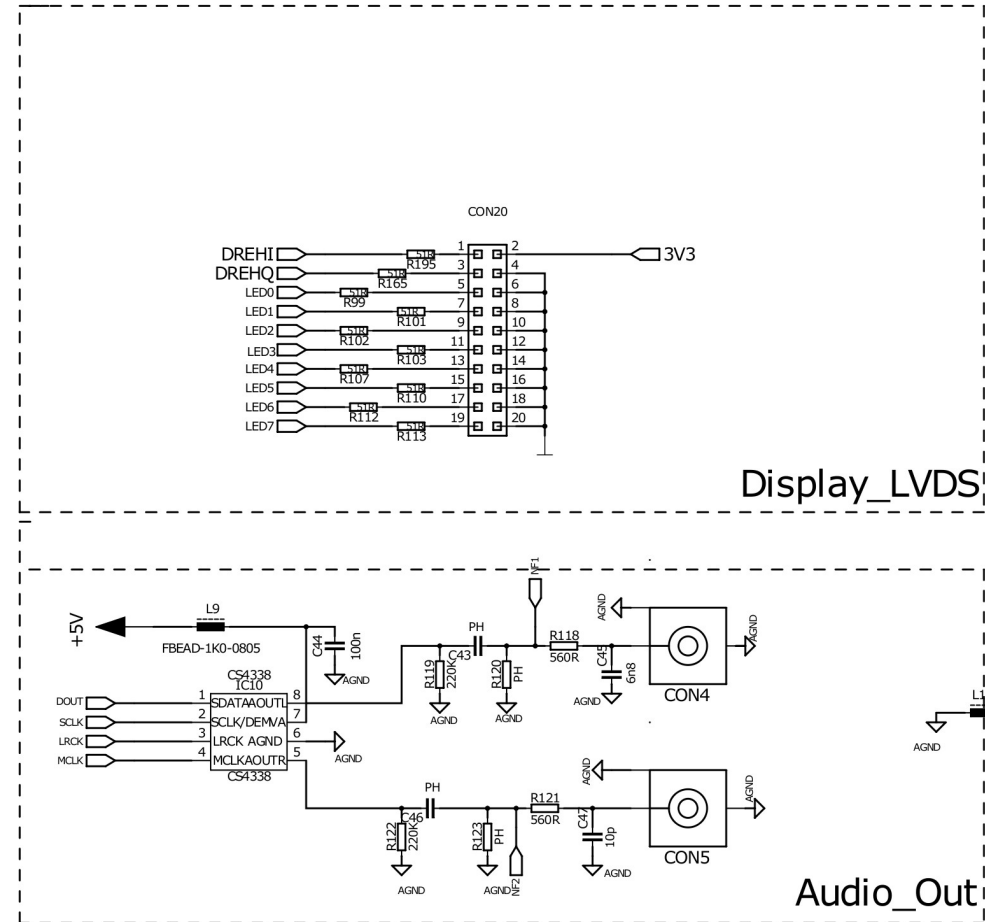
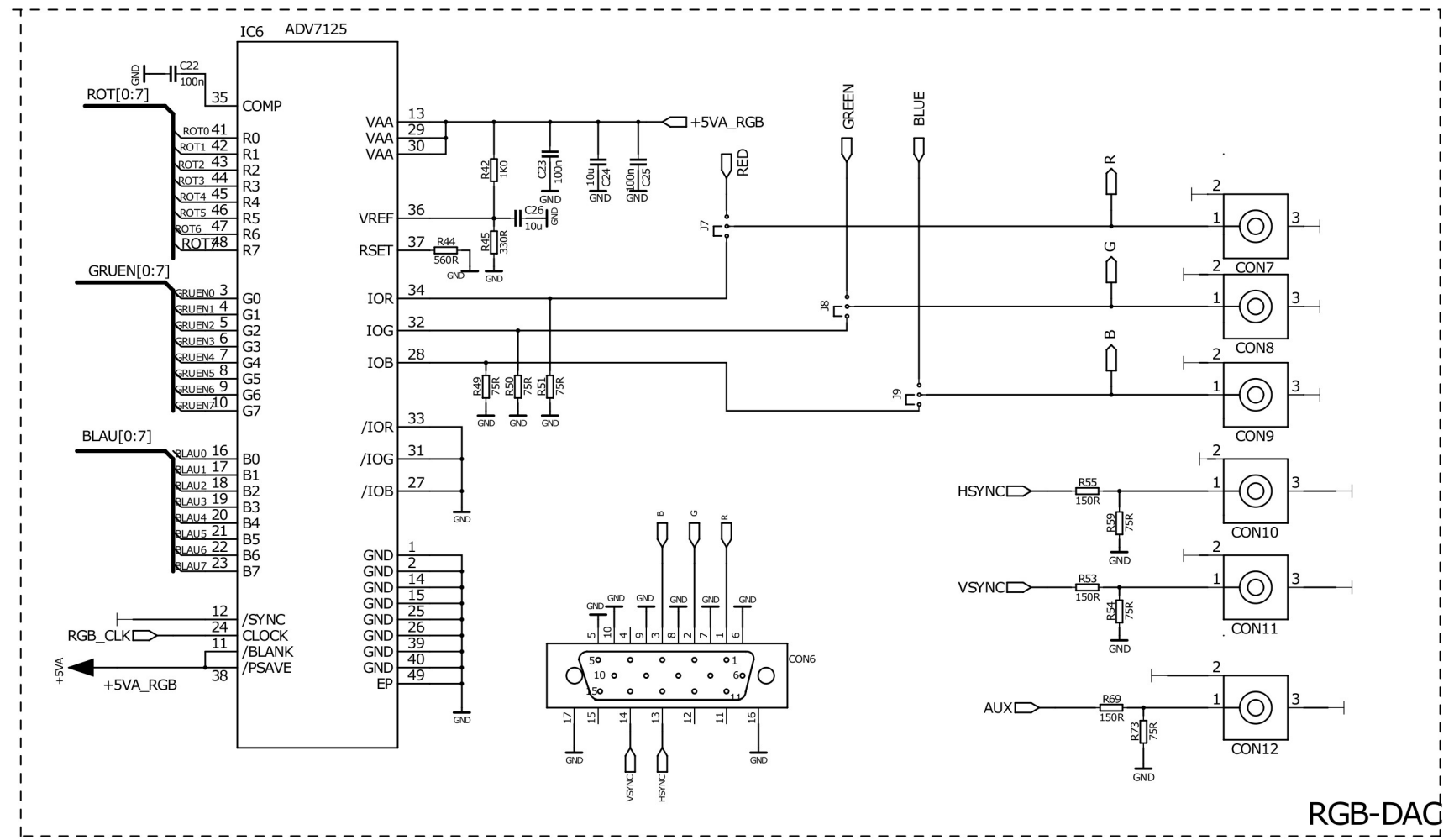
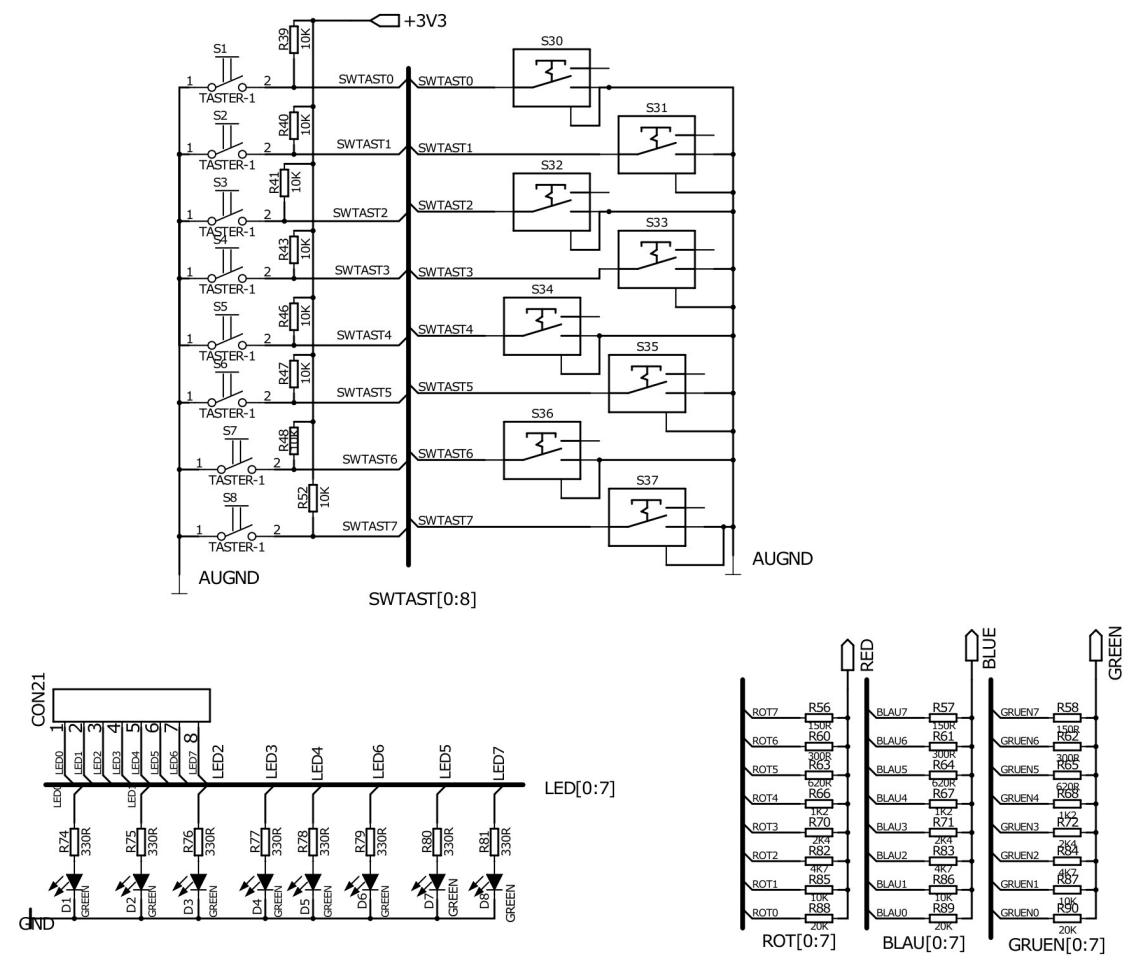


Abbildung 4: Schaltplan SDR-500 Rev. 3, S. 2/3

SHIUA CHYUAN Precision IND. Co. LTD; Part-Nr.:SCJ-0345-2-X-S

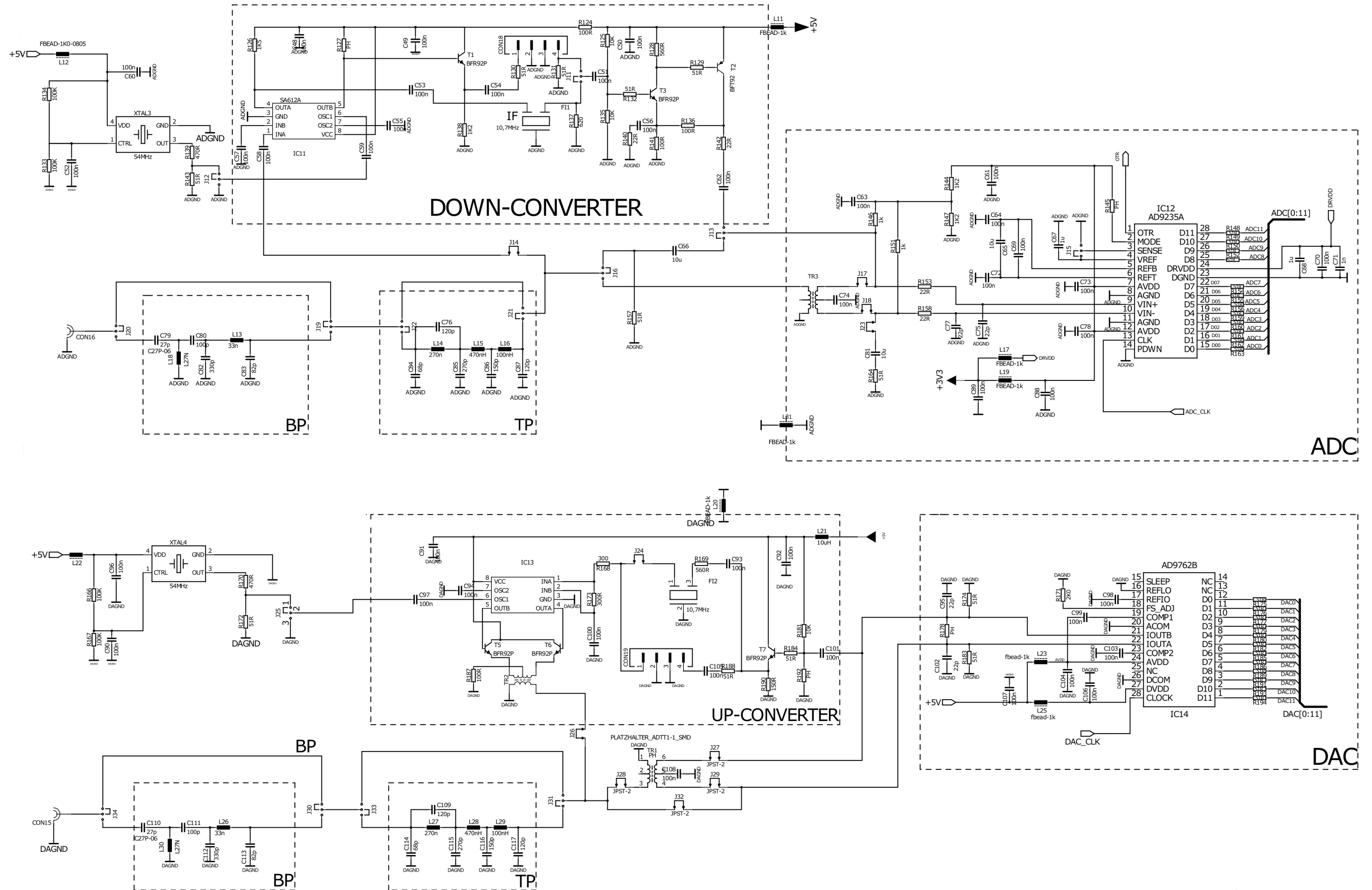


Abbildung 5: Schaltplan SDR-500 Rev. 3, S. 3/3

# Bestückungsdruck

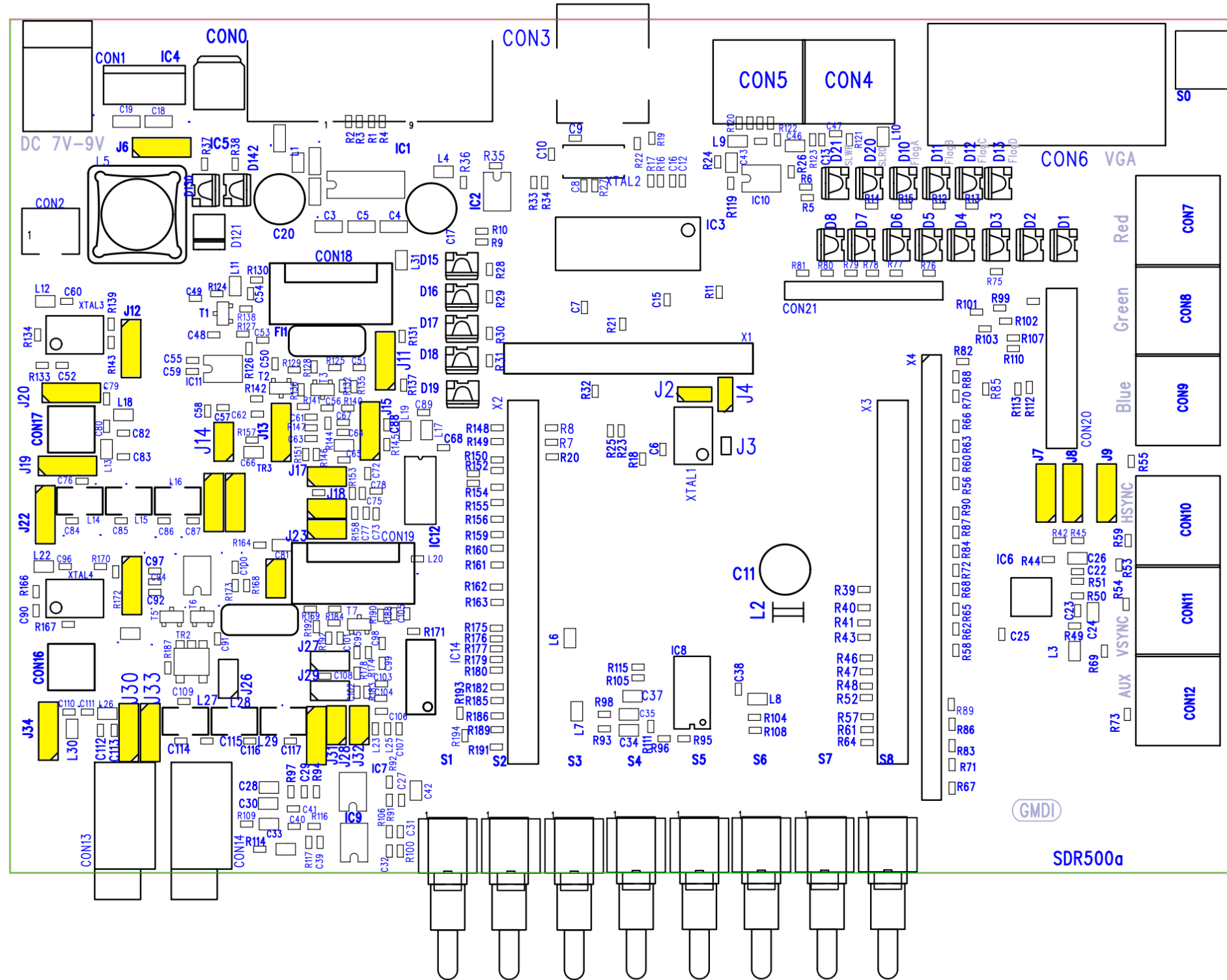


Abbildung 6: Bestückungsdruck SDR-500 Rev. 3

# Schaltbild FPGA-Konfiguration

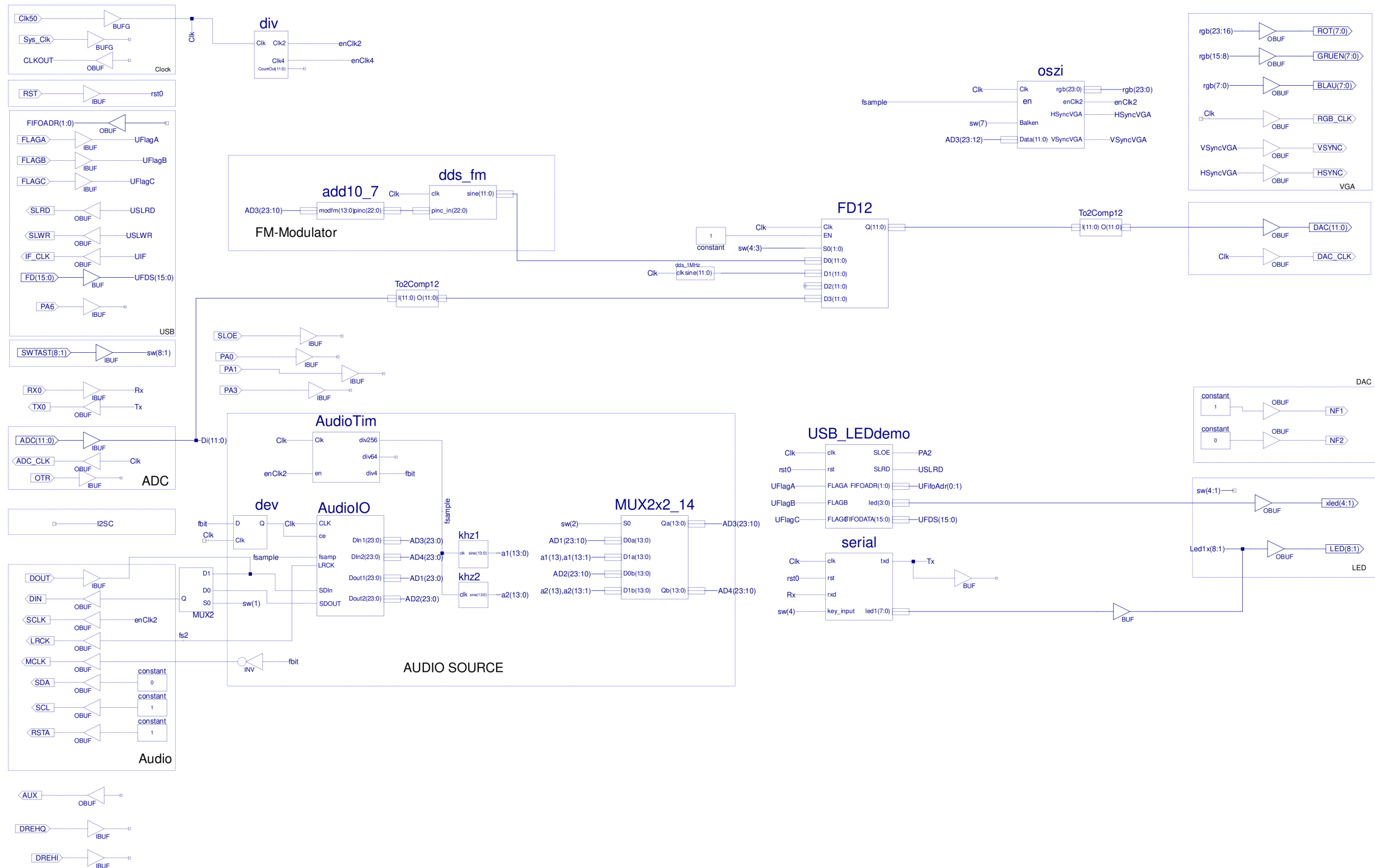


Abbildung 7: Schaltbild FPGA-Konfiguration

## Abbildungsverzeichnis

Abbildung 1: Blockdiagramm SDR-500.....	3
Abbildung 2: Anordnung der Teilbaugruppen.....	4
Abbildung 3: Schaltplan SDR-500 Rev. 3, S. 1/3.....	24
Abbildung 4: Schaltplan SDR-500 Rev. 3, S. 2/3.....	25
Abbildung 5: Schaltplan SDR-500 Rev. 3, S. 3/3.....	26
Abbildung 6: Bestückungsdruck SDR-500 Rev. 3.....	27
Abbildung 7: Schaltbild FPGA-Konfiguration.....	28

## Literaturverzeichnis

[DATASHEET\_CY]      Datenblatt  
CY7C68013A, CY7C68014A, CY7C68015A, CY7C68016A,  
EZ-USB® FX2LPTM USB Microcontroller High-Speed USB  
Peripheral Controller  
Rev. \*Z, April 26, 2017  
Document Number: 38-08032  
Cypress Semiconductor Corporation



*... Made in Germany*

Signalgeneratoren



Tastköpfe



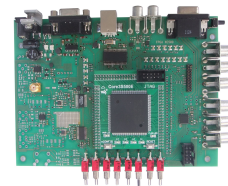
Präzisionsverstärker



Messbereichserweiterungen



DSP-Systeme



HUBER SIGNAL PROCESSING  
Unternbergstr. 13  
83278 Traunstein  
Germany

Tel: +49 (0)861.213978-0  
Fax: +49 (0)861.213978-40

[sales@huber-signal.com](mailto:sales@huber-signal.com)  
[www.huber-signal.com](http://www.huber-signal.com)

Doc. Nr.: 0125-002  
v1.1, 06.10.2020, Änderungen vorbehalten  
© HUBER SIGNAL PROCESSING