XBee/XBee-PRO Code Development

Contents

Technical Support

Live Chat: www.maxstream.net

E-mail: rf-xperts@maxstream.net

Phone: (801) 765-9885

M100320 [2006.05.15]

XBee / XBee-PRO Code Development

Three methods of software development are available to customers wanting to develop their own application to the XBee or XBee-PRO Module. Customer applications should be written to interface to one of the following software interfaces:

- Freescale-provided SMAC
- Freescale 802.15.4 PHY and MAC
- Figure 8 ZigBee stack

The software can be downloaded from Freescale's website [www.freescale.com]. This manual will discuss how to interface the XBee module to each piece of code and how the XBee and XBEE-PRO interface with MaxStream XBIB and XBIB-DEV interface boards.

Software Development Tools

Software development for the XBee or XBee-PRO Module should be done using the Metrowerks CodeWarrior Development Studio for HC(S)08 Microcontrollers. When developing using the "Freescale-provided SMAC" method, the free version (version CWX-H08-SE) may suffice. All other development most likely should be done using CodeWarrior version CWS-H08-C64K-CX. Other versions of CodeWarrior include simulation and additional development tools; however, the cited CodeWarrior versions support the minimum requirements. The Freescale Software is available for download at the following URL: www.freescale.com/codewarrior.

Two tools exist for downloading code to the XBee or XBee-PRO Modules. A "P&E Microsystems, Inc. USB HCS08/HCS12 Multilink" programmer/debugger [www.pemicro.com] can be used to program the modules. All MaxStream XBIB interface boards contain a six-pin header that will interface to this programmer.

Alternatively, the MaxStream X-CTU software [www.maxstream.net] can be used to program the modules through the serial port using the DIN/CONFIG, DOUT, RTS and DTR/PIN_SLEEP lines. The X-CTU Software interacts with the MaxStream Bootloader to program the remaining sectors of flash. Programming with the Multilink programmer/debugger will permanently erase the MaxStream Bootloader.

Using the MaxStream Bootloader

XBee and XBee-PRO Modules ship with the MaxStream Bootloader. The Bootloader sector contains a unique 64-bit serial number and calibration information for each module. The stored calibration values allow the XBee-PRO Modules to achieve 18 dBm power output. If the Multilink programmer is used to program the modules, stored calibration values are lost. In this case, the XBee-PRO can only achieve 12 dBm power output.

The unique serial number and calibration information can be preserved by programming application code using the X-CTU programming utility. To do this, the linker must reserve a section in flash from 0xFE00 – 0xFFFF for the Bootloader and redirect the Interrupt vector table. These changes have already been made in the "Ptc_XCTU_802_15_4.prm" and "Ptc_XCTU_ZigBee.prm" files.

The X-CTU requires application code to be built using the Intel hex format. The linker used in the Metrowerks Code Warrior IDE will generate hex files if the string "-EnvSRECORD=hex" is added to the Linker Command Line Arguments.

To use the X-CTU software to program application code into the modules, an MXI file must be created. The MXI file must have the same name as the corresponding hex file. A sample MXI file is provided in the "X-CTU" folder. The MXI and hex files should be copied into the "Program Files\MaxStream\X-CTU\Update\XBee" folder.

To access the unique 64-bit address contained in the Bootloader sector, an application should create a ProtectedConfig structure. An example of this structure is provided in the "ProtConfig.c" and "ProtConfig.h" files which are located in the source subdirectory of each project. With the structure defined as shown in these files, the unique 64-bit address can be read from ProtectedConfig.serialNumber[].

XBee / XBee-PRO Hardware Components

XBee. The following are the main components of the XBee Module:

- MC9S08GT60 microcontroller
- MC13193 RF chip

Additionally, an RF switch is included that switches the antenna between Transmit and Receive Modes. A simplified schematic of the XBee Module is located in Appendix A [p12].

XBee-PRO. The XBee-PRO Module also includes the MC9S08GT60 micro-controller and MC13193 RF chip. The XBee-PRO Module also contains a switch that switches the antenna between RF transmit and receive modes; however, after the switch is a Low Noise Amplifier (LNA) for Receive Mode and a Power Amplifier (PA) for Transmit Mode.

The XBee-PRO Module also contains:

- An analog switch that can switch the MC9S08GT60 A/D reference from the 2 volt regulated output of the MC13193 and a pin on the XBee-PRO module.
- A digital potentiometer that is used to adjust the power output of the PA.

A simplified schematic of the XBee-PRO Module is located in Appendix A [p14].

Datasheets for the MC9S08GT60 and MC13193 are available at www.freescale.com.

XBee / XBee-PRO Interface between MC9S08GT60 and MC13193

The following table displays pin connections between the MC9S08GT60 micro-controller and the Freescale MC13193 RF transceiver.

MC13193 pin		MC9S08GT60 pin	
Pin #	Name	Pin #	Name
10	GPIO2	23	PTD3/TPM2CH2
11	GPIO1	24	PTD4/TPM2CH0
12	*RESET	6	PTC4
13	RXTXEN	5	PTC3/SCL1
14	*ATTN	4	PTC2/SDA1
15	CLK0	47	PTG2/EXTAL
16	SPICLK	16	PTE5/SPSCK1
17	MOSI	15	PTE4/MOSI
18	MISO	14	PTE3/MISO1
19	*CE	13	PTE2/*SS
20	*IRQ	12	IRQ

Asterisk (*) indicates the pin is low-asserted.

Additional Connected Pins Used on MC9S08GT60

The following table displays other MC9S08GT60 micro-controller pins that are connected to various parts on the XBee and XBee-PRO Modules.

MC9S08GT60		Module	Functional Name - Description
Pin #	Name		
32	PTB7/AD1P7	XBee-PRO	VDET – voltage output from power amplifier
31	PTB6/AD1P6	XBee-PRO	VSup – voltage proportional to the module supply voltage VSup = 0.5*(supply voltage)
48	PTG3	XBee-PRO	Volt_Div – supply voltage for VSup on pin 31. 1 – when reading VSup, 0 – otherwise to conserve power.
30	VREF	XBee-PRO	VREF_OUT – output from switch that selects VREF – VREF can be connected to VDDA of MC13193 or VREF (pin 14) on module
39	PTA4/KBI1P4	XBee-PRO	VDDA/*VREF - select for VREF input. 1 – selects MC13193 VDDA, 0 – selects VREF (pin 14) on module
8	PTC6	Both	*TX/RX – Selects RF transmit or receive mode. 0 – transmit, 1 – receive. This pin should always be used with pin 37.
37	PTA2/KBI1P2	Both	TX/*RX – Selects RF transmit or receive mode. 1 – transmit, 0 – receive. This pin should always be used with pin 8.
36	PTA1/KBI1P1	Both	XBeeID – This pin is tied to ground for XBee-PRO. Floating for XBee
38	PTA3/KBI1P3	XBee-PRO	*DIG_POT_CS – This is the chip select for the PA digital potentiometer. The other lines tied to the digital potentiometer are MOSI and SPICLK. This line must be low while data is written to the digital potentiometer.
22	PTD2/TPM1CH2	XBee-PRO	DIG_POT_SRC – Source voltage for the PA digital potentiometer. This line should be high at all times except when entering a low power mode.

Asterisk (*) indicates the pin is low-asserted.

MC9S08GT60 micro-controller pins connected to XBee / XBee-PRO Module Pins

The following table displays other MC9S08GT60 micro-controller pins that are connected to various parts on the XBee and XBee-PRO Modules.

XBee/XBee-PRO Module		MC9S08GT60	
Pin #	Pin Name	Pin #	Pin Name
2	DOUT	10	PTE0/TXD1
3	DIN/*CONFIG	11	PTE1/RXD1
4	CD	7	PTC5
5	*RESET	1	*RESET
6	PWM0/RSSI	20	PTD0/TPM1CH0
7	PWM	21	PTD1/TPM1CH1
8	BKGD	45	PTG0/BKGD/MS
9	SLEEP_RQ	40	PTA5/KBI1P5
11	AD4/DIO4/RF_TX	29	PTB4/AD1P4
12	*CTS	35	PTA0/KBI1P0
13	ON/*SLEEP	9	PTC7
14	VREF	35	VREFH – On the XBee Output of analog switch on the XBee-PRO
15	ASSOC/AD5/DIO5	30	PTB5/AD1P5
16	*RTS	41	PTA6/KBI1P6
17	COORD/AD3/DIO3	28	PTB3/AD1P3
18	AD2/DIO2	27	PTB2/AD1P2
19	AD1/DIO1	26	PTB1/AD1P1
20	AD0/DIO0	25	PTB0/AD0P0

Asterisk (*) indicates the pin is low-asserted.

Freescale SMAC

The Freescale SMAC is provided as a ".lib" file. The Freescale SMAC must be modified slightly to support the XBee-PRO Modules. MaxStream has prepared a version of the SMAC 4.1 library that contains these modifications called SMAC_XBEE_DEV_4_1.lib.

Starting Development with the SMAC Application Template

The SMAC distributed by Freescale includes a sample project called "SMAC Application Template". The project is a simple code framework that can be quickly modified to work with the MaxStream PHY.

To modify the "SMAC Application Template" project to work with the XBee/XBee-PRO Modules, some files must be copied from the "SMAC" folder (supplied with this document) to locations within the Freescale SMAC source code. The following steps are required:

Add –DTARGET_XBEE to the compiler command line arguments to define "TARGET_XBEE" and delete any other TARGET defined in the command line.

Copy the "application_XBee.h" and "vectortable_XBee.c" files from the "SMAC\Source" folder to the SMAC "S08\apps\SMAC Application Template\Sources" folder.

Add the copied "application_XBee.h" and "vectortable_XBee.c" files to the project Sources Group.

Modify the following in the "application.h" file (Sources Group):
Add the following to line 2:

```
#include "application_XBee.h"
```

Disable or remove the "vectortable.c" file in the Sources Group.

Copy the "SMAC_XBEE_DEV_4_1a.lib" file from the "SMAC\Libs" folder to the SMAC "S08\smac\3.1\bin" folder.

Add the copied "SMAC_XBEE_DEV_4_1a.lib" file to the SMAC Group in the project.

Disable or remove the default SMAC Library (not SMAC_XBEE).

Disable or remove the "smac.mcp" file from the SMAC Group.

Ensure the "GT60_target.lib" library in the Target Group is selected for the project.

If using the X-CTU to program the radios, the following must also be done:

X-CTU:
Modify the following in the "vectortable_XBee.c" file (Sources Group):
Remove comment marker to enable USE_MAXSTREAM_BOOTLOADER define in line 34:

```
#define USE_MAXSTREAM_BOOTLOADER
```

Disable or remove the "P&E_ICD_linker.prm" file in the Prm Group.

Modify the following in the "Crt0.c" file. (This file is located in the "Select MCU Target.mcp" project, which can be opened from the Target Group.)
Change "#pragma CONST_SEG NV_REGISTERS" on line 57 to:

```
#if !defined BOOTLOADER_ENABLED || defined FOR_BOOTLOADER_ONLY
#pragma CONST_SEG DEFAULT            //←-- Change to "CONST_SEG DEFAULT"
// HCS08 NV register struct with values.
const volatile NV_REG_t none_volatile_struct =
```

(Note – The none_volatile_struct can be removed here and in "main.c" if desired.)

Copy the "Ptc_X-CTU_Smac.prm" linker file from the "SMAC\Linker" folder into the SMAC "S08\apps\SMAC Application Template\prm" folder.

Change the PRM file in the Linker settings to "Ptc_X-CTU_Smac.prm".

Freescale 802.15.4 PHY and MAC

The Freescale 802.15.4 PHY is provided from Freescale as source code with compiled library (.lib) files already made for different platforms, including the Freescale SARD and EVB development boards. To use the Freescale 802.15.4 PHY with the XBee/XBee-PRO Modules, it is necessary to make a PHY library for the XBee/XBee-PRO platform. This has already been done by MaxStream and is provided as the “802.15.4_PHY_XBEE_DEV_1061.lib” and “802.15.4_PHY_XBEE_1061.lib” files for the 802.15.4 and ZigBee applications, and as the “SMAC_XBEE_DEV_4_1a.lib” for the SMAC application. Make sure the version of Freescale 802.15.4 software matches the version of the PHY library or compiler; otherwise, linker errors may result.

The “802.15.4_PHY_XBEE_1061.lib” file must only be used with the MaxStream Bootloader. This PHY library uses the calibration information stored in the Bootloader to achieve up to +18dBm power output for the XBee-PRO Modules. This version of the PHY makes use of the A/D converter on the MC9S08GT60 to maintain the correct power output (XBee-PRO Modules only). If this PHY is used on an XBee-PRO module, the A/D converter should not be used in the application.

The “802.15.4_PHY_XBEE_DEV_1061.lib” file can be used with or without the MaxStream Bootloader. This PHY library does not use the calibration information in the Bootloader or the A/D converter. Instead, it provides a maximum +12dBm power output (nominal) for the XBee-PRO Modules. Applications on the XBee-PRO that require the A/D converter should use this PHY Library.

The “SMAC_XBEE_DEV_4_1a.lib” file should be used in all SMAC applications (with or without the MaxStream Bootloader). This library provides a maximum of +12dBm power output (nominal) for the XBee-PRO Modules. The A/D converter is not used in this library.

The Freescale 802.15.4 MAC is provided as compiled library (.lib) files with different features. There is no need to modify these files (especially since Freescale does not provide source code for the MAC).

Starting Development with “My_Wireless_App”

The 802.15.4 MAC distributed by Freescale includes a sample project called “My_Wireless_App”. The project contains simple pieces of working code that can be quickly modified to work with the MaxStream PHY.

To modify the “My_Wireless_App” project to work with the XBee/XBee-PRO Modules, some files must be copied from the “802.15.4” folder (supplied with this document) to locations within the Freescale source code. The following steps are required:

Add –DTARGET_XBEE to the compiler command line arguments to define “TARGET_XBEE” and delete any other TARGET defined in the command line.

Copy the “Target_XBee.h” and “App_Target_XBee.h” files from the “802.15.4\Source” folder to the Freescale “Demos\MyWirelessApp\src\Code\Sys” folder.

Add the “Target_XBee.h” and “App_Target_XBee.h” files to the Sys Group in the project.

Modify the following in the “Target.h” file (Sys Group):

Add the following to line 25:

```
#include “Target_XBee.h”
```

Comment out the default target defined around line 37:

```
//#define TARGET_DIG534_1 // Sniffer
//#define TARGET_AXIOM_GB60 // 802.15.4 Dev Platform
//#define TARGET_RD01                  //←-- Comment out this line
#endif
```

→

(Continued from previous page)

Modify the following in the "AppTarget.h" file (Sys Group):

Add the following to line 26:

```
#include "App_Target_XBee.h"
```

Comment out the default target defined around line 41:

```
//#define TARGET_DIG534_1 // Sniffer
//#define TARGET_AXIOM_GB60 // 802.15.4 Dev Platform
//#define TARGET_RD01                  //←-- Comment out this line
#endif
```

The remaining instructions are dependent upon which tool will be used for programming the modules.

P&E Multilink:
Copy the "802.15.4_PHY_XBEE_DEV_1061.Lib" file from the "802.15.4\Libs" folder into the Freescale "Freescale_Reference_Libs\Phy" folder.

Add the copied "802.15.4_PHY_XBEE_DEV_1061.Lib" file to the 802.15.4_Libs Group in the project.

Disable or remove the default PHY Library (not PHY_XBEE).

Use the default Linker file "Ptc.ach".

X-CTU:
Copy the "802.15.4_PHY_XBEE_1061.Lib" file from the "802.15.4\Libs" folder into the Freescale "Freescale_Reference_Libs\Phy" folder.

Add the copied "802.15.4_PHY_XBEE_1061.Lib" file to the 802.15.4_Libs Group in the project.

Disable or remove the default PHY Library (not PHY_XBEE).

Copy the "ProtConfig.c" and "ProtConfig.h" files from the "802.15.4\Source" folder into the Freescale "Demos\MyWirelessApp\src\Code\App" folder.

Add the "ProtConfig.c" and "ProtConfig.h" files to the MyApps Group in the project.

Modify the following in the "Crt0.c" file (Sys Group):

Change "#pragma CONST_SEG NV_REGISTERS" on line 58 to:

```
#if !defined BOOTLOADER_ENABLED || defined FOR_BOOTLOADER_ONLY
#pragma CONST_SEG DEFAULT          //←-- Change this line to CONST_SEG DEFAULT
// HCS08 NV register struct with values.
const volatile NV_REG_t none_volatile_struct =
```

(Note – The none_volatile_struct can be removed here and in main.c if desired.)

Copy the "Ptc_X-CTU_802_15_4.prm" linker file from the "802.15.4\Linker" folder into the Freescale "Demos\MyWirelessApp\src\Build" folder.

Make the following changes to the Linker settings:

Change the PRM file to "Ptc_X-CTU_802_15_4.prm".
Add "-WmsgSd1100" and "-EnvSRECORD=hex" to the Linker command line arguments.

Figure 8 ZigBee Stack

The Freescale ZigBee solution uses the Figure 8 ZigBee stack. A few modifications to the stack are necessary to support operation on the XBee / XBee-PRO Modules.

Starting Development with the Homelighting SRC03391 Project

The Figure 8 Stack includes a sample project called “Homelighting – SRC03391”. The project contains code for a simple light switch controller and can be quickly modified to work with the MaxStream PHY.

To modify the “Homelighting – SRC03391” project to work with the XBee/XBee-PRO Modules, some files must be copied from the “ZigBee” folder (supplied with this document), to locations within the Figure 8 stack source code. The following steps are required:

Make the following changes to the compiler command line arguments:
Add –DTARGET_XBEE to define “TARGET_XBEE”.
Delete any other TARGET defines (i.e. –DTARGET_SARD, etc).
Delete any GB60 or GT60 defines (i.e. –DGT60_SARD, etc).
Delete any LCD or Bootloader defines.

Copy the “Target_XBee.h” and “App_Target_XBee.h” files from the “ZigBee\Src” folder to the Figure 8 “Z-Stack\MAC_core\MC13192\Src\Ghdr” folder.

Add the “Target_XBee.h” and “App_Target_XBee.h” files to the MAC\Ghdr Group in the project.

Modify the following in the “Target.h” file (MAC\Ghdr Group):
Add the following to line 25:

```
#include “Target_XBee.h”
```

Comment out the default target (defined around line 36):

```
//#define TARGET_DIG534_1 // Sniffer
//#define TARGET_AXIOM_GB60                    //←-- Comment out this line
//#define TARGET_RD01 // For all boards which uses the Freescale reference design 01 layout
#endif
```

Modify the following in the “AppTarget.h” file (MAC\Ghdr):
Add the following to line 26:

```
#include “App_Target_XBee.h”
```

Comment out the default target (defined around line 41):

```
//#define TARGET_DIG534_1 // Sniffer
//#define TARGET_AXIOM_GB60                    //←-- Comment out this line
//#define TARGET_RD01 // For all boards which uses the Freescale reference design 01 layout
#endif
```

Copy the “OnBoard_XBee.c” and “OnBoard_XBee.h” files from the “ZigBee\Source” folder into the Figure 8 “Z-Stack\Zmain” folder.

Add the “OnBoard_XBee.c” and “OnBoard_XBee.h” files to the ZMain Group in the project.

Disable or remove the “OnBoard.c” file from the project (ZMain Group).

Modify the following in the “OnBoard.h” file (ZMain Group):
Add the following to line 27:

```
#include “OnBoard_XBee.h”
```

Comment out the #else and the 5 EVAL_SW defines (about lines 314 – 319)

```
//#else
//          #define EVAL_SW_MASK 0x00
//          #define EVAL_SW4 0x00…
```

→

(Continued from previous page)

The following instructions add serial port support (if necessary):

Serial Port:
Modify the following in the “MTSPCI.c” file (OSAL Group):
Add the following (around line 36):

```
#ifdef TARGET_XBEE
#include “gb60_io.h”
#endif
```

Modify the following in the “OnBoard.h” file (ZMain Group):
Add the following around line 44:

```
#ifdef TARGET_XBEE
#define ZTOOL_P1
#endif
```

The remaining instructions depend on which tool will be used for programming the modules.

P&E Multilink:
Copy the “802.15.4_PHY_XBEE_DEV_1061.Lib” file from the “ZigBee\Libs” folder into the Figure 8 “Z-Stack\MAC_core\MC13192\Libs” folder.

Add the “802.15.4_PHY_XBEE_DEV_1061.Lib” file to the MAC\Libs Group in the project.

Disable or remove the default PHY Library (Not PHY_XBEE).

Use the default Linker “Ptc.ach” file.

X-CTU:
Copy the “802.15.4_PHY_XBEE_1061.Lib” file from the “ZigBee\Libs” folder into the Figure- “Z-Stack\MAC_core\MC13192\Libs” folder.

Add the “802.15.4_PHY_XBEE_1061.Lib” file to the MAC\Libs Group in the project.

Disable or remove the default PHY Library (Not PHY_XBEE).

Copy the “ProtConfig.c” and “ProtConfig.h” files from the “ZigBee Source” folder into the Figure 8 directory where the Application files reside (i.e. ZStack_SRC03391 project Application files reside at “Z-Stack\Projects\Homelighting\SRC03391\ Source”).

Add the “ProtConfig.c” and “ProtConfig.h” files to the Application Group in the project.

Modify the following in the “Crt0.c” file (MAC\Sys Group):
Replace the #pragma on line 61 with the following:

```
#pragma CONST_SEG DEFAULT        //don’t place none_volatile_struct in NV_REG section
```

(Note – The none_volatile_struct can be removed here and in Main.c if desired.)

Copy the “Ptc_X-CTU_ZigBee.prm” linker file from the “ZigBee\Linker” folder into the Figure 8 “Z-Stack\MAC_core\MC13192\Src\Build” folder

Make the following changes to the Linker settings:
Change the PRM file to “Ptc_X-CTU_ZigBee.prm”.
Ensure “-WmsgSd1100” is included in the Linker command line arguments.

The next steps help verify the application was ported correctly:

Verification:
When the module powers up, LED1 (on XBIB-DEV interface board) should be lit. The module is waiting for user intervention to generate a random serial number (zmainEstablishExtAddr() in ZMain.c).

Press SW1 on the XBIB-DEV board. LED1 should turn off, and LED2 should turn on.

Press SW1 again on the XBIB-DEV board. LED2 should turn off, and LED3 should turn on, meaning the Coordinator has started successfully.

MaxStream PHY Low Power Operation

The “XBee-Phy.h” file is supplied in the 802.15.4, SMAC and ZigBee “Source” folders (supplied with this document). This file contains a prototype for the XBee_Phy_IO() function. This function will set the PHY-level I/O lines (and A/D Converter, XBee-PRO only) for either low-power or active operating modes based on the argument (PHY_SLEEP or PHY_WAKE respectively). This file can be added to the 802.15.4, SMAC or ZigBee projects if low power operation is required.

Application Low Power Notes

Several steps are listed below to help reduce power consumption when developing an application. These steps are supplemental to the process described in the Freescale documentation.

To Enter Sleep:

Configure the module ID pin (PTA1) as a low driving output:

```
PTAPE_PTAPE1 = 0;
PTAD_PTAD1 = 0;
PTADD_PTADD1 = 1;
```

Call XBee_Phy_IO() to configure the PHY IO lines for sleep:

```
XBee_Phy_IO(PHY_SLEEP);
```

(XBee_Phy_IO() is defined in “XBee-Phy.h”. See “MaxStream PHY Low Power Operation” section.)

Set the MAC Attribute gMPibRxOnWhenIdle_c to false to disable the receiver when idle.

Call AttEnable. (This macro is defined in Target_XBee.h.)

If using the Freescale ASP layer or Power Library, send the sleep request primitive (Hibernate Request, Doze Request, etc) to sleep the MC13193.

If applicable, issue the Stop command to sleep the MC9S08GT60:

```
asm STOP;
```

(see MC9S08GT60 datasheet for information on Stop modes.)

To Wake from Sleep:

AttDisable / AttEnable may need to be called to wake the MC13193 from sleep if it is in Hibernate or Acoma mode.

Set the MAC Attribute gMPibRxOnWhenIdle_c to true to enable the receiver when idle. This may require setting the channel again.

Setup the module ID pin (PTA1) as an input with the pull-up enabled:

```
PTAPE_PTAPE1 = 1;
PTADD_PTADD1 = 0;
```

Call XBee_Phy_IO() to configure the PHY IO lines for wake:

```
XBee_Phy_IO(PHY_WAKE);
```

RS-232 and USB Interface Boards

XBee Development Kits ship with XBIB-DEV development boards. The boards facilitate interfacing between an RF module and host. The XBIB-DEV contains a power supply and interfaces through RS-232 (XBIB-R-DEV) or USB (XBIB-U-DEV) connections. Certain pins on the RF module connect to the interface board to provide the RS-232 and USB functionality. USB functionality is made through a virtual COM (RS-232) port; which means the same XBEE pins that are connected through the RS-232 level translator are connected to a USB interface that provides the same functionality as the RS-232 interface. A connector is provided for the Freescale BDM interface. In addition, four push button switches and four LEDs are provided. This is done to mimic the switches and LEDs on the Freescale development boards. A RESET switch is also provided. The following table describes the connections from the MC9S08GT60 to the XBIB-DEV interface board.

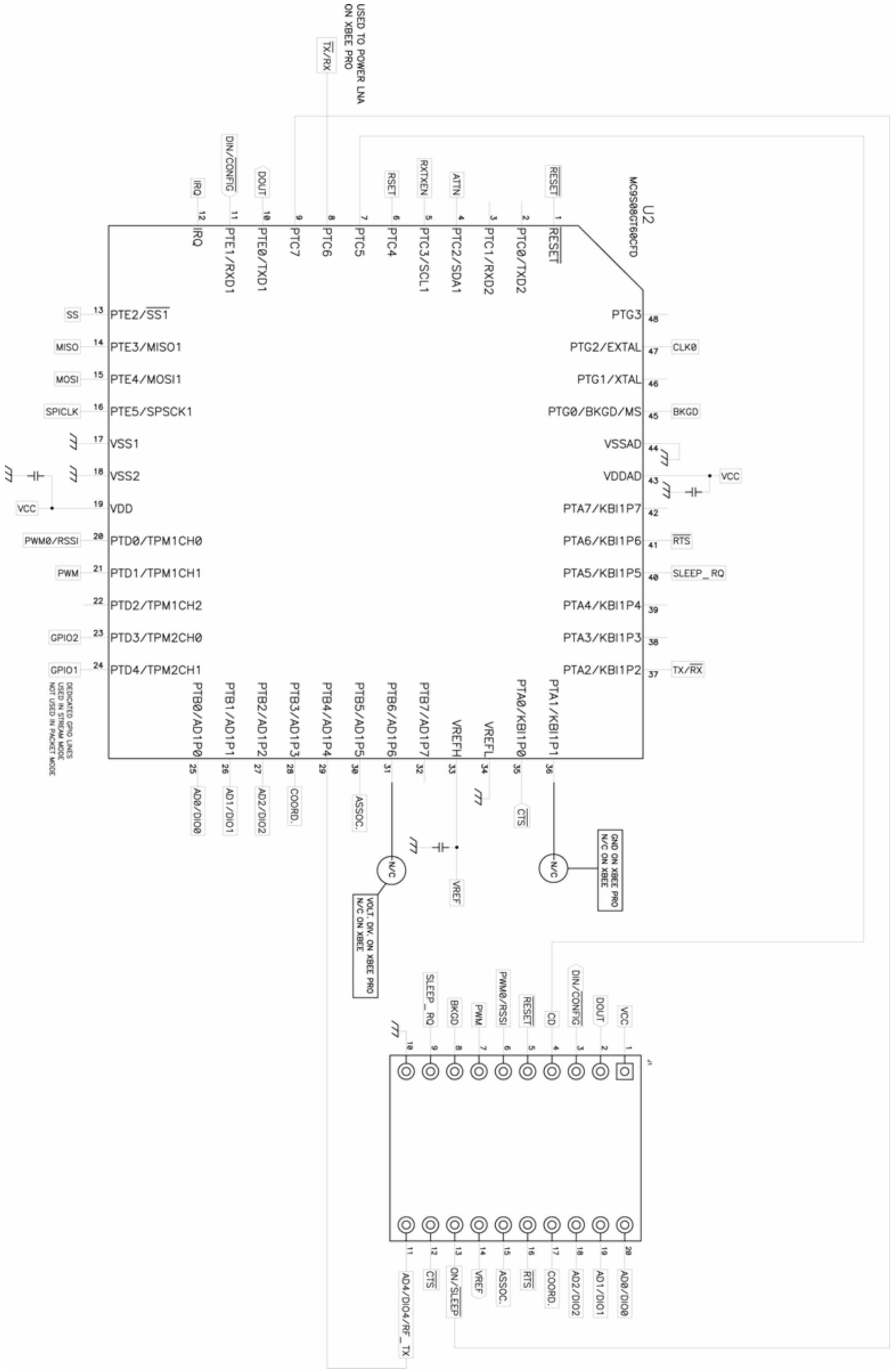
Pin # - XBee/XBee-PRO Module pin name	**XBIB-R-DEV Connection**
	XBIB-U-DEV Connection
2 - DOUT	RXD – J2 pin 2 (RS-232), J5 pin 2
	RXD – U6 pin 24 (USB), J7 pin 2
3 - DIN/CONFIG	TXD – J2 pin 3 (RS-232), J5 pin 3
	TXD – U6 pin 25 (USB), J7 pin 3
4 - CD/DOUT_END/DO8	LED1 & DOUT LED circuit, J5 pin 4
	LED1 & DOUT LED circuit, J7 pin 4
5 – RESET	RESET, J5 pin 5
	RESET, J7 pin 5
6 – PWM0/RSSI	RSSI LED circuit input, J5 pin 6
	RSSI LED circuit input, J7 pin 6
7 – PWM1	LED2, J5 pin 7
	LED2, J7 pin 7
8 – reserved (MC13192 BKGND Debug)	J1 pin 1 (Programming/Debug Header, J5 pin 8
	J2 pin 1 (Programming/Debug Header, J7 pin 8
9 – *DTR/SLEEP_RQ/DI8	DTR – J2 pin 4 (RS-232), J5 pin 9
	DTR# – U6 pin 21 (USB), J5 pin 9
11 – RF_TX/AD4/DIO4	LED3, J8 – pin 11
	LED3, J8 – pin 11
12 – *CTS/DIO7	CTS - J2 pin 8 – (RS-232), J8 pin 10
	CTS# U6 pin 22 (USB), J8 pin 10
13 - PWRDN	J8 pin 9
	J8 pin 9
14 – VREF	J8 pin 7
	J8 pin 7
15 – Associate/AD5/DIO5	LED4, J8 pin 6
	LED4, J8 pin 6
16 - *RTS /DIO8	RTS- J2 pin 7 (RS-232), J8 pin 5
	RTS# - U6 pin 23 (USB), J8 pin 5
17 – COORD_SEL/AD3/DIO3	SW1, J8 pin 4
	SW1, J8 pin 4
18 – AD2/DIO2	SW2, J8 pin 3
	SW2, J8 pin 3
19 – AD1/DIO1	SW3, J8 pin 2
	SW3, J8 pin 2
20 – AD0/DIO0	SW4, J8 pin 1
	SW4, J8 pin 1

Asterisk (*) indicates the pin is low-asserted.

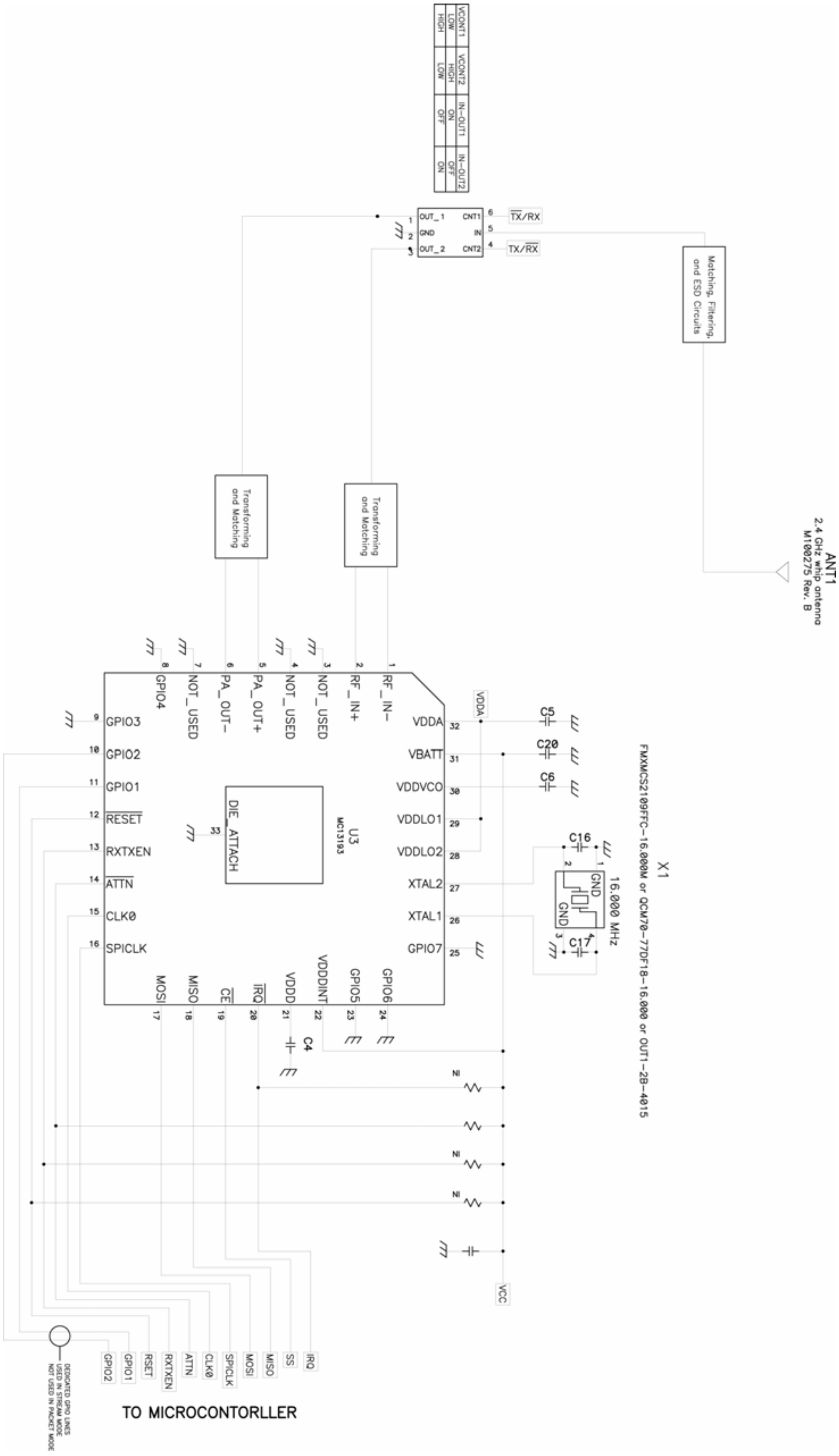
Appendix A: Reference Schematics

XBee OEM RF Module

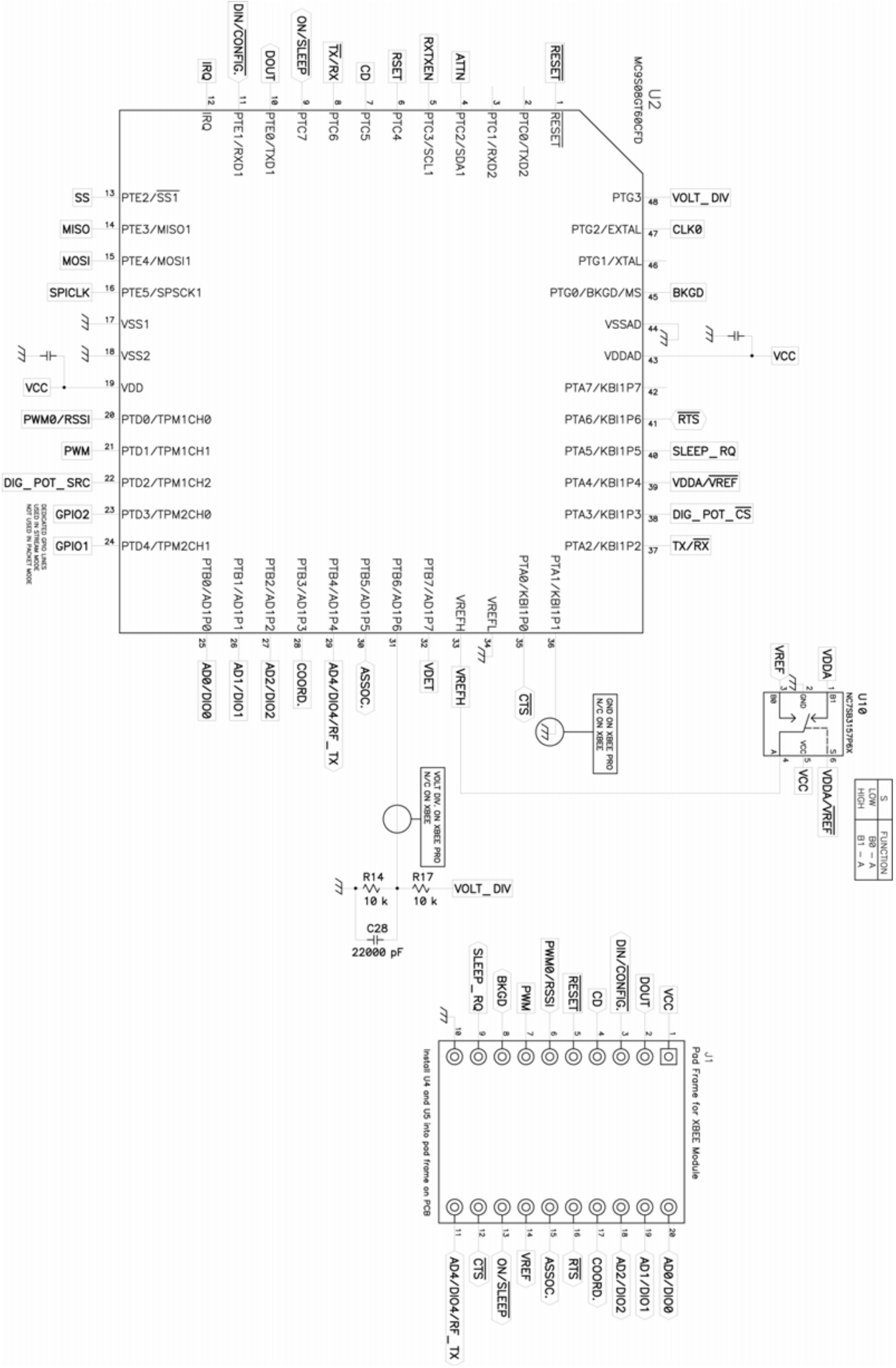
XBee Module - Simplified schematic (1 of 2)



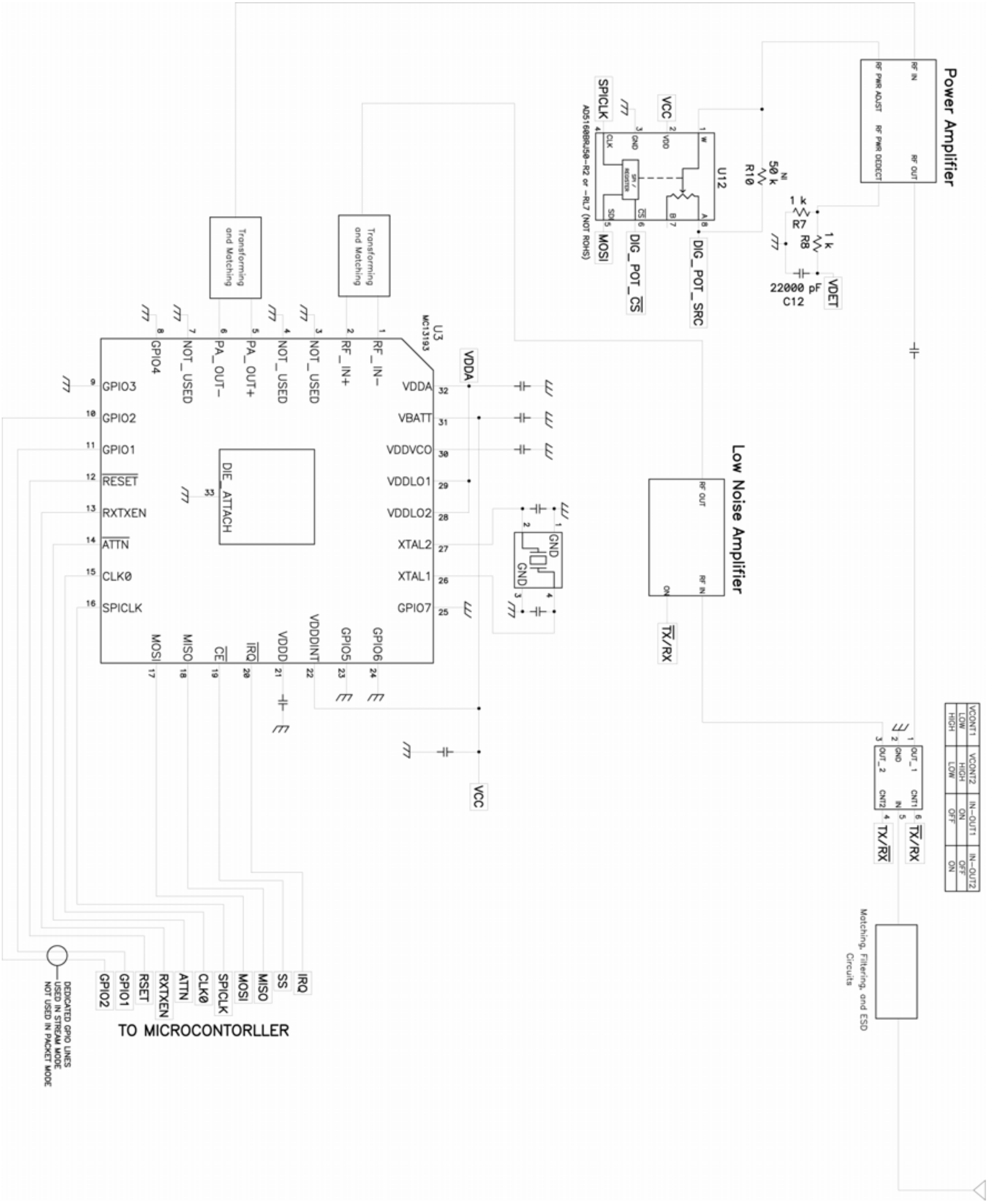
XBee Module - Simplified schematic (2 of 2)



XBee-PRO Module - Simplified schematic (1 of 2)

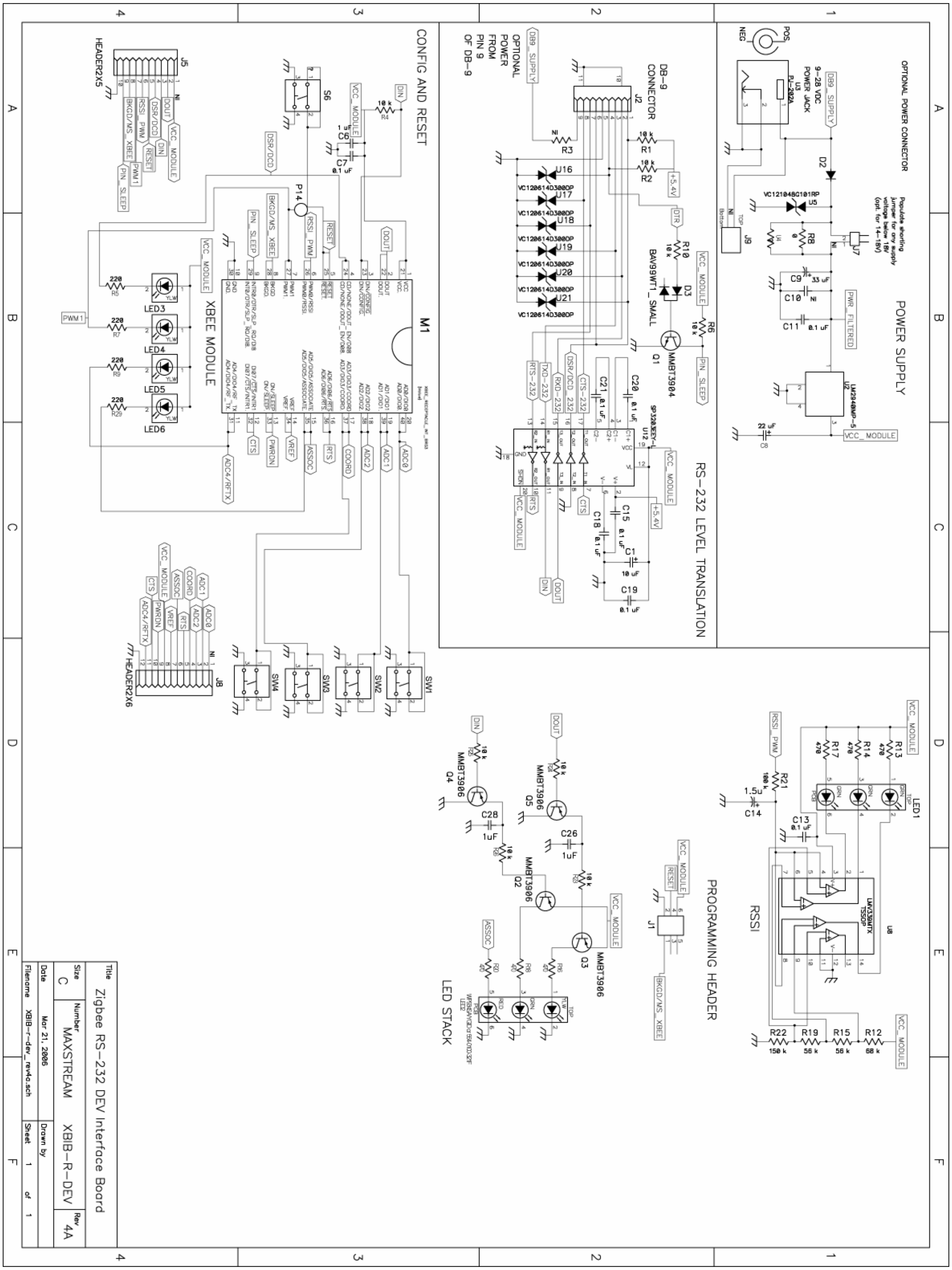


XBee-PRO Module - Simplified schematic (2 of 2)

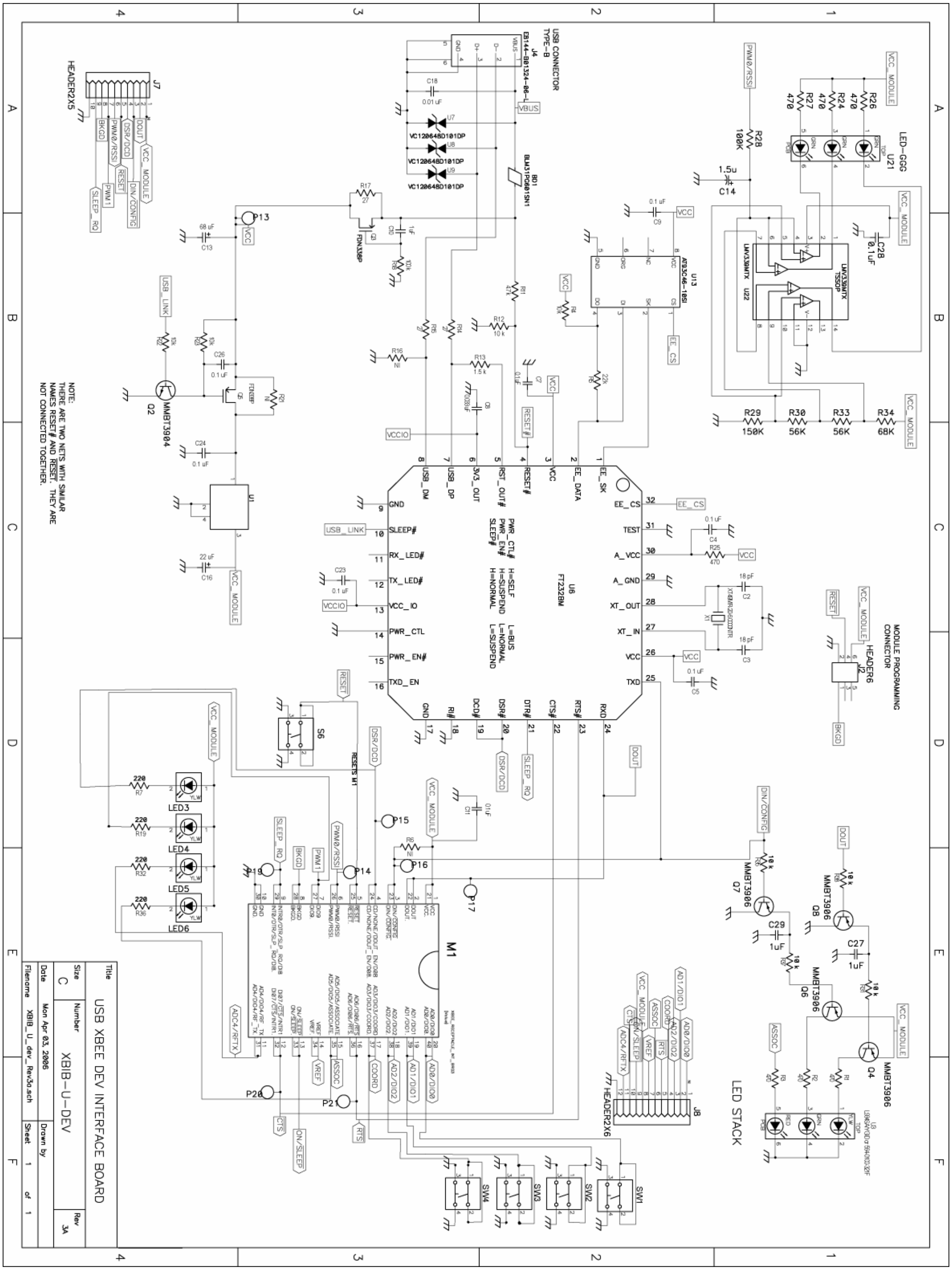


VCONT1	VCONT2	IN—OUT1	IN—OUT2
LOW	HIGH	ON	OFF
HIGH	LOW	OFF	ON

XBIB-R-DEV RS-232 Development Board - Reference Schematic



XBIB-U-DEV USB Development Board - Reference Schematic


USB XBEE DEV INTERFACE BOARD
Title
Size
C
Number
XBIB-U-DEV
Rev
3A
Date
Mon Apr 03, 2006
Drawn by
Filename
XBIB_U_dev_Rev3a.sch
Sheet
1
of
1
NOTE:
THERE ARE TWO NETS WITH SIMILAR NAMES RESET# AND RESET. THEY ARE NOT CONNECTED TOGETHER.
USB CONNECTOR TYPE-B
J4
U6
FT232BM
U13
AT93C46-10SI
U21
LED-GGG
U22
LMV339MTX
TSSOP
J7
HEADER2X5
J8
HEADER2X6
J2
HEADER6
MODULE PROGRAMMING CONNECTOR
M1
LED STACK
Q2
MMBT3904
Q4
Q6
Q7
Q8
MMBT3906
S6
SW1
SW2
SW3
SW4
LED3
LED4
LED5
LED6
U1
P13
P14
P15
P16
P17
P19
P20
P21
VCC_MODULE
VCC
VCCIO
VBUS
USB_LINK
RESET
RESET#
DOUT
DIN/CONFIG
DSR/DCD
PWM0/RSSI
PWM1
BKGD
SLEEP_RQ
ON/SLEEP
CTS
RTS
ASSOC
VREF
COORD
AD0/DIO0
AD1/DIO1
AD2/DIO2
AD4/RFTX
EE_CS
PWR_CTL# H=SELF L=BUS
PWR_EN# H=SUSPEND L=NORMAL
SLEEP# H=NORMAL L=SUSPEND
R29 150K
R30 56K
R33 56K
R34 68K
R28 100K
C14 1.5u
C28 0.1uF
C29 1uF
C27 1uF
R24 470
R26 470
R27 470
220
RESETS M1