**JTAG Connector**

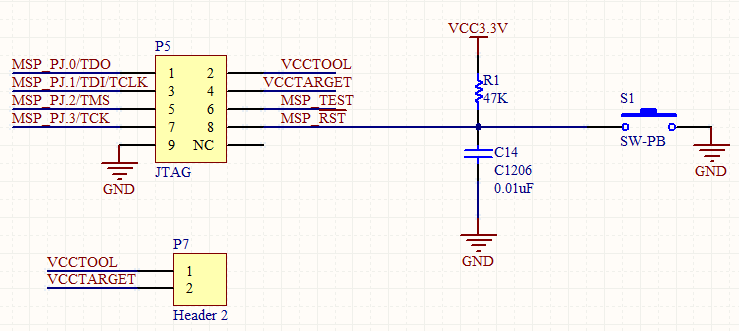


Figure 1: JTAG Programmer Connector Schematic

In order to program the MSP430, we needed a way to load code. The MSP430F5438A requires a 4-wire JTAG programming protocol. Using the Texas Instruments SLAU278H JTAG configuration, a schematic of this interface was developed as shown in Figure 1 [1].s The MSP-FET430UIF was used as the JTAG programmer for the MSP430F5438A through Code Composer Studio version 5. Note that the MSP-FET430UIF cannot be used to program the MSP430F5438A in Code Composer Studio version 4 due to an incompatible firmware update for the programmer. This information is currently not well documented in the literature provided by Texas Instruments; however, information on this issue may be found on the TI E2E Community Support Forums. As shown in Figure 1, the four pins, TDO, TDI, TMS, and TCK are used to program the MSP430F5438A. Additionally, the two pins, VCC TOOL and VCC TARGET are used to either power the programmer via USB power supplied by the host computer or powering from the target MSP430F5438A board. Depending on the user’s requirements, the user may choose to power the programmer off of the board or via USB through a jumper connection on the P7 header.

The programmer also has a connection to the /RST pin of the processor for resetting the processor after programming. The reset switch provides a quick way to re-run code on the MSP430F5438A in the event that the processor gets hung-up, or to simply reset the processor. In order to reset the processor, /RST is brought to ground. When the processor is not in reset, and in order to maintain normal operation, /RST is pulled up to VCC. The reset circuit consisting of R1, C14, and the momentary push button is called a button debounce circuit. This circuit prevents spikes of high and low voltages corresponding to high and low input to the reset pin of the processor. This debounce circuit relies principally on the capacitor which serves to resist sudden voltage changes on the line.

**RS232 Interface**

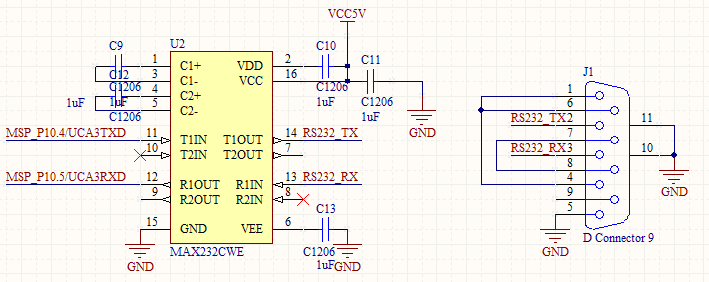


Figure 2: RS232 Interface Schematic

As a back-up in case there was an issue with the USB interface with the MSP430F5438A we employed an RS232 interface as shown in Figure 2. This would provide an additional interface between the MSP430F5438A and the host computer for XBee RX and TX data. As you can see in the schematic, the MAX232 is powered externally by 5V but it requires 10V to operate which is supplied internally through dual charge-pump DC-DC voltage converters. The first charge pump on the MAX232 takes in 5V doubling the voltage to approximately 10V. While the second charge pump takes the +10V, inverts the input, outputting -10V. The charge pump, capacitor values, and pin configuration for typical operation can be found in the datasheet in the appendix for the MAX232. Furthermore, for interfacing a host computer over serial, a DB9 serial connector is necessary as shown in Figure 2.

**USB Circuit**

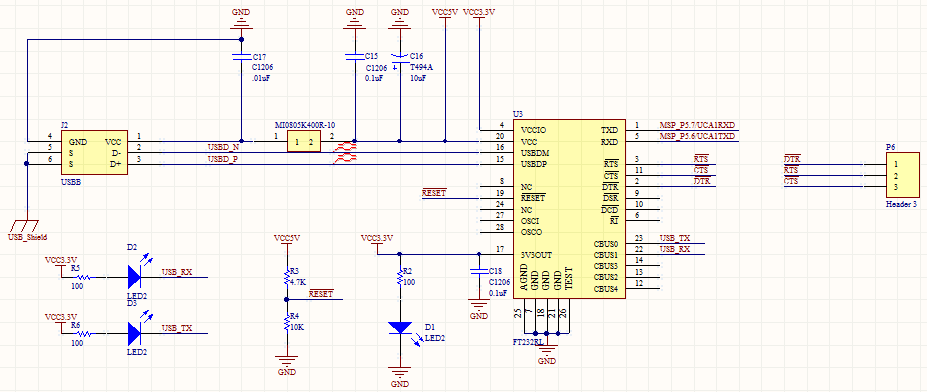


Figure 3: USB Interface Schematic

Figure 3 displays the USB Interface Schematic for the custom MSP430 board. The FT232RL chip from FTDI is a USB to serial interface allowing the MSP430F5438A to communicate XBee RX and TX data over a USB connection with the host computer. The design of this circuit was facilitated using an application note by FTDI on hardware design guidelines for FTDI ICs [2]. As shown in Figure 3, the FT232RL is in a self-powered or bus powered configuration, drawing power from the USB-B connector connected to the host computer. Additionally, the FT232RL has a +3.3V output from an integrated LDO regulator which the datasheet advises should be decoupled to ground using a 100nF capacitor. The pin VCCIO is the power supply for the UART interface and CBUS group pins. The MSP430F5438A datasheet recommends a maximum of +3.6V at any pin, the VCCIO pin on the FT232RL must be set to +3.3V in order to ensure recommended operating conditions have been met. Since the FT232RL outputs +3.3V at pin 17, VCCIO could be powered by the FT232RL. Due to an oversight, VCCIO was tied to +5V so the PCB trace had to be removed and a white wire was added to connect pin 17 to VCCIO. Note that Figure 3 shows the proper configuration.

You can see in Figure 3 that a ferrite bead is connected in series with the USB power supply to reduce EMI noise from being radiated down the USB cable to the USB host from the FT232RL and associated circuitry. Obviously, the value of the ferrite bead depends on the total current drawn. A 1.5A 40Ω ferrite bead was employed by Laird Technologies. Furthermore, decoupling capacitors C15 and C16 were also added to minimize EMI emission and improve ESD immunity. The signals D- and D+ were routed as a differential pair in order to reject common-mode noise. This means that D- and D+ had very equal lengths in order to maintain the timing of signals. In addition, the overall length of these two signals was made to be as short as possible.

As a visual indication that the FT232RL was correctly powered, a power green LED was connected to the 3V3OUT pin. Additionally, since any of the CBUS I/O pins can be configured to drive an LED, LEDs were placed on CBUS0 and CBUS1 as visual indicators when data is being transmitted or received.

In Figure 3, /RESET on the FT232RL is connected to voltage divider consisting of R3 connected to 5V, in series with R4 connected to ground. This circuit serves to hold the IC in reset while a USB cable is not connected to the peripheral.

**XBee Circuit**

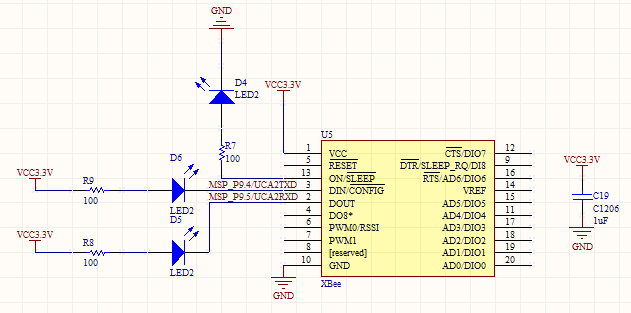


Figure 4: XBee Schematic

The XBee Series 1 chip antenna wireless transceiver was used to send as well as receive data from the XBee on the Beagle Bone to the host computer and vice versa. This XBee is controlled by the MSP430F5438A through a UART on the MSP.

The primary purpose of the LEDs on ON, DIN, and DOUT of the XBee is for debugging purposes. Additionally, a decoupling capacitor was placed across power and ground on the XBee. In an attempt to reduce reflections from the wireless transmitter, no components were placed immediately beneath the XBee in order to prevent turning a component into a disruptive antenna.

NEED KHASHI TO ADD INFO ABOUT XBEE CONFIGURATION!!!

**Microprocessor**

The design of the XBee transceiver board centers around the processor. This processor needed to be able to arbitrate data between an XBee and the host computer. The processor had to be reliable, powerful enough to arbitrate data, easy to develop with from both a software and hardware perspective as well as readily available. Since Capstone provides MSP430 development boards by Olimex and Texas Instruments offers free samples of most MSP430s we reasoned that we could begin development immediately on the MSP430 architecture. We chose the MSP430F5438A processor because we needed a processor with at least two UARTs, one for XBee communication and one for USB communication with the host computer. In addition, we wanted to add an RS232 serial interface as a backup in case the USB interface did not operate as desired. Therefore we needed a processor with at least three UARTs. Texas Instruments only has MSP430s with either two or four UARTs but not three so we decided to not sacrifice a UART and go with four. The MSP430F5438A has four UARTs, in addition to coming in a 100 pin LQFP package.

**MSP430F5438A**

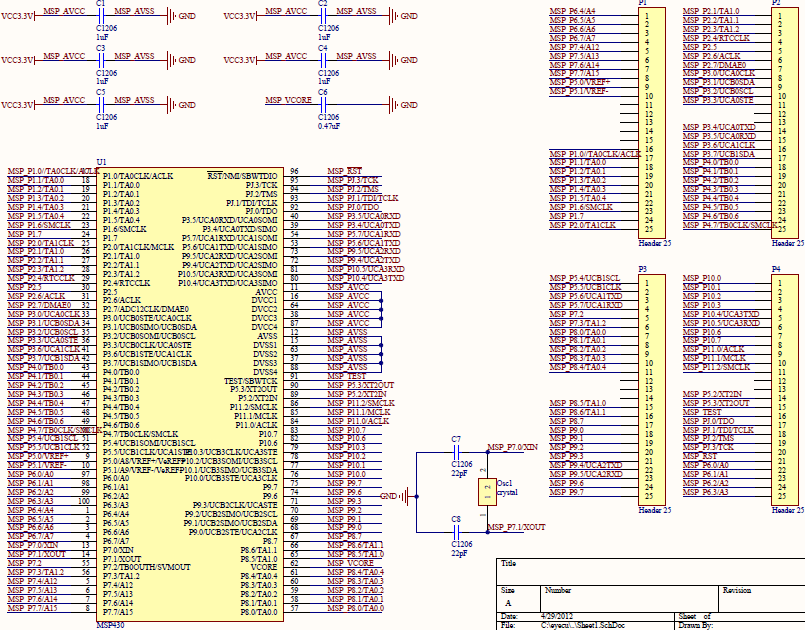


Figure 5: MSP430F5438A Processor Pin Breakout

The processor we are using on the MSP430 board is a MSP430F5438A. A schematic of the processor can be seen in Figure 5. As you can see in the schematic, the MSP430 runs off of 3.3V which is generated by the USB circuit. The MSP430 is supported by an external oscillator circuit with a 32.768kHz crystal. The MSP430F5438A has an internal 32.768kHz oscillator; however, it is recommended to use an external oscillator for time sensitive applications.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bibliography  |  |  | | --- | --- | | [1] | Future Technology Devices International Ltd., "Application Note AN\_146 USB Hardware Design Guidelines for FTDI ICs," Future Technology Devices International Limited, 2010. | | [2] | Texas Instruments Incorporated, "MSP430 Hardware Tools: User's Manual SLAU278H," 2009, Revised 2011. | |