USB to I²C/SPI Communcation Interface for use with SigmaStudio

EVAL-ADUSB2EBZ

FEATURES

- USB to I²C and SPI conversion
- Compatibility with 1.8V and 3.3V target devices
- SigmaStudio integration for most SigmaDSP processors
- On-board power regulators capable of supplying target board
- Standard Aardvark-compatible programming header
- SPI control of up to 5 slave devices
- Low-profile surface mount USB Mini B connector
- Plug-and-play operation

APPLICATIONS

- Downloading code and register settings to SigmaDSP processors and codecs with SigmaStudio
- Real-time tuning of SigmaDSP production units with SigmaStudio

FUNCTIONAL BLOCK DIAGRAM

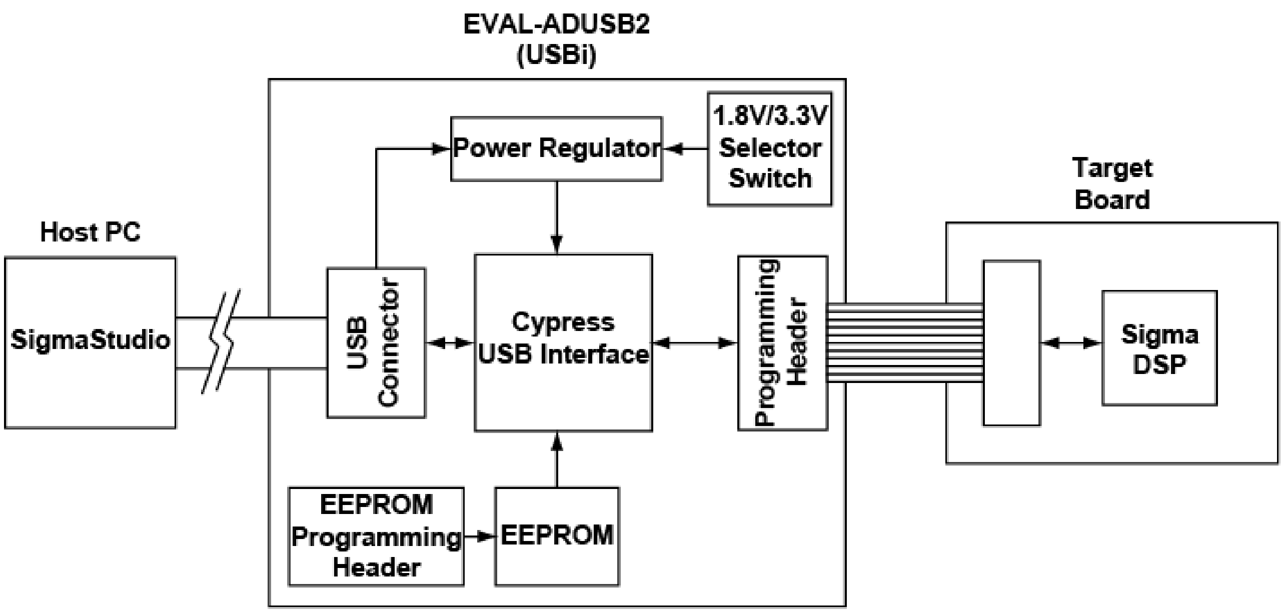


Figure 1. Functional Block Diagram

GENERAL DESCRIPTION

The EVAL-ADUSB2EBZ, also known as the USBi, is a standalone communications interface and programmer for SigmaDSP systems. It translates USB control commands from SigmaStudio to the I²C and SPI communications protocols. The USBi is powered over the USB cable, so no external power supply is required.

The ribbon cable and 10-pin header form a bridge to the target board to connect the communications signals to the target IC. The ribbon cable also carries +5 V power from the USB hub, which can be used to power the target board if desired.

The onboard regulators enable both 1.8 V and 3.3 V IOVDD operation, allowing for increased compatibility with target devices.

Up to five slave devices can be controlled by the USBi simultaneously. To control multiple SPI devices, additional latch signals are provided, although they are not connected to the ribbon cable.

The USBi can be used to control SigmaDSP systems in real-time via SigmaStudio, and is capable of programming an EEPROM in self-boot systems. It is an ideal solution for in-circuit programming and tuning of prototype systems.

Rev. PrA

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.461.3113

TABLE OF CONTENTS

REVISION HISTORY

8/24/07 – PrA – Initial Revision

USING THE USB INTERFACE WITH SIGMASTUDIO

INSTALLING THE DRIVERS

SigmaStudio must be installed in order to use the USBi. Once SigmaStudio has been properly installed, connect the USBi to an available USB port with the included USB cable. At this point, Windows XP will recognize the device, and prompt the user to install drivers.

Figure 2. Windows Found New Hardware Notification

Choose "Install from a list of specific location (Advanced)."

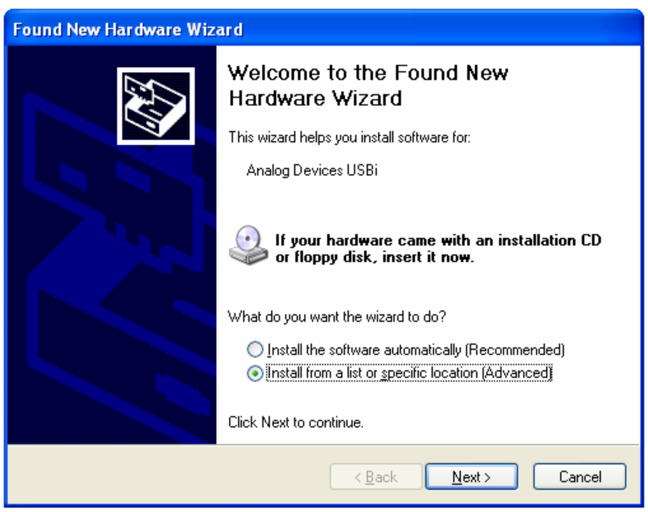


Figure 3. Windows Found New Hardware Wizard Screen 1

Choose "Search for the best driver in these locations," then check "Include this location in the search," and then browse to the SigmaStudio 3.0\USB drivers directory.

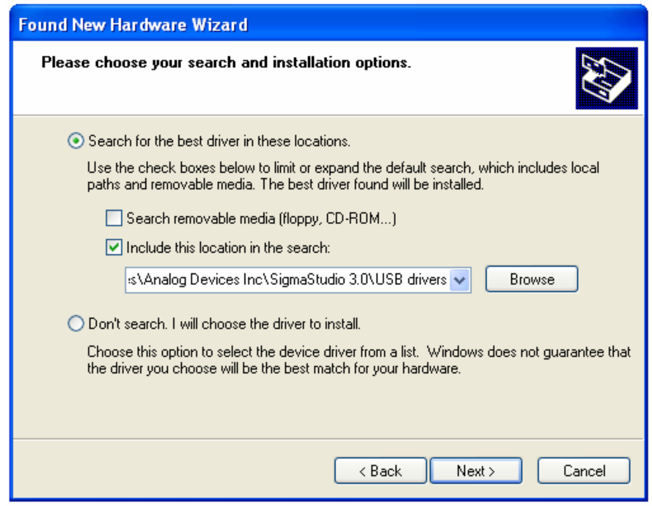


Figure 4. Windows Found New Hardware Wizard Screen 2

When the warning about Windows Logo testing appears on the screen, click "Continue Anyway."

Figure 5. Windows Logo Testing Warning

ADDING THE USBI TO A SIGMASTUDIO PROJECT

To use the USBi in conjunction with SigmaStudio, first select it in the Communication Channels subsection of the Toolbox in the Hardware Configuration tab, and add it to the project space.

Figure 6. Adding the USBi Communication Channel

If SigmaStudio cannot detect the USBi on the computer's USB port, then the background of the "USB" label will be red. This may happen when the USBi is not connected, or when the drivers are incorrectly installed.

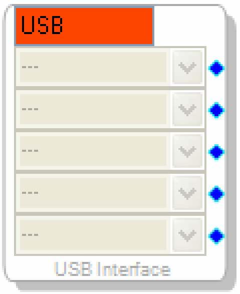


Figure 7. USBi Not Detected by SigmaStudio

If SigmaStudio detects the USBi on the computer's USB port, the background of the "USB" label will change to orange.

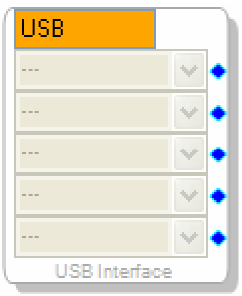


Figure 8. USBi Detected by SigmaStudio

CONFIGURING THE USBI TO COMMUNICATE WITH AN IC

To use the USBi to communicate with the target IC, it needs to be connected by click-dragging a wire between the USBi's blue pin and the IC's green pin. The USBi's corresponding drop-down menu will automatically be filled with the default mode and channel for that IC.

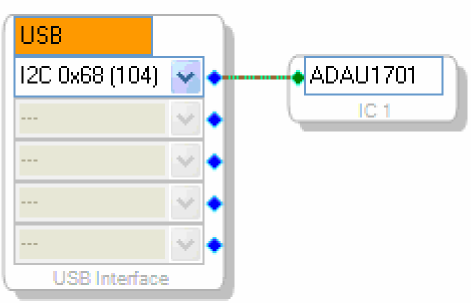


Figure 9. Connecting the USBi to an IC

To change the communication mode and channel, click the drop down menu and select the appropriate mode and channel from the list.

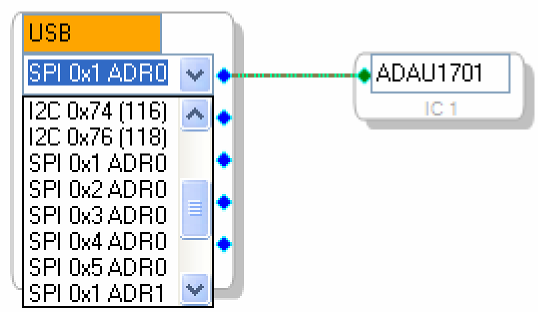


Figure 10. Selecting the Communications Mode and Channel

CONFIGURING THE USBI TO COMMUNICATE WITH MULTIPLE ICS

The USBi can communicate with up to 5 ICs simultaneously. To communicate with more than one IC, simply add another IC to the project and connect it to the USBi's next available pin.

Multiple Address Operation with I²C

The USBi can support up to 4 identical devices on the same bus, as long as the ADDR0 and ADDR1 pins of the target devices are independently set to 4 different addresses, matching the addresses in the drop-down menu in the Hardware Configuration tab of SigmaStudio.

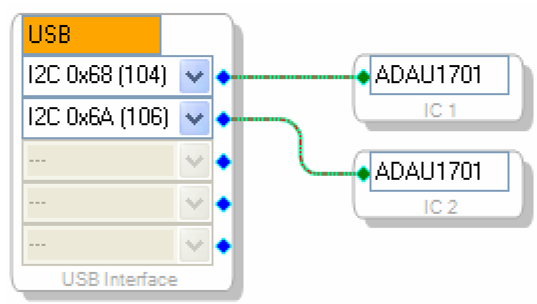


Figure 11. Multiple Address Operation with I²C

Multiple Address Operation with SPI

The USBi can support up to 2 identical devices on the same SPI latch, as long as the ADDR0 pins of the target devices are independently set to 2 different addresses, matching the addresses in the drop-down menu in the Hardware Configuration tab of SigmaStudio.

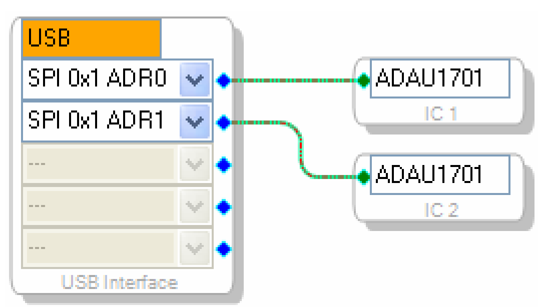


Figure 12. Multiple Address Operation with SPI

Multiple Latch Operation with SPI

The USBi can support devices on 5 different SPI latches. In the case where multiple latches are used, the additional SPI latch signals from the USBi, which are not connected to the ribbon cable, need to be manually wired to the target.

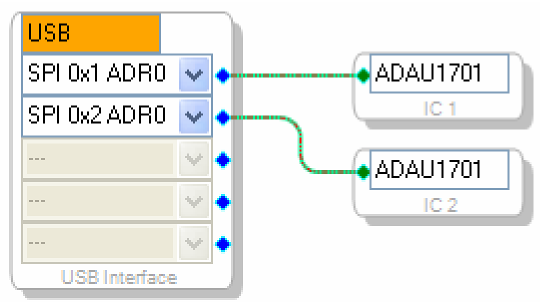


Figure 13. Multiple Latch Operation with SPI

The locations of extended SPI latch signals are shown in Figure 14.

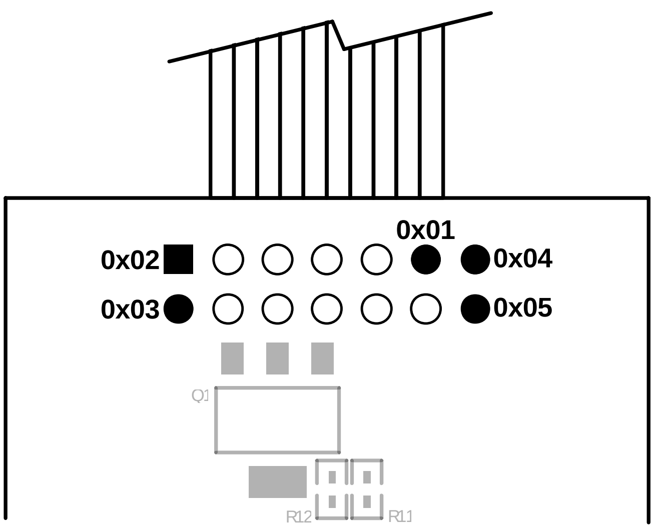


Figure 14. Extended SPI Latch Signal Pin Out (Bottom View of Board)

Combined Multiple Latch and Multiple Address Operation with SPI

A combination of multiple latch and multiple address schemes can be used, but the total number of devices cannot exceed 5.

CONTROLLING THE USBI

The USBi has several functions for controlling the target hardware. The control options are accessed in SigmaStudio by right-clicking on the USBi cell in the Hardware Configuration window.

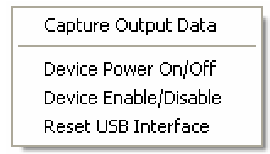


Figure 15. USBi Control Menu

Capture Output Data

Access the Capture Window, which displays a log of all communication between the PC and the target IC.

Device Power On/Off

This function toggles the line that supplies power to the target board. By default, the device power is On.

Device Enable/Disable

For supported ICs, this function will toggle low power mode.

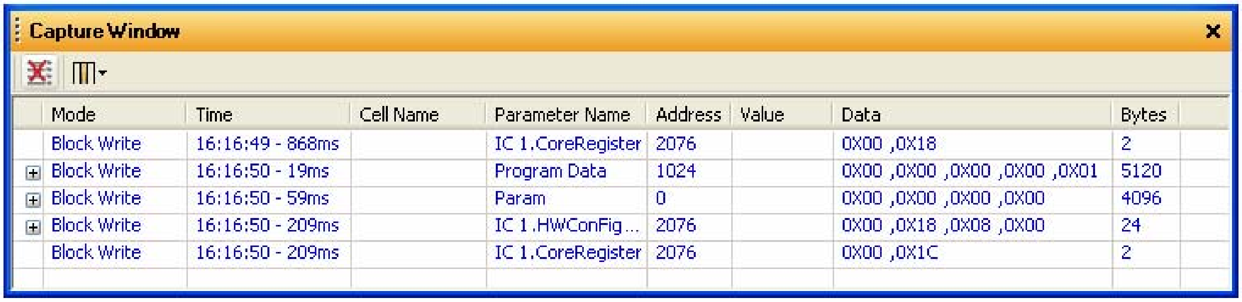
Reset USB Interface

This function performs a software reset of the USB driver, and will cause the Cypress USB Microcontroller to re-load its firmware.

MONITORING THE USBI

Using the capture window, it is possible to view all outgoing communications transfers from the PC to the target IC. For each write, the write mode, time of write, cell name (if applicable), parameter name, address, data (in decimal and hex), and length are shown.

For block writes where more than one memory location is written, only the first location is shown. The expand/collapse button in the leftmost column allows the user to view the full data write.



Mode	Time	Cell Name	Parameter Name	Address	Value	Data	Bytes
Block Write	16:16:49 - 868ms		IC 1.CoreRegister	2076		0X00 ,0X18	2
Block Write	16:16:50 - 19ms		Program Data	1024		0X00 ,0X00 ,0X00 ,0X00 ,0X01	5120
Block Write	16:16:50 - 59ms		Param	0		0X00 ,0X00 ,0X00 ,0X00	4096
Block Write	16:16:50 - 209ms		IC 1.HWConFig...	2076		0X00 ,0X18 ,0X08 ,0X00	24
Block Write	16:16:50 - 209ms		IC 1.CoreRegister	2076		0X00 ,0X1C	2

Figure 16. Output Capture Window

USING THE USBI TO PROGRAM A SELF-BOOT EEPROM

After compiling a project, the registers and RAM contents can be written to a target EEPROM for self-boot. In order to use this functionality, an E2Prom IC must be connected to the USBi in the Hardware Configuration window. After verifying that the EEPROM's Write Protect pin is disabled on the target board, right click the target IC (SigmaDSP), and click "Write Latest Compilation to E2PROM."

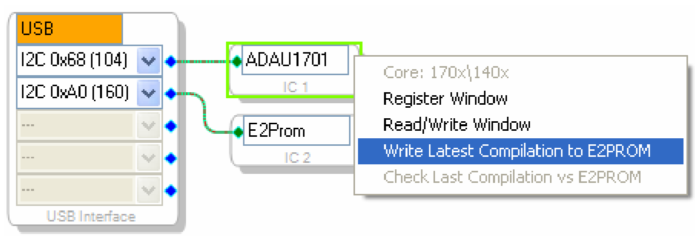


Figure 17. Writing to the Self-Boot EEPROM

CIRCUIT SCHEMATICS

USB CONNECTOR

The connection between the host PC and the Cypress USB Interface device is a standard USB cable, which carries D+ and D- signals for data communications, a 5V power supply, and ground. The D+ and D- lines are one-wire communications interface carried by half-duplex differential signals on a twisted pair. The clock is embedded in the data using the Non-Return-to-Zero Inverted (NRZI) line code. These signal lines connect directly to pins on the Cypress USB Interface.

A surface-mounted USB Mini B jack was selected for its low profile and increasing ubiquity in consumer electronics.

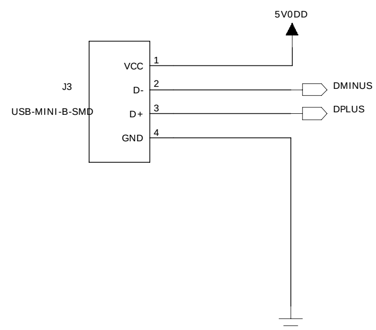


Figure 18. USB Connector Schematic

POWER REGULATOR

The Cypress USB Interface I/O ports are capable of operating in both 1.8V and 3.3V modes, depending on the target device in the system. Two regulators, one for 5 V to 3.3 V regulation and the other for 5 V to 1.8 V regulation, run simultaneously when the board is powered. A switch (S4) is provided to easily switch the IOVDD supply between the two regulators. LED D4 provides visual feedback that the board is being supplied with +5 V power from the PC's USB port.

The position of switch S4 should not be changed when the board is connected to the USB bus.

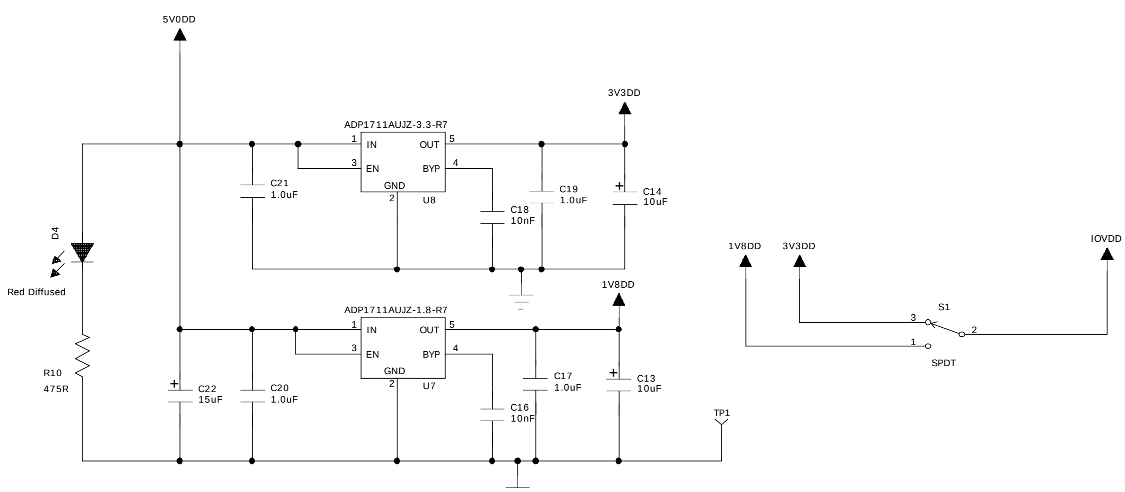


Figure 19. Power Regulator Schematic

CYPRESS USB INTERFACE

The Cypress USB Interface is the core of the system, including all of the necessary functionality to convert USB commands into corresponding I2C or SPI read/write transfers, and acts as a FIFO to route data between the host PC and the target device.

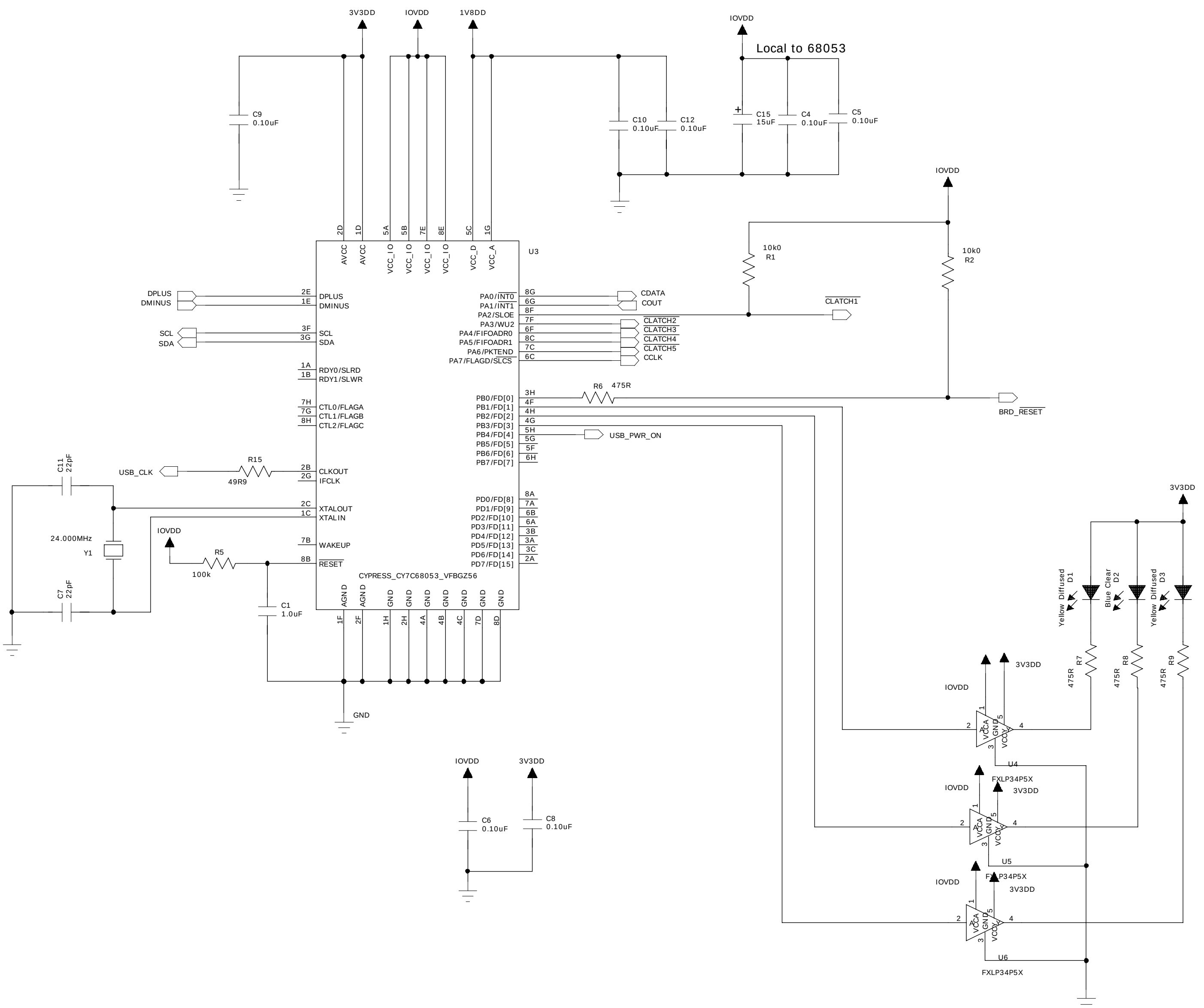


Figure 20. Cypress USB Interface Schematic

CRYSTAL OSCILLATOR SCHEMATIC

The Cypress USB Interface is its own clock master, and the board includes a crystal oscillator circuit with a 24 MHz piezoelectric crystal resonator to provide stability to the oscillator circuit. The crystal resonator is driven in parallel by the XTALOUT and XTALIN pins of the Cypress USB Interface.

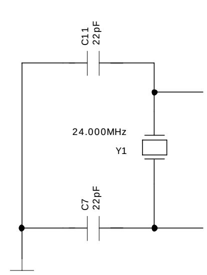


Figure 21. Crystal Oscillator Schematic

LEDs

The LEDs provide feedback to the user about the status of the Cypress USB Microcontroller.

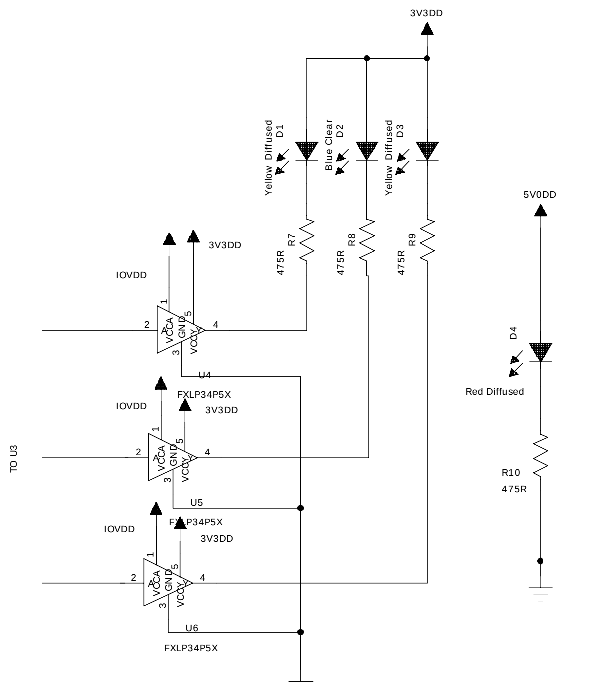


Figure 22. LEDs Schematic

Reference Designator	Color	Functionality
D1	Yellow	I^2C mode is active
D2	Blue	GPIO LED, for firmware debug purposes
D3	Yellow	SPI mode is active
D4	Red	+5 V power being is supplied over the USB bus

EEPROM

The EEPROM is an important system element that identifies the board to the host PC and stores the firmware for the Cypress USB Interface. The EEPROM is programmed during manufacturing via the J2 connector.

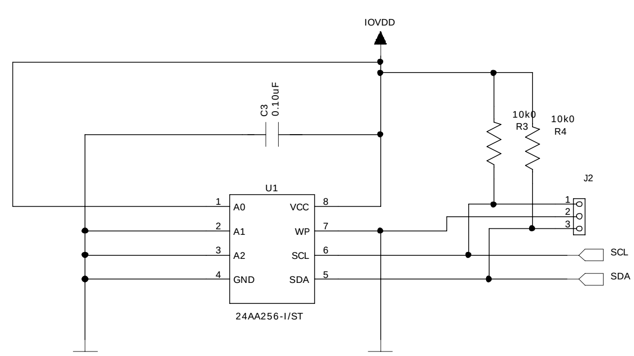


Figure 23. EEPROM Schematic

TARGET BOARD POWER SWITCH

The USBi is capable of supplying power to the target board after the Cypress USB Microcontroller has finished its boot up process. The USB_PWR_ON signal connects to the base of Q2 and turns on both transistors when driven high.

This circuit also enables a software-controlled target reset from SigmaStudio.

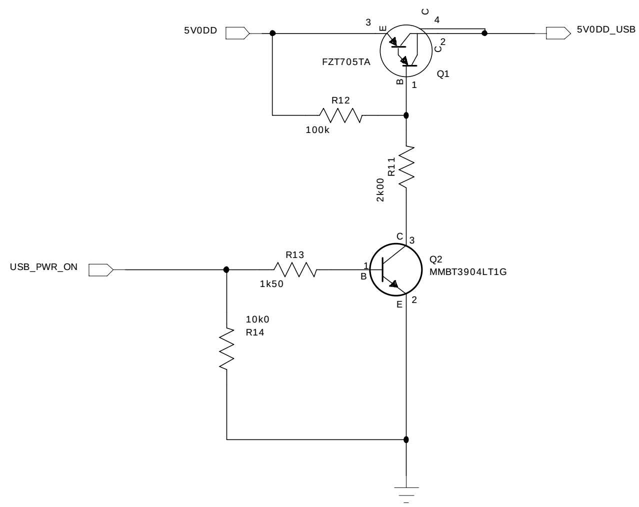


Figure 24. Target Power Switch Schematic

TARGET BOARD PROGRAMMING HEADER

In order to properly boot the Cypress USB Microcontroller from the EEPROM, it is necessary to remove all other devices from the I²C bus. The ADG721BRMZ analog switch will remain open, isolating the I²C bus from the target, until the boot process has completed.

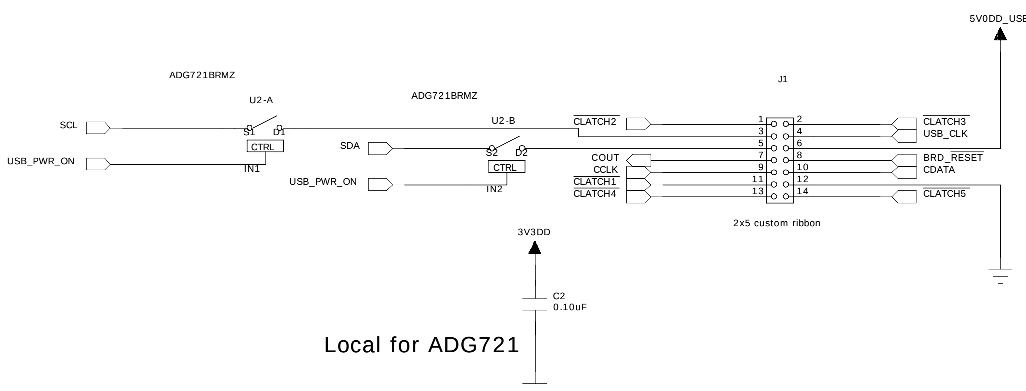


Figure 25. Target Board Programming Header Schematic

BOARD LAYOUT

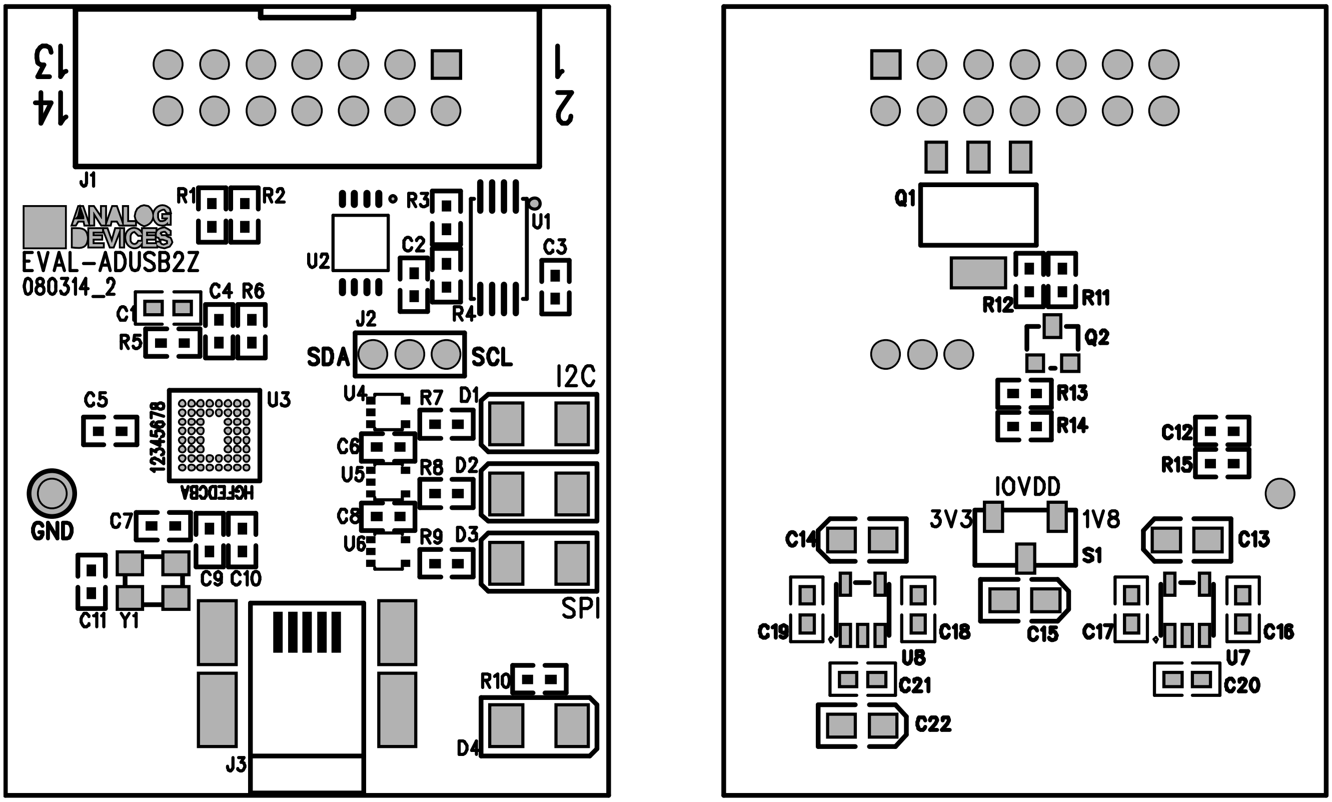


Figure 26. Board Layout

BOARD SCHEMATICS

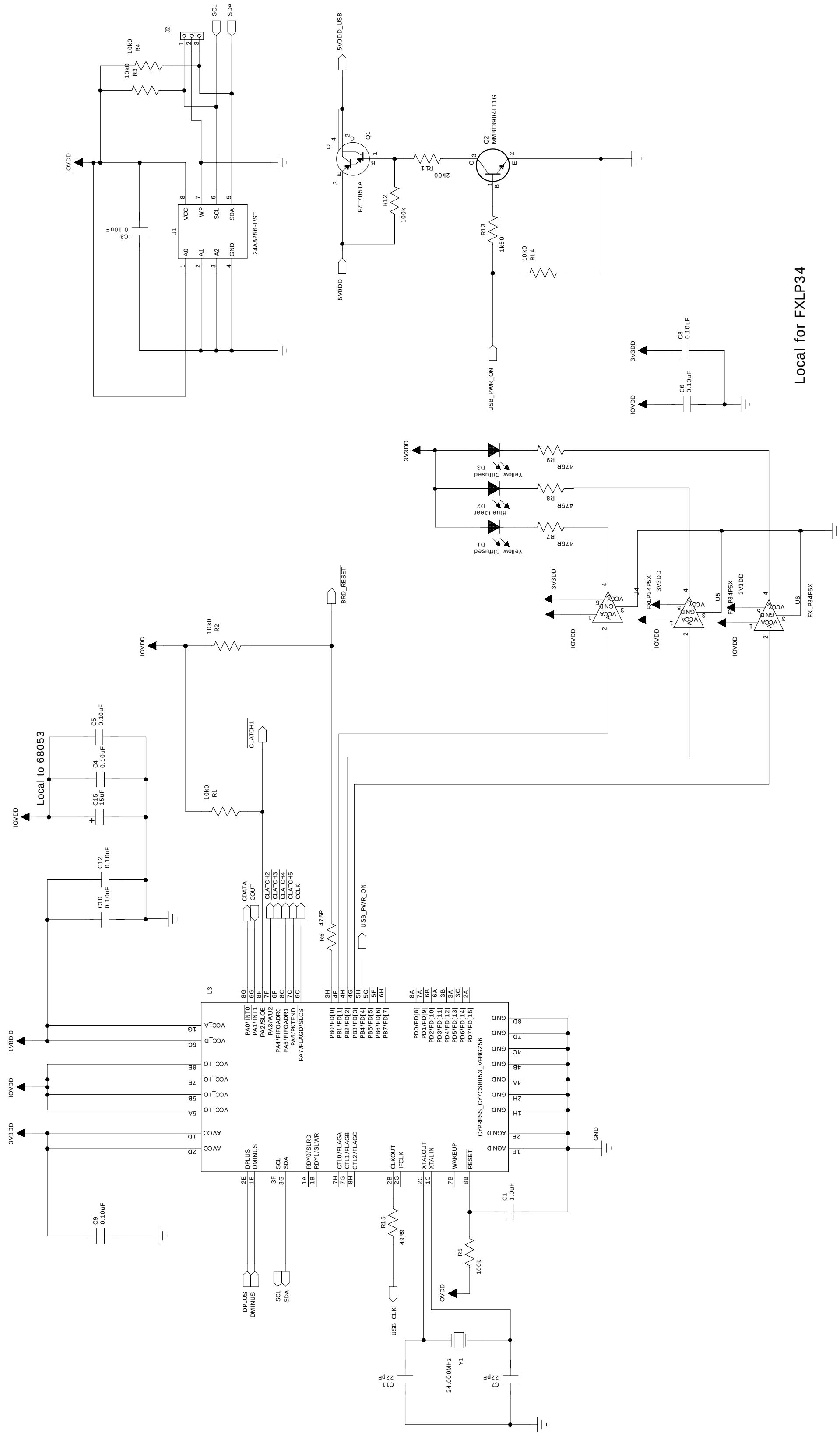


Figure 27. Board Schematics Page 1

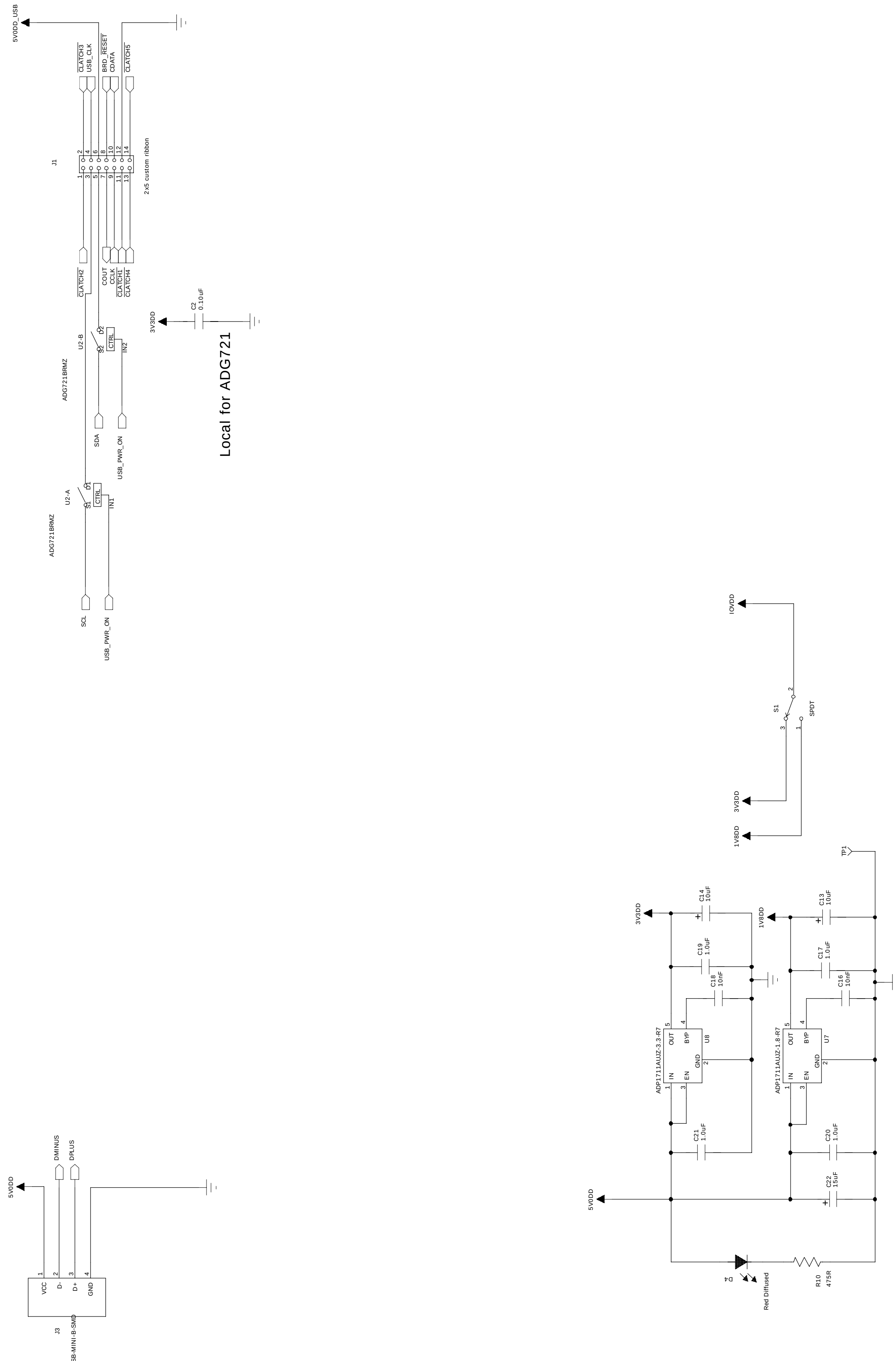


Figure 28. Board Schematics Page 2

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
EVAL-ADUSB2EBZ		Evaluation Board	

Purchase of licensed I^2C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.