Dual Serial Shield
User Manual

PN: 2050

Berkshire Products, Inc.
Phone: 770-271-0088
http://www.bkp-store.com/

Rev: 1.00

Table of Contents

Dual Serial Shield User Manual

The latest versions of manuals, sample code and other tools can be found on our site at:

http://bkp-store.com/

If you have any questions, corrections, or feedback about this manual please contact us at:

http://bkp-store.com/index.php?route=information/contact

1 Introduction

This shield has a SPI connected Dual UART (**NXP SC16IS762**) with 64 byte FIFOs on each channel. With FIFOs this large you are able to receive and send large packets of data. So large in fact that you may not need to use interrupts.

This board will mount on Ardiuno boards such the UNO or those with the same footprint. It will also mount on the new Freescale ARM based Freedom boards.

Note: It will not work with MEGA style Arduino boards since they do no have the SPI signals on IO pins 10-13. They only have them on the 6 pin (2x3) header on the middle of the board.

1.1 XB compatibility

The Dual Serial Shield can be used with XBee[1] radio modules to allow remote communications and control. Xbees are designed to use very little power except when the radios are transmitting data.

The board can be used with the following types of XBee & XBeePro radios in Series 1 & 2 footprints:

- DigiMesh[2] 2.4G
- DigiMesh[2] 900MHz
- 802.15.4
- 900MHz
- 868MHz
- Zigbee
- WiFi

For best results we recommend that you use XBee radios with the RP-SMA antenna connector or the mini whip antenna. These will experience the least interference from the copper areas on the PCB. The XBee radios with chip and PCB antennas will still work but their range may be reduced.

Note 1: XBee is a registered trademark of Digi International Inc.
Note 2: DigiMesh is a registered trademark of Digi International Inc.

2 Dual UART Features

- Dual full-duplex UART
- 64 bytes FIFO (transmitter and receiver)
- Fully compatible with industrial standard 16C450 and equivalent
- Baud rates up to 230.4K with 3.6864 MHz crystal
- Auto hardware flow control using RTS/CTS
- Auto software flow control with programmable Xon/Xoff characters
- Single or double Xon/Xoff characters
- Up to eight programmable I/O pins – GPIO or MODEM control
- Built-in IrDA encoder and decoder supporting IrDA SIR with speeds up to 115.2 kbit/s
- Software reset
- Transmitter and receiver can be enabled/disabled independent of each other
- Receive and Transmit FIFO levels
- Programmable special character detection
- Fully programmable character formatting
- 5-bit, 6-bit, 7-bit or 8-bit character
- Even, odd, or no parity
- 1, 1⁄2, or 2 stop bits
- Line break generation and detection
- Internal Loopback mode
- Sleep current less than 30 µA at 3.3 V
- 5 V tolerant inputs

3 Making Connections and Jumper Settings

Refer to following PCB diagram for component placement.

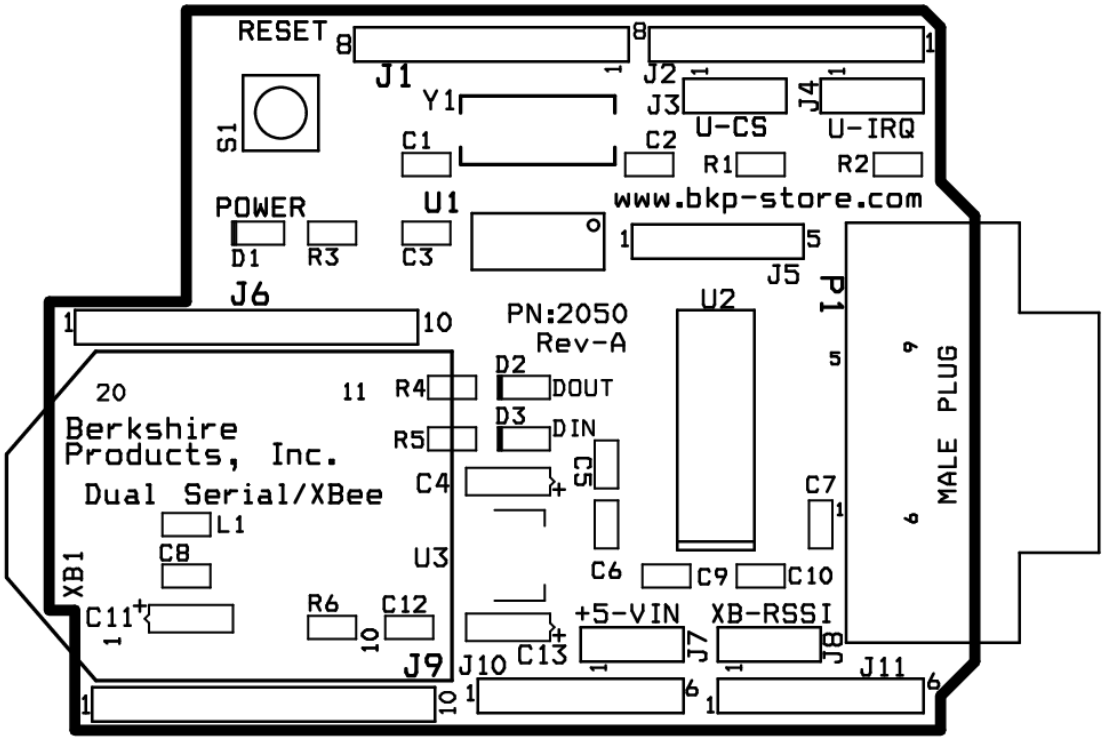


Figure 3-1

3.1 Power Input Jumper – J7

The board can accept DC power from the Arduino or Freedom main board. There is a three pin header at **J7** that selects the input from the Arduino +5V regulator or the external Vin from the 2.1mm power connector:

Pin 1-2: – Select Arduino +5V input
Pin 2-3: – Select external Vin

Note: Do not exceed +15V input to the board! You can measure the input voltage at **J10** Pin 6 and GND, Pin 4 or 5 on **J10**.

If you plan to use the UART Port A as a TTL UART only, then you should be able to use the Arduino +5V source (USB) to power the shield.

If you plan to use an XBee radio then you will need to provide an external power source and put **J7** jumper on Pins 2-3. The power supply must able to source enough current to power the board and the XBee radio in TX mode. This will typically be up to 500mA (0.5A) for the higher powered XBee radios.

3.2 UART SPI Bus Chip Select Option Jumper – J3

The Dual UART is connected to the SPI Bus from the Arduino or Freedom board. The UART is rated for 15MHz SPI Bus speeds, but the maximum speed will be dependent on the number of devices on the bus.

The three pin jumper on the shield, **J3** is used to select one of two Arduino IO pins for the chip select line:

Pin 1-2: – Select IO Pin 8 (default for sample code)
Pin 2-3: – Select IO Pin 4

3.3 UART Interrupt Option Jumper – J4

The Dual UART has an open-drain interrupt pin that is connected to a three pin jumper at **J4** to generate interrupts on one of two Arduino or Freedom pins. You can disable the interrupts totally by removing the shorting jumper. You can even use the IO pins as inputs only and poll the pins to see if the UART needs attention. The active level is logic LOW. The pins connections are:

Pin 1-2: – Select IO Pin 2
Pin 2-3: – Select IO Pin 7

Note: Currently there is no library support for interrupts.

3.4 UART Port-A Connections – J5, J6, J9

The Port A UART can be used with an XBee module or it can be used as a TTL level UART. To run it as an XBee, plug in an XBee Series 1 or 2 radio into the 2mm receptacles within the XBee silkscreen outline. Be careful and double check before you power up. It is possible to get the modules offset in the receptacles!

The following table shows the Pin connections for using the UART in a TTL level mode.

Pin ID	*Name*	*Description*	*I/O*	*Alternate Function*[1]
J9 – Pin 3	TxD	Transmit Data	Out	
J9 – Pin 2	RxD	Receive Data	In	
J6 – Pin 5	RTS	Request to Send	Out	
J6 – Pin 9	CTS	Clear to Send	In	
J5 – 1	DTR	Data Terminal Ready	Out	GPIO 5
J5 – 2	DSR	Data Set Ready	In	GPOI 4
J5 – 3	CD	Carrier Detect	In	GPIO 6
J5 - 4	RI	Ring Indicator	In	GPIO-7
J5 – 5	GND	Signal Ground	-	

Table 3-1

1: Alternate Functions allow these pin to be used as General Purpose IO. Please see the datasheet for modes of operation. The library will also need to be modified for GPIO rather than MODEM control signals.

Note: The TTL levels are 3.3V when high on the outputs. The inputs are 5.0V tolerant.

3.5 UART Port-B Connections – P1

The Port B UART is wired for RS-232 levels only as follows:

Pin ID	*Name*	*Description*	*I/O*
P1 – Pin 1	CD	Carrier Detect	In
P1 – Pin 2	RxD	Receive Data	In
P1 – Pin 3	TxD	Transmit Data	Out
P1 – Pin 4	DTR	Data Terminal Ready	Out
P1 – Pin 5	GND	Signal Ground	-
P1 – Pin 6	DSR	Data Set Ready	In
P1 – Pin 7	RTS	Request to Send	Out
P1 – Pin 8	CTS	Clear to Send	In
P1 – Pin 9	RI	Ring Indicator	In

Table 3-2

Note: This port is connected as DTE with a DB-9 Male connector. This is the same a COM port on a PC. If you want to connect the shield to a PC you will need a Null MODEM cable.

3.6 XBee RSSI Option Jumper – J8

Some XBee radios have an RSSI output pin that outputs a PWM signal to indicate relative receive signal strength based on the last reception. There is an RC filter on the board to smooth this signal to a DC level. The DC level is connected to a three pin jumper at **J8** to allow you to monitor the signal on one of two Arduino analog input pins as follows:

Pin 1-2: – Select Analog Pin 0
Pin 2-3: – Select Analog Pin 5

3.7 Reset Switch – S1

This switch is wired in parallel with the Arduino reset line. Pressing this switch will reset the shield and the Arduino or Freedom board.

4 Configuring the XBee Radio

The examples shown here use an XBee 2.4G DigiMesh version with Digi's X-CTU software which is free and available from Digi International (www.digi.com). To program the radios we use the UARTSBee V4 device which provides power and a USB UART connection to the PC. They are available from Amazon (www.amazon.com) and other sources and look similar to this picture:

Figure 4-1

These devices are also useful for connecting to your PC to use as the "base station" to talk back and forth with the remote I/O module's XBee radio.

Note: Be aware that this board may not be able to supply enough power for an XBee Pro type radio if you plug it into a USB hub.

Note: When the device is first plugged in you may get an alert from your PC that it needs to install drivers for a new device. The UARTSBee uses the FTDI chip for USB to Serial conversion and most PCs have these drivers pre-installed. If yours doesn't, go to www.ftdi-chip.com to get the latest ones for your PC and OS.

4.1 Starting X-CTU

When you start X-CTU you should see a screen like the following:

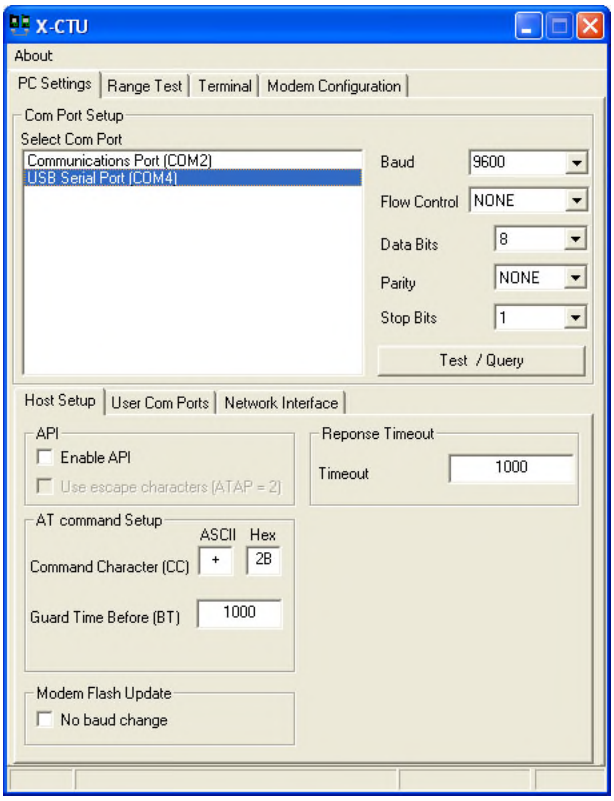


Figure 4-2

In this example the USB → Serial was on COM4. The baud rate should be 9600 to start with, since that is how the XBees are shipped from Digi. Click the Test / Query button to see if the XBee is active. If so, you should get a window like this:

Figure 4-3

Is this case the test was OK. It found an XB24 or a 2.4G DigiMesh. If it had been the higher power Pro it would have been XBP24. The firmware version is shown along with its serial number (IEEE Address) which is HEX: 0013A200 40870B85. The address is always 8 Bytes (64 bits) and leading zeros are not shown. Most Digi XBees start with 0013A200. It is also shown on the label on the bottom of the module.

4.2 Configuring the Networking

The next step is to click the Modem Configuration tab and then press the Read button to get a screen like the following:

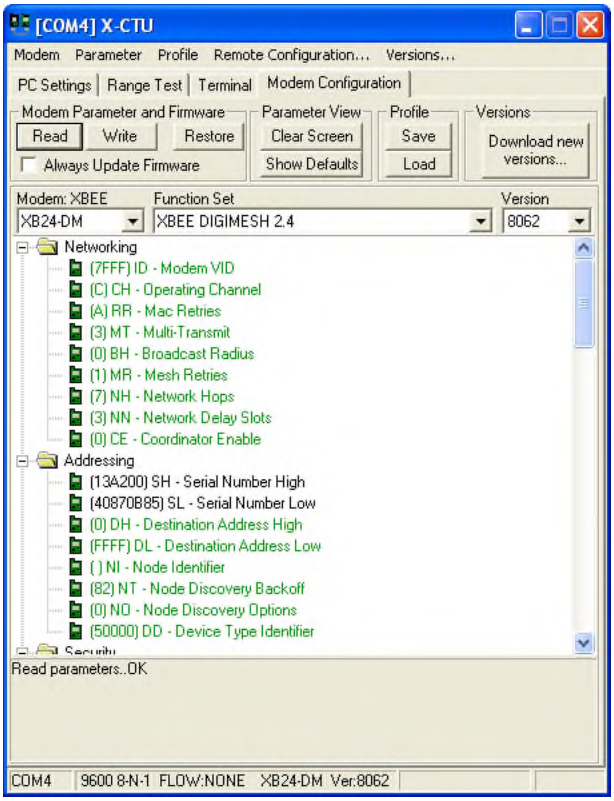


Figure 4-4

Under the Networking section there are options that you can change. Note that different types of modules will have different options. The ones shown here are for DigiMesh. For this module type the only two changes that we will talk about are the VID and the Channel.

- **Modem VID:** This is a HEX number that sets the PAN (Personal Area Network) ID for the network. Valid range is 0-0xFFFF. All the XBees on your network should use the same VID.
- **Operating Channel:** This sets the operating channel number (Uses 802.15.4 channel numbers). Note two things here:

1. XBee modules have more channels available than XBee Pro modules, so be sure to select an operating channel that both types can use if you have mixed modules on your network.
2. Select a channel number that minimizes conflict with any Wi-Fi networks you have. There are docs on the web that show 802.15.4 channel assignments.

See the Digi manual for specifics on other networking options.

4.3 Configuring the Addressing

XBee modules talk to each other using their serial numbers or a 16 bit addressing scheme. We will show the 802.15.4 64 bit scheme here. The Digi XBee manuals show you how to use short addresses.

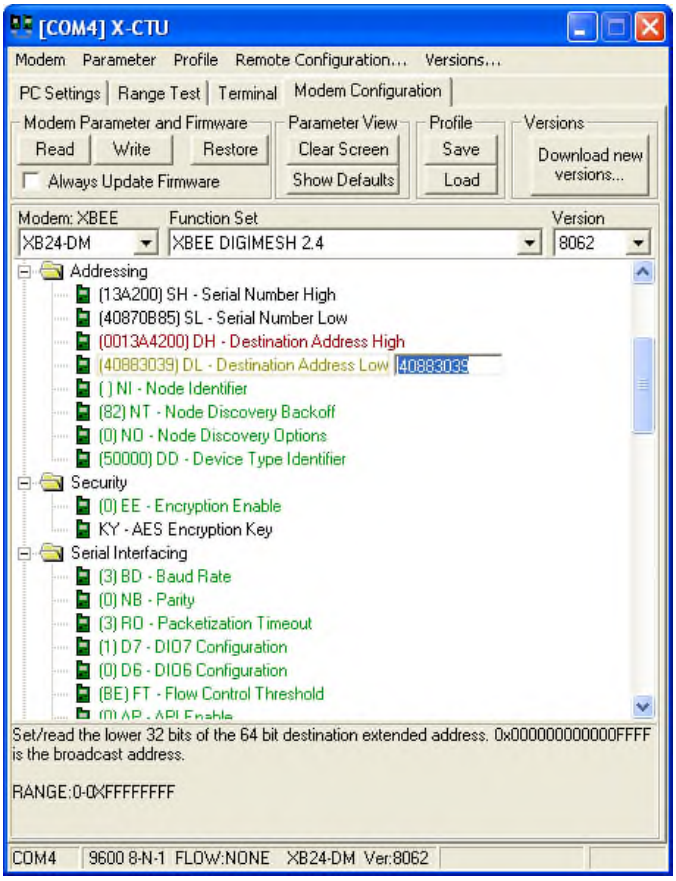


Figure 4-5

In this example we are programming a module for an I/O board that will talk to a host module (base station) at regular intervals. So in this case we set the High order 4 Bytes to the standard XBee address of 0x0013A200 and the lower 4 bytes to the address of our base station module 0x40883039. Once this is set, the module will know what address to send API packets to when an input changes or when a sample must be sent.

4.4 Configuring the Serial Interfacing

There are a few options to set here:

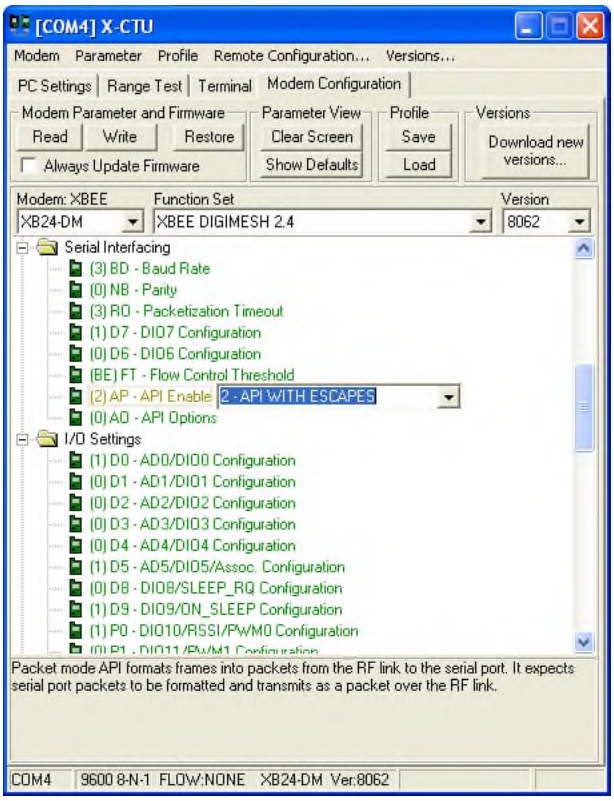


Figure 4-6

One option that needs setting here is the API enable. API mode must be enabled if the XBee module on the I/O board is going to be able to transmit input pin information and set the outputs. API mode 2 is used by the Arduino XBee library if you plan to use an Arduino type board as your base station.

Also note that the Digital I/O pins DIO6 and DIO7 are configured here. The reason for that is they can also be used as serial port flow control pins. If you are using DIO6 and DIO7 on your module then you should configure them here as Inputs or Outputs.

4.5 Configuring I/O Settings

Here is where you configure the I/O pins for DIO0 to DIO5 (see prior section for DIO6 & DIO7):

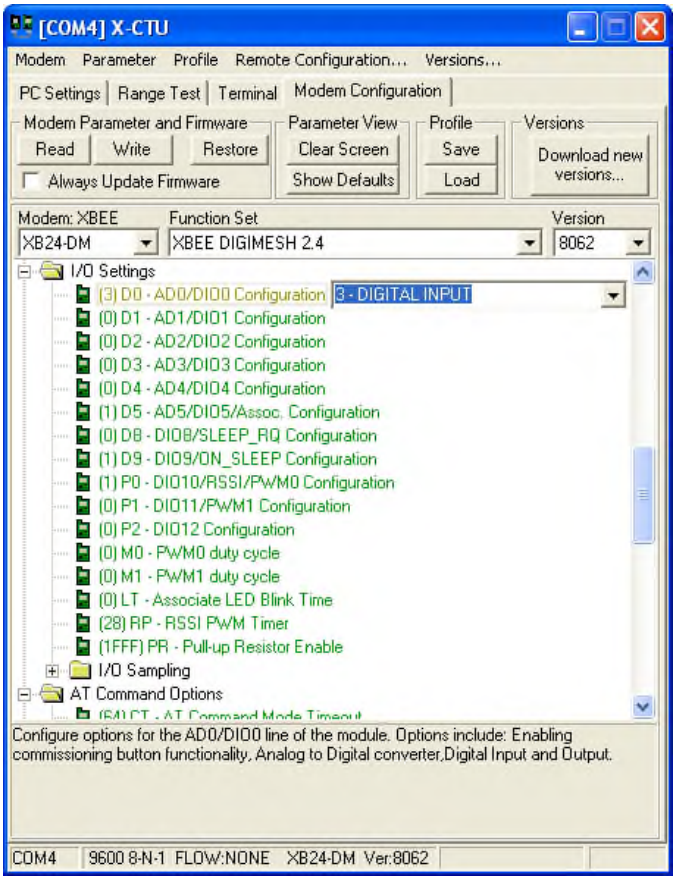


Figure 4-7

Configure DIO0 to DIO5 pins as input or output to match your I/O board setup. DIO8 to DIO12 are not used and should be left as **0-Disabled**.

The PR field defaults to 0x1FFF or binary 1 1111 1111 1111. Each bit set to one (1) defines a pull-up resistor on input pins DIO12 to DIO0. The default values should be fine unless you are using the input pins for analog inputs. In that case the lower six bits should be zero for any pin that is an analog input.

4.6 Configuring I/O Sampling

This is the last area of X-CTU covered here:

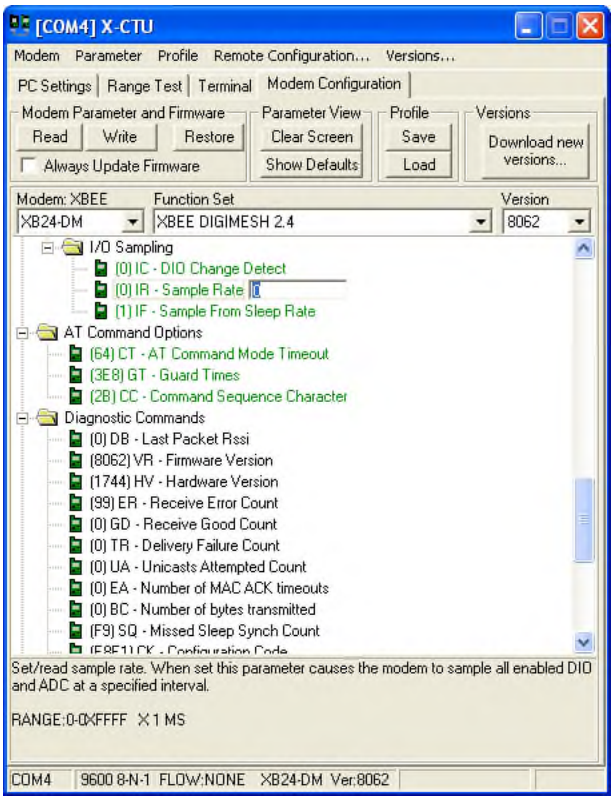
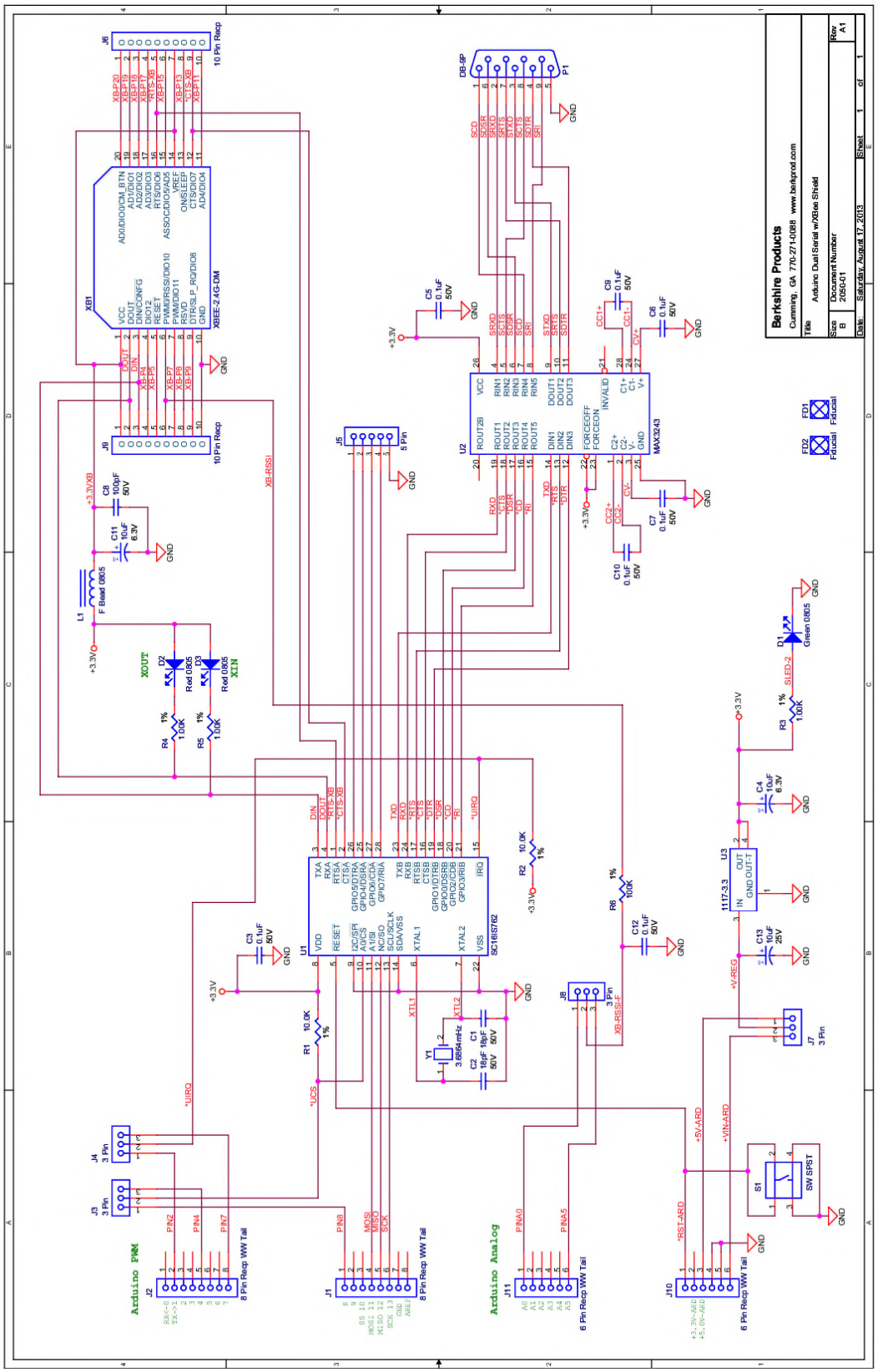


Figure 4-8

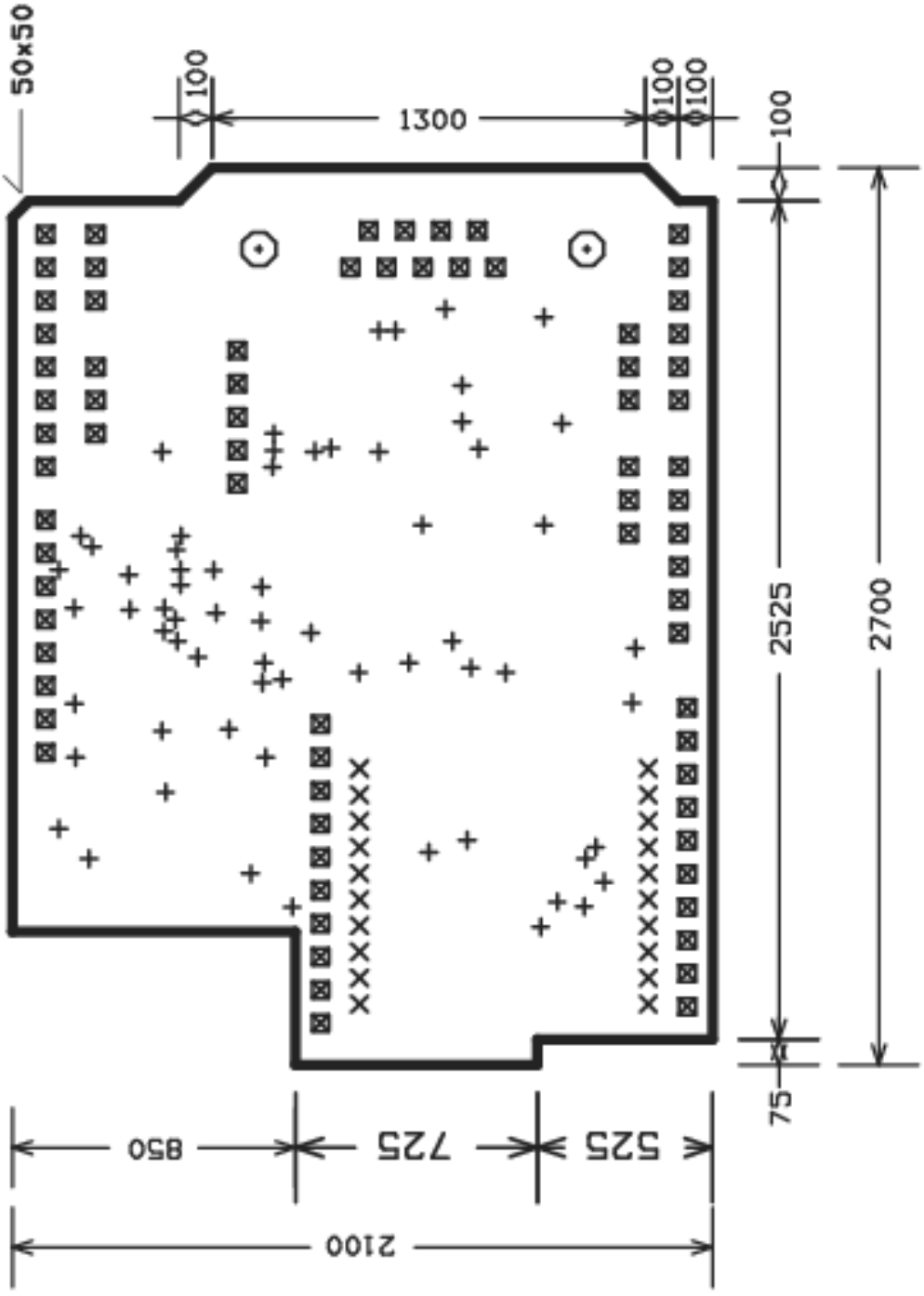
The **DIO Change Detect** is a bit field and should be set to a hex value. Any bit that is set will cause the XBee to send a packet when the I/O line for that bit changes (goes low to high or high to low) and the pin is set as an input. For example if you set this field to 0x2C (binary 0100 1100) then it would send a data packet if DIO6, DIO3 or DIO2 changes state on its input.

The **Sample Rate** tells the XBee module to send a data packet at regular intervals to report the status of digital and analog input pins. The number is in milliseconds so a reasonable minimum value is about 0x80 or 128mS. If an edge change is programmed and it occurs between samples then a packet will be sent then as well. The maximum value for this field is 0xFFFF or 65,353mS. A value of zero stops the samples.

5 Schematic Diagram



6 PCB Mechanical



Note: Dimensions are in mils.

7 Appendix A – Specifications

Power Requirements:

- +5V @ 60mA – minimum (without XBee module)
- +15V maximum @ 500mA

Do not exceed +15V input.

Environmental:

- -30° to +65°C - Operating
- -40° to +85°C - Storage
- 5% to 95% Relative Humidity, non-Condensing

UART Port A

- TTL Signal Levels
- Xbee Series 1 & 2 Compatible

UART Port B

- DB-9 wired as DTE
- RS-232 Signal Levels

8 Appendix B - Warranty

Berkshire Products, Inc. warrants to the original consumer or other end user purchaser that this product is free from defects in materials or workmanship for a period of one (1) year from the date of purchase. During the warranty period, and upon proof of purchase, the product will be repaired or replaced (with the same or functionally equivalent model) at our option, without charge for either parts or labor.

This warranty does not apply to defects due directly or indirectly to misuse, abuse, negligence, accident, repairs or alterations made by the customer or another party.

UNDER NO CIRCUMSTANCES WILL BERKSHIRE PRODUCTS, INC. BE LIABLE IN ANY WAY TO ANY PURCHASER FOR DAMAGES, LOST REVENUE, LOST WAGES, OR ANY OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THIS PRODUCT.

Berkshire Products, Inc. reserves the right to make modifications in this product without prior notification.