FT232BM Designers Guide
Version 1.1

Introduction

Welcome to the FT232BM Designer's Guide. The Designers Guide includes printouts of a number of FT232BM reference schematics and explanations of the key points of each schematic. These are intended to be used in conjunction with the FT232BM data sheet, the current version of which should also be downloaded from the FTDI web site.

The schematic files are downloadable separately as a ZIP archive which contains the schematics both in OrCAD SDT 16-bit DOS format and in OrCAD Capture for Windows 32-bit format.

The OrCAD SDT 16-bit DOS format schematics are readable by OrCAD SDT version 3.2 and above. These consist of files with the following extensions –

- .sch = OrCAD 16-bit DOS binary schematic file
- .lib = OrCAD 16-bit DOS binary component library file
- .src = OrCAD DOS library source (text) file

The OrCAD Capture for Windows schematics are readable by OrCAD Capture version 7.2 and above. These consist of a file with a .dsn extension.

Notes for Protel users

OrCAD 16-bit DOS schematics can be imported into Protel schematic capture for Windows. Before reading in the schematic (.sch) file, create a Protel library first by reading in the OrCAD library source (.src) file and save it in Protel binary library format. Both OrCAD and Protel use the same default extensions for schematic and library files, so if you do not wish to overwrite the original OrCAD files, save the Protel versions to a different folder.

Figure 1.0 FT232BM – 5 volt Bus Powered Example Schematic (232-5VB)

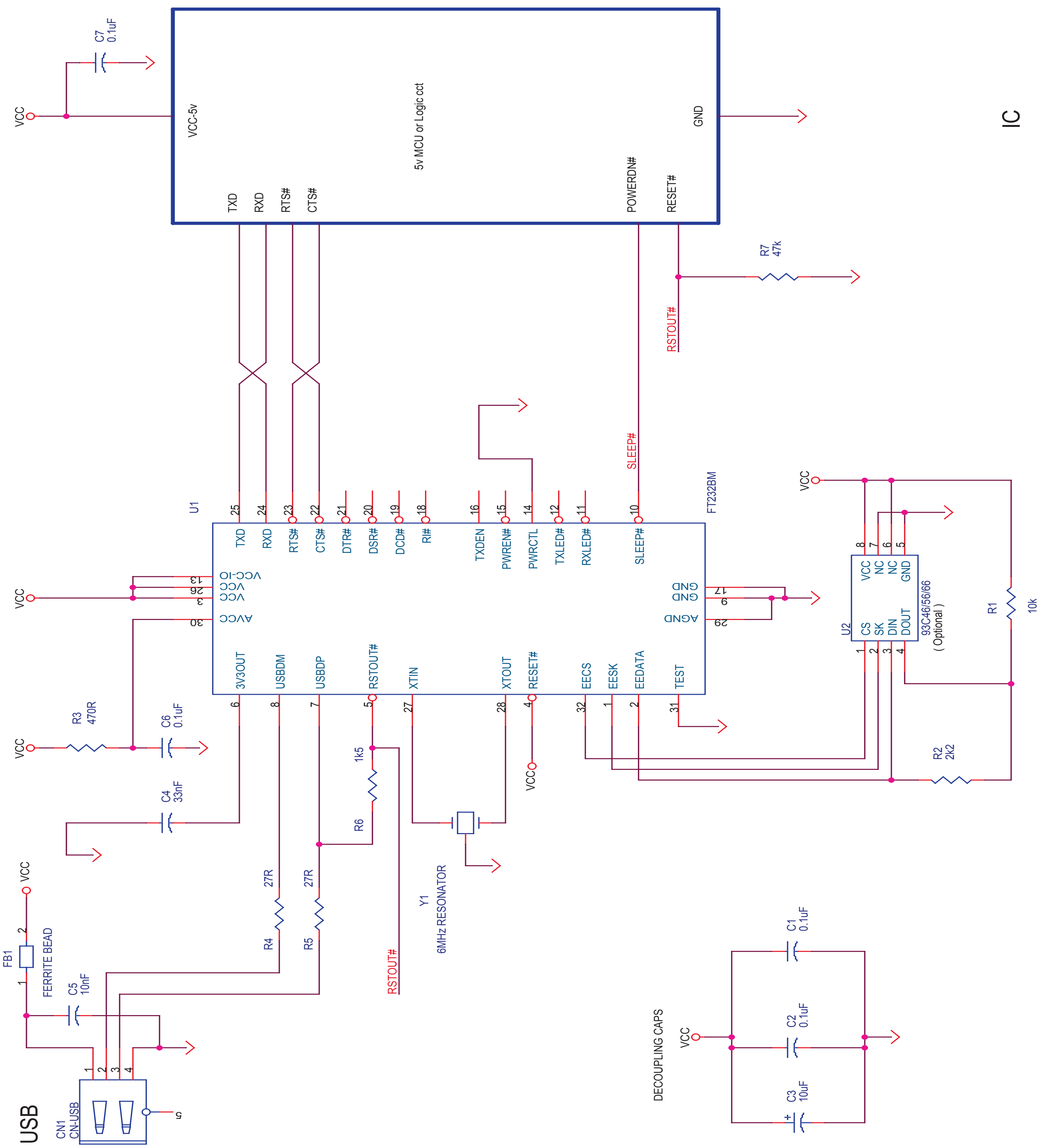

Figure 1.0 is an example of a 5 volt, USB bus powered design using the FT232BM connected to a 5v MCU or other external logic.

- **In this example, we assume that the total current of the design is <= 100mA (low power), and that the MCU / logic can detect USB suspend mode using either the SLEEP# or PWREN# pins of the FT232BM and put itself and any circuitry it is controlling into a low power state in order to meet the total USB suspend current requirement of 500uA or less.**
- **RSTOUT# is used to provide a power-on reset to the external logic in this example. If the MCU has it's own power-on reset logic then there is usually no need to use RSTOUT# to reset the device and this connection and the 47k pull-down can be omitted.**
- **PWRCTL is tied to GND to tell the device to indicate a bus powered device in it's USB descriptor.**
- **RTS / CTS handshaking is used in this example. If the MCU has no dedicated handshaking signals then general purpose IO pins can usually be used to implement the handshaking. If the MCU is guaranteed to accept data sent from the FT232BM at the programmed baud rate, then a single wire handshake will do (tie CTS# of the FT232BM to GND).**

General Design Notes:

- **SLEEP# goes inactive (high) at power-on and goes low during USB suspend. PWREN# is high on power-on and only goes low (active) after the device has been configured (successfully enumerated) by USB. During USB suspend PWREN# will go high – the opposite polarity to SLEEP#. For a low power bus powered USB device , either SLEEP# or PWREN# can be used for power control, however for a high power bus powered USB device (100mA .. 500mA) you <u>must</u> use PWREN# for power control as no USB device is allowed to draw more than 100mA from the bus until USB configuration is complete.**
- **RSTOUT# has no pull-down capability – it drives to 3.3v when not in reset, and goes tri-state during power-on reset. If used to reset an external device, a pull-down resistor must be added to make it low during reset.**
- **When RTS/CTS hardware handshaking is enabled CTS# can be used to stop the FT232BM transmitting data to the MCU / external logic. When CTS# is active (low) the FT232BM will transmit any data in it's internal buffers. On taking CTS# high, the FT232BM will stop transmitting data. Due to the asynchronous nature of the interface, there is a latency of 0 to 3 characters between taking CTS# high and data transmission stopping. The FT232BM drives RTS# high when the available buffer space inside the device drops below 32 bytes. This allows the MCU / logic to continue to send up to 30 characters to the FT232BM after RTS# goes high without causing buffer over-run.**
- **A suitable 3-pin ceramic resonator could be a Murata CSTLS6M00G53 or equivalent. If you prefer to use a 2 pin resonator or a crystal refer to Figures 4 and 5 of the FT232BM data sheet for details.**
- **A suitable ferrite bead could be a Steward MI0805K400R-00 or equivalent. This is also available from DigiKey as Part #** 240-1035-1. For specifications consult the Steward web site - **http://www.steward.com**

Figure 2.0 FT232BM – 5 volt Self Powered Example Schematic (232-5VS)

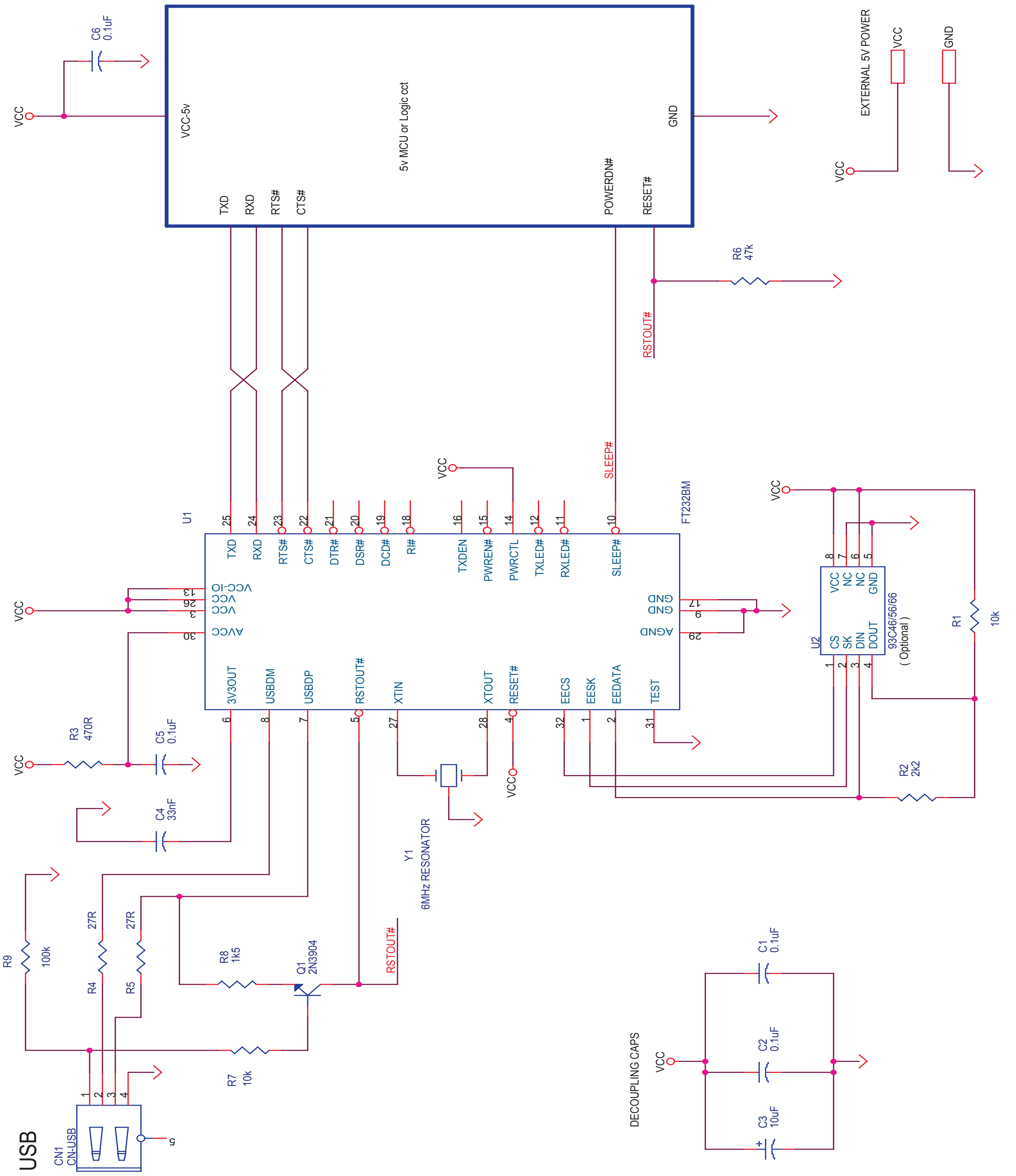

Figure 2.0 is an example of a 5 volt, USB self powered design using the FT232BM connected to a 5v MCU or other external logic. A USB self power design has it's own PSU and does not draw it's power from the USB bus. In such a case, no special care need be taken to meet the USB suspend current (0.5mA) as the device does not get its power from the USB port.

- **In this case it is still useful to connect SLEEP# (or PWREN#) to the CPU as this will let the CPU know that the PC is in suspend mode and thus unable to communicate with the device. If the device requires to "wake up" the PC then the MCU should connect one of it's IO Ports to the Ring Indicator pin (RI#). The default state of RI# should be high - strobing this low for a few milliseconds then taking it high again will cause a USB resume sequence thus requesting the PC to wake up. To use this feature, Remote Wake-Up must be enabled in the 93C46 EEPROM.**
- **RSTOUT# is used to provide a power-on reset to the external logic in this example. If the MCU has it's own power-on reset logic then there is usually no need to use RSTOUT# to reset the device and this connection and the 47k pull-down can be omitted.**
- **PWRCTL is tied to VCC to tell the device to indicate a self powered device in it's USB descriptor.**
- **RTS / CTS handshaking is used in this example. If the MCU has no dedicated handshaking signals then general purpose IO pins can usually be used to implement the handshaking. If the MCU is guaranteed to accept data sent from the FT232BM at the programmed baud rate, then a single wire handshake will do (tie CTS# of the FT232BM to GND).**
- **Self powered designs should NOT force current back into the Host PC (or HUB) via the USB Port when the said Host / Hub is powered down and the self powered device is still powered-up from it's own PSU. This rule includes injecting current into the powered down Host / Hub via the 1k5 pull-up on USB D+. Failure to do this can result in unreliable operation in the field. This is an integral part of the USB specification and applies to <u>all</u> USB Self Powered devices (not just FT232BM peripherals). In this design, a simple NPN transistor circuit is used to prevent this condition. When the Host / Hub is powered up the transistor is turned on (saturated) and pulls the top end of the 1k5 resistor on USB D+ to RESETO# (3.3v nominal unless the FT232BM is still in reset). If the Host / Hub powers down, the transistor turns off, so no current is forced down USB D+ in this condition.**

General Design Notes – See Previous Example

Figure 3.0

FT232BM – 5 volt Bus Powered Example Schematic with Power Switching (232-5VSW)

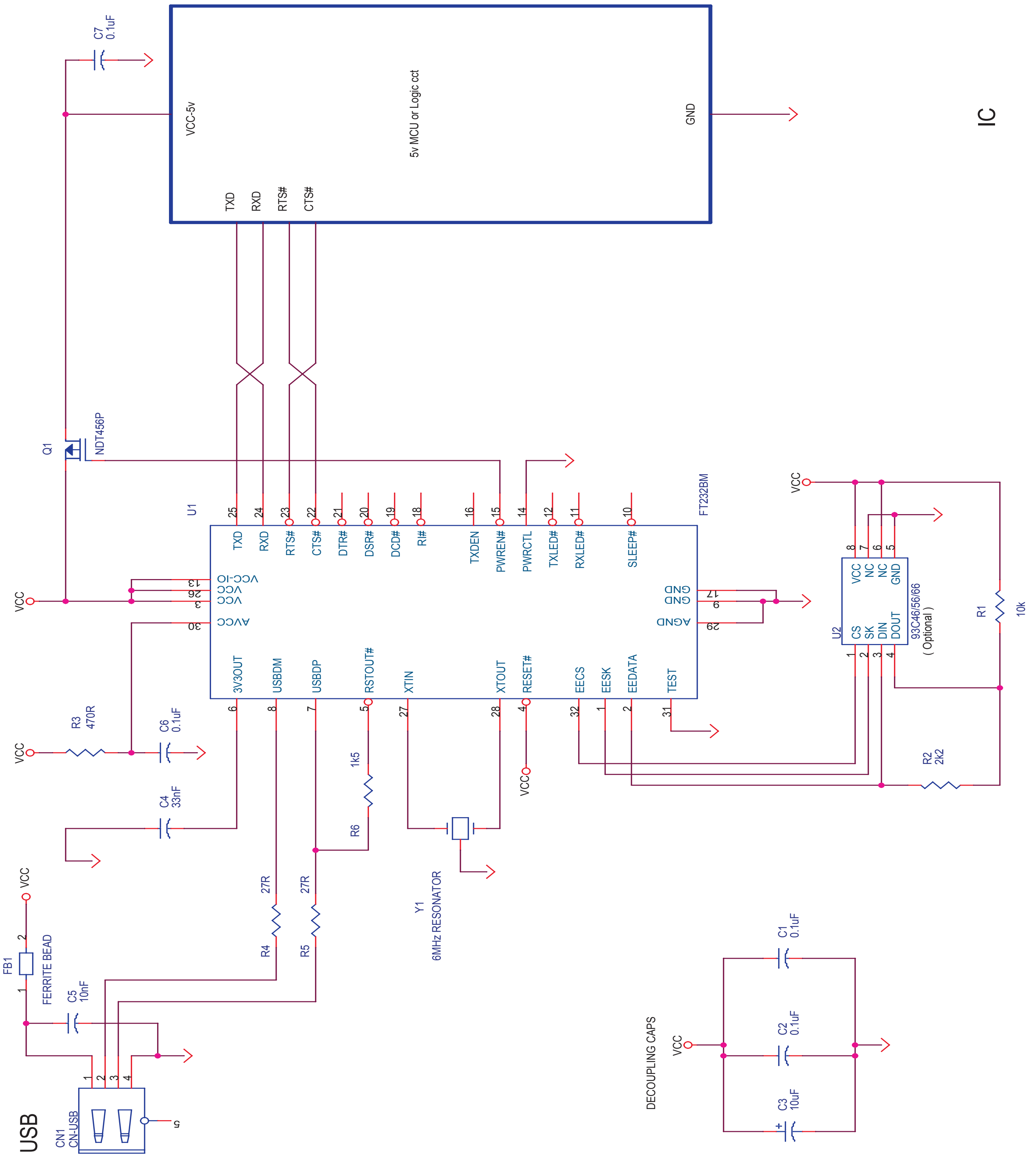

Figure 3.0 is an example of a 5 volt, USB bus powered design using the FT232BM connected to a 5v MCU or other external logic. In this design, the FT232BM controls the power to the auxiliary circuitry using PWEREN# to shut off power to this circuitry when –

1. **The FT232BM is in reset.**
2. **The FT232BM has not yet been configured (successfully recognised and enumerated over USB).**
3. **USB is not in suspend / sleep mode.**

- **A P-Channel Logic Level MOSFET is used as a power switch to control the power to the auxiliary devices – in this example we use a Fairchild NDT456. If the auxiliary circuitry normally consumes over 100mA OR has a power-on surge current that is likely to exceed the USB requirements, then we would recommend using a dedicated high-side power switch incorporating surge control (such as a Micrel MIC2025) in place of the MOSFET.**
- **When using this circuit, enable the "Pull-Down on Suspend" option in the EEPROM. This will ensure minimum leakage current during sleep (suspend) mode by gently pulling down the UART interface pins of the FT232BM pins to GND during USB suspend.**
- **The auxiliary circuitry attached to the FT232BM device must have it's own power-on-reset circuitry and should NOT use RESETO# to generate a reset for this circuitry. RESETO# does not generate a reset during USB sleep (suspend) when the auxiliary logic is powered-off, thus cannot be used as a reset in this case.**
- **A "USB High-Power Bus Powered Device" (one that consumes more than 100mA and up to 500mA) of current from the USB bus during normal operation must use this power control feature to remain compliant as the USB specification does not allow a USB peripheral to draw more than 100mA of current from the USB Bus until the device has been successfully enumerated. A "USB High-Power Bus Powered Device" cannot be plugged into a USB Bus-Powered Hub as these can only supply 100mA per USB port.**
- **The Power (current) consumption of the device is set in a field in the 93C46 EEPROM attached to the FT232BM. A "USB High-Power Bus Powered Device" must use the 93C46 to inform the system of it's power requirements.**
- **PWRCTL is tied to GND to tell the device to indicate a bus powered device in it's USB descriptor.**
- **RTS / CTS handshaking is used in this example. If the MCU has no dedicated handshaking signals then general purpose IO pins can usually be used to implement the handshaking. If the MCU is guaranteed to accept data sent from the FT232BM at the programmed baud rate, then a single wire handshake will do (tie CTS# of the FT232BM to GND).**

General Design Notes – See Previous Examples

Figure 4.0 FT232BM – 3.3 volt Bus Powered Example Schematic (232-3VB)

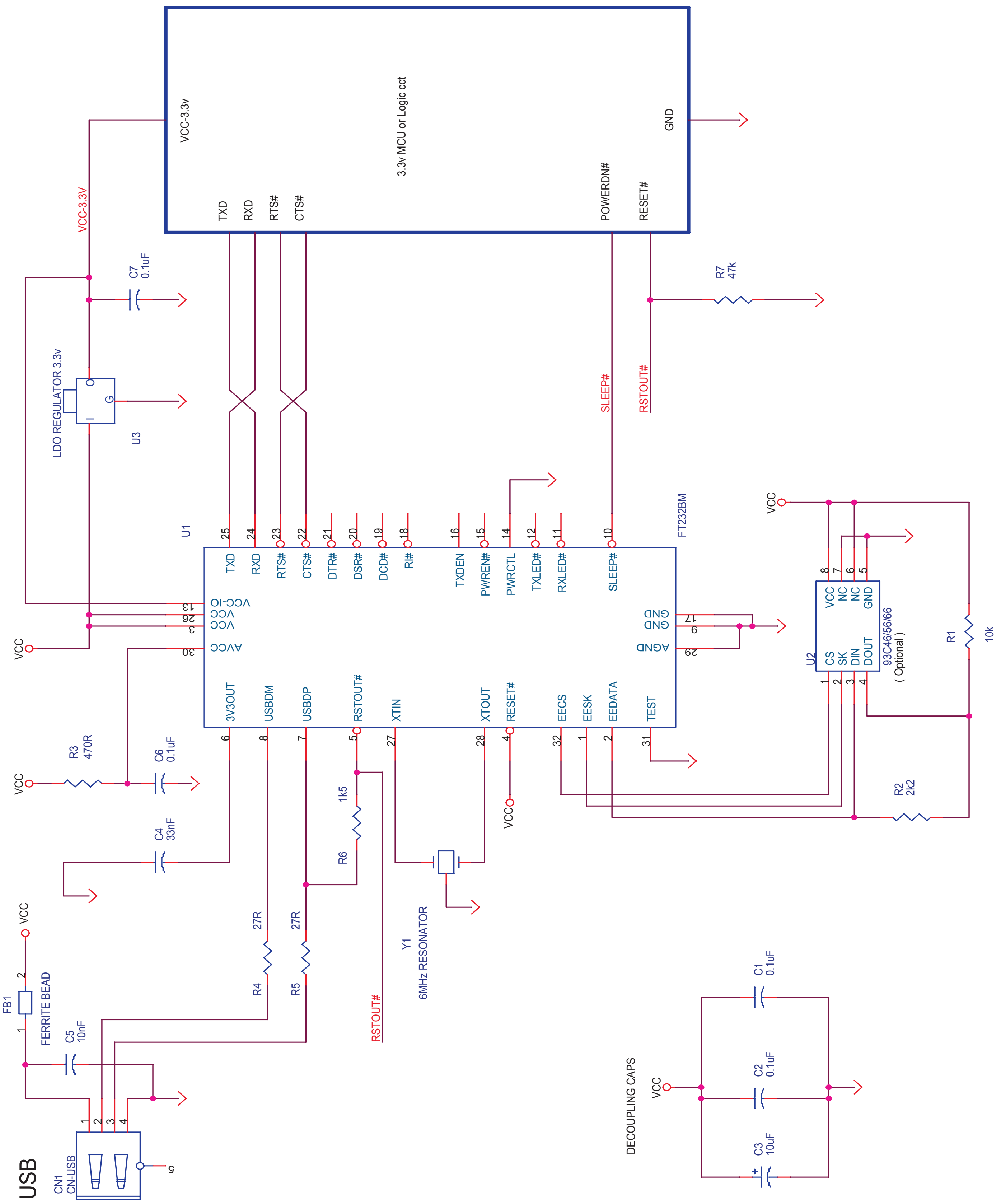

Figure 4.0 is an example of a 3.3 volt, USB bus powered design using the FT232BM connected to a 3.3v MCU or other external logic.

- **The main difference between this circuit and the 5 volt circuit of Figure 1.0 is that a 3.3 volt LDO regulator i.c. is used to provide a 3.3v supply to the auxiliary circuiry.**
- **VCC-IO is driven from the 3.3v LDO regulator i.c. in order to drive the UART interface from the FT232BM to the MCU / external logic at 3.3v level instead of 5v level.**
- **As the USB supply rail can drop to 4.4 volts or less under load, an LDO (Low Dropout) voltage regulator must be used in this instance.**
- **The 3.3v LDO voltage regulator must also have a low quiescent (no load) current in order to ensure that the USB suspend current requirement (<= 500uA) is met during USB suspend.**
- **In this example, we assume that the total current of the design is <= 100mA (low power), and that the MCU / logic can detect USB suspend mode using either the SLEEP# or PWREN# pins of the FT232BM and put itself and any circuitry it is controlling into a low power state in order to meet the total USB suspend current requirement of 500uA or less.**
- **RSTOUT# is used to provide a power-on reset to the external logic in this example. If the MCU has it's own power-on reset logic then there is usually no need to use RSTOUT# to reset the device and this connection and the 47k pull-down can be omitted.**
- **PWRCTL is tied to GND to tell the device to indicate a bus powered device in it's USB descriptor.**
- **RTS / CTS handshaking is used in this example. If the MCU has no dedicated handshaking signals then general purpose IO pins can usually be used to implement the handshaking. If the MCU is guaranteed to accept data sent from the FT232BM at the programmed baud rate, then a single wire handshake will do (tie CTS# of the FT232BM to GND).**

General Design Notes – See Previous Examples

Figure 5.0 FT232BM – 3.3 volt Self Powered Example Schematic (232-3VS)

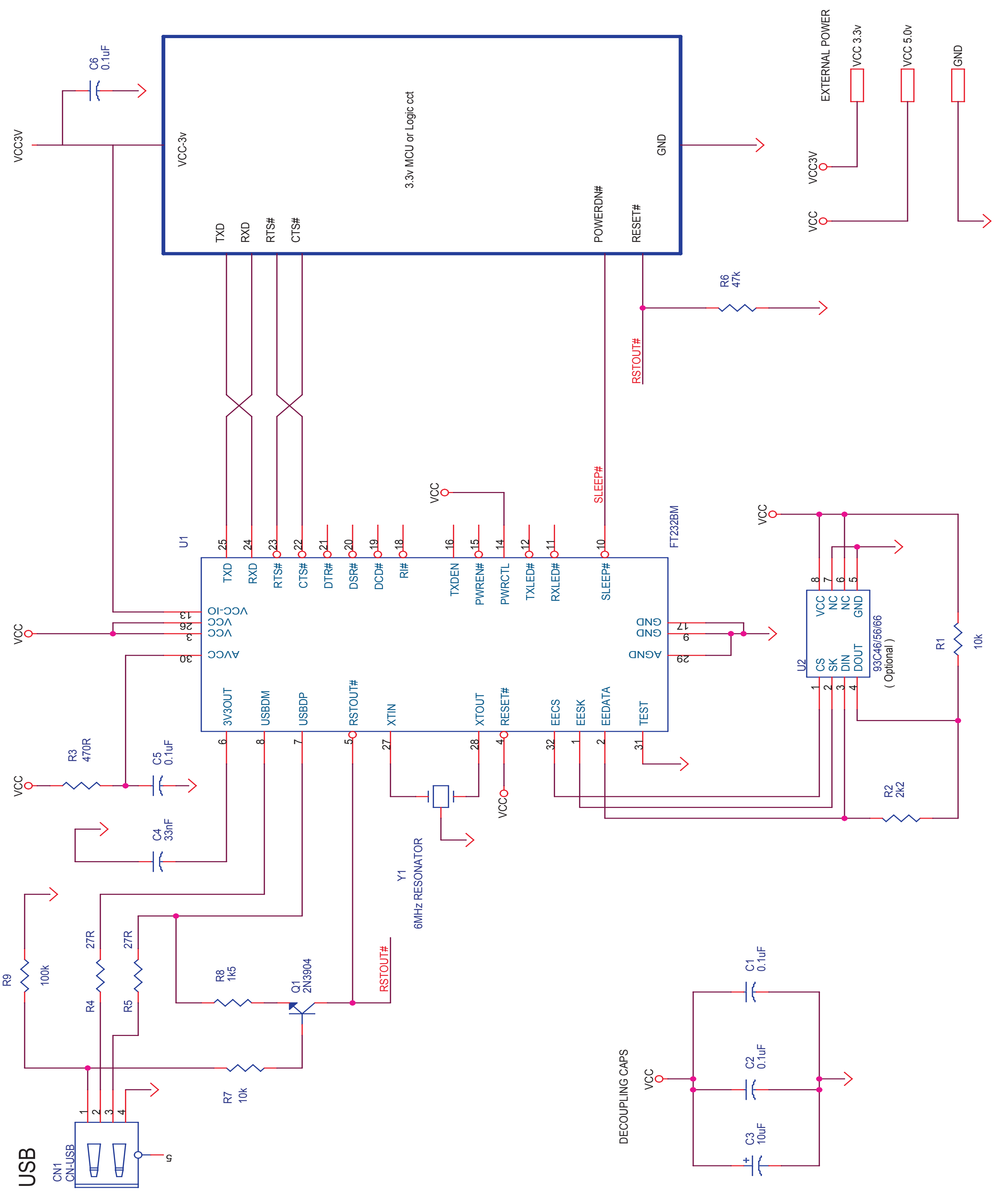

Figure 5.0 is an example of a 3.3 volt, USB self powered design using the FT232BM connected to a 3.3v MCU or other external logic. A USB self power design has it's own PSU and does not draw it's power from the USB bus. In such a case, no special care need be taken to meet the USB suspend current (0.5mA) as the device does not get its power from the USB port. The differences between this circuit and that of Figure 2.0 are minimal. See the notes in Figure 2 for the main details.

- **In this case the internal PSU need to supply 3.3 volts to the auxiliary circuitry and 5 volts to the FT232BM i.c.**
- **The VCCIO power line to the FT232BM is driven from the 3.3v supply in order to drive the auxiliary logic at the correct voltage level.**

Figure 6.0 FT232BM –5v BUS Powered USB => RS232 Converter Example Schematic (USB-232B)

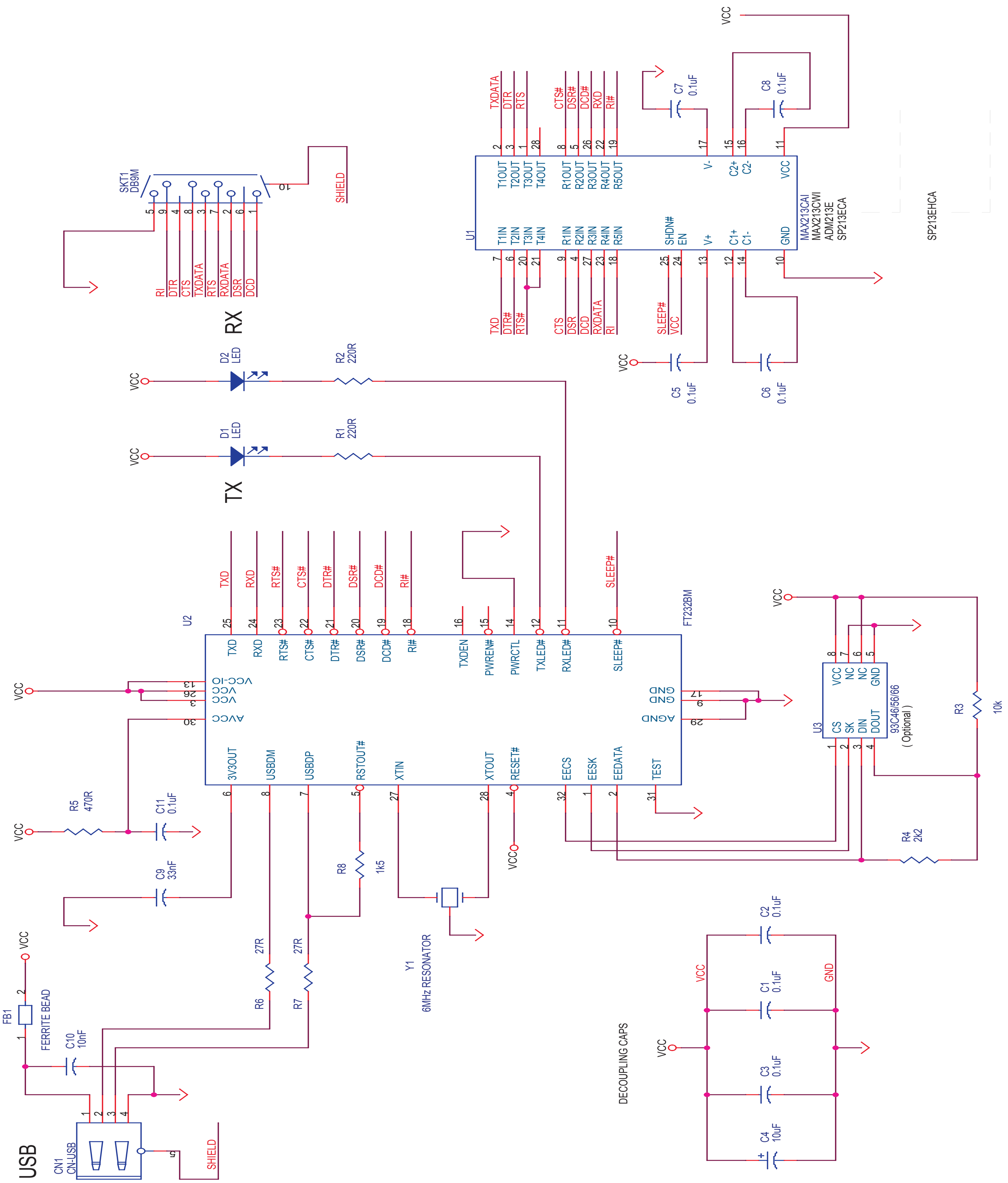

Figure 6.0 is an example of a 5 volt, USB bus powered design using the FT232BM connected to a TTL ⇔ RS232 level converter i.c .

- **For RS232 applications, the baud rate of the finished product is limited by the ac. driving characteristics of the level converter i.c. rather than that of the FT232BM.**
- **This example uses the popular "213" series of TTL to RS232 level converters. These devices have 4 transmitters and 5 receivers in a 28 LD SSOP package and feature an in-built voltage converter to convert the 5v (nominal) VCC to the +/- 9volts required by RS232. An important feature of these devices is the SHDN# pin which can power down the device to a low quiescent current during USB suspend mode**
- **The device used in this schematic is a Sipex SP213EHCA which is capable of RS232 communication at up to 500k baud. If a lower baud rate is acceptable, then several pin compatible alternatives are available such as Sipex SP213ECA , Maxim MAX213CAI and Analog Devices ADM213E which are good for communication at up to 115,200 baud. If a higher baud rate is desired, use a Maxim MAX3245CAI part which is capable of RS232 communication at rates of up to 1M baud.**
- **Note : the MAX3245 is not pin compatible with the 213 series devices, also it's SHDN pin is active high so connect this to PWREN# instead of SLEEP#.**

Document Revision History

DG232B Version 1.0 – Initial document created 05 August 2002
DG232B Version 1.1 - Created 06 August 2002
- Added USB-232B application

Disclaimer

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.
Future Technology Devices International Ltd. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

Contact Information

Future Technology Devices Intl. Limited
St. George's Studios
93/97 St. George's Road,
Glasgow G3 6JA,
United Kingdom.
Tel : +44 (0)141 353 2565
Fax : +44 (0)141 353 2656
E-Mail (Sales) : sales@ftdichip.com
E-Mail (Support) : support@ftdichip.com
E-Mail (General Enquiries) : admin@ftdichip.com
Web Site URL : http://www.ftdichip.com

Agents and Sales Representatives

At the time of writing our Sales Network covers over 40 different countries world-wide. Please visit the Sales Network page of our Web Site site for the contact details our distributor(s) in your country.