FT232BM USB UART (USB - Serial) I.C.

The FT232BM is the 2nd generation of FTDI's popular USB UART i.c. This device not only adds extra functionality to it's FT8U232AM predecessor and reduces external component count, but also maintains a high degree of pin compatibility with the original, making it easy to upgrade or cost reduce existing designs as well as increasing the potential for using the device in new application areas.

1.0 Features

HARDWARE FEATURES

- Single Chip USB ⇔Asynchronous Serial Data Transfer
- Full Handshaking & Modem Interface Signals
- UART I/F Supports 7 / 8 Bit Data, 1 / 2 Stop Bits and Odd/Even/Mark/Space/No Parity
- Data rate 300 => 3M Baud (TLL)
- Data rate 300 => 1M Baud (RS232)
- Data rate 300 => 3M Baud (RS422/RS485)
- 384 Byte Receive Buffer / 128 Byte Transmit Buffer for high data throughput
- Adjustable RX buffer timeout
- Full hardware assisted hardware or X-On / X-Off handshaking
- In-built support for event characters and line break condition
- Auto Transmit Buffer control for RS485
- Support for USB Suspend / Resume through SLEEP# and RI# pins
- Support for high power USB Bus powered devices through PWREN# pin
- Integrated level converter on UART and control signals for interfacing to 5v and 3.3v logic
- Integrated 3.3v regulator for USB IO
- Integrated Power-On-Reset circuit
- Integrated 6MHz – 48Mhz clock multiplier PLL
- USB Bulk or Isocronous data transfer modes
- 4.4v to 5.25v single supply operation
- UHCI / OHCI / EHCI host controller compatible
- USB 1.1 and USB 2.0 compatible
- USB VID, PID , Serial Number and Product Description strings in external EEPROM
- EEPROM programmable on-board via USB

VIRTUAL COM PORT (VCP) DRIVERS for

- Windows 98 and Windows 98 SE
- Windows 2000 / ME / XP
- Windows CE **
- MAC OS-8 and OS-9
- MAC OS-X **
- Linux 2.40 and greater

D2XX (USB Direct Drivers + DLL S/W Interface)

- Windows 98 and Windows 98 SE
- Windows 2000 / ME / XP

APPLICATION AREAS

- USB ⇔RS232 Converters
- USB ⇔ RS422 / RS485 Converters
- Upgrading RS232 Legacy Peripherals to USB
- Cellular and Cordless Phone USB data transfer cables and interfaces
- Interfacing MCU based designs to USB
- USB Audio and Low Bandwidth Video data transfer
- PDA ⇔ USB data transfer
- USB Smart Card Readers
- Set Top Box (S.T.B.) PC - USB interface
- USB Hardware Modems
- USB Wireless Modems
- USB Instrumentation
- USB Bar Code Readers

[** = In planning or under development]

2.0 Enhancements

This section summarises the enhancements of the 2nd generation device compared to it's FT8U232AM predecessor. For further details, consult the device pin-out description and functional descriptions.

- **Integrated Power-On-Reset (POR) Circuit**
  The device now incorporates an internal POR function. The existing RESET# pin is maintained in order to allow external logic to reset the device where required, however for many applications this pin can now be either left N/C or hard wired to VCC. In addition, a new reset output pin (RSTO#) is provided in order to allow the new POR circuit to provide a stable reset to external MCU and other devices. RSTO# was the TEST pin on the previous generation of devices.

- **Integrated RCCLK Circuit**
  In the previous devices, an external RC circuit was required to ensure that the oscillator and clock multiplier PLL frequency was stable prior to enabling the clock internal to the device. This circuit is now embedded on-chip – the pin assigned to this function is now designated as the TEST pin and should be tied to GND for normal operation.

- **Integrated Level Converter on UART interface and control signals**
  The previous devices would drive the UART and control signals at 5v CMOS logic levels. The new device has a separate VCC-IO pin allowing the device to directly interface to 3.3v and other logic families without the need for external level converter i.c.'s

- **Improved Power Management control for USB Bus Powered, high current devices**
  The previous devices had a USBEN pin, which became active when the device was enumerated by USB. To provide power control, this signal had to be externally gated with SLEEP# and RESET#. This gating is now done on-chip - USBEN has now been replaced with the new PWREN# signal which can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. A new EEPROM based option makes the device pull gently down it's UART interface lines when the power is shut off (PWREN# is High). In this mode, any residual voltage on external circuitry is bled to GND when power is removed thus ensuring that external circuitry controlled by PWREN# resets reliably when power is restored.
- **Lower Suspend Current**
  Integration of RCCLK within the device and internal design improvements reduce the suspend current of the FT232BM to under 200uA (excluding the 1.5k pull-up on USB DP) in USB suspend mode. This allows greater margin for peripherals to meet the USB Suspend current limit of 500uA.
- **Support for USB Isocronous Transfers**
  Whilst USB Bulk transfer is usually the best choice for data transfer, the scheduling time of the data is not guaranteed. For applications where scheduling latency takes priority over data integrity such as transferring audio and low bandwidth video data, the new device now offers an option of USB Isocronous transfer via an option bit in the EEPROM.

- **Programmable Receive Buffer Timeout**
  In the previous device, the receive buffer timeout used to flush remaining data from the receive buffer was fixed at 16ms timeout. This timeout is now programmable over USB in 1ms increments from 1ms to 255ms, thus allowing the device to be better optimised for protocols requiring faster response times from short data packets.

- **TXDEN Timing fix**
  TXDEN timing has now been fixed to remove the external delay that was previously required for RS485 applications at high baud rates. TXDEN now works correctly during a transmit send-break condition.

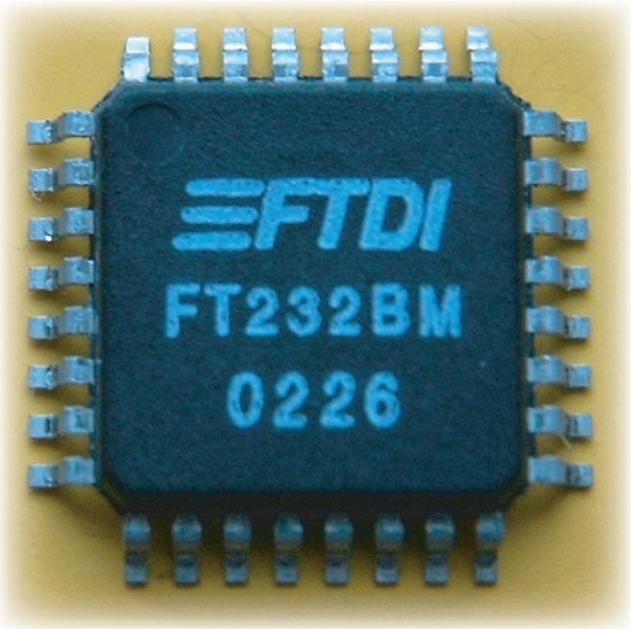
- **Relaxed VCC Decoupling**
  The 2nd generation devices now incorporate a level of on-chip VCC decoupling. Though this does not eliminate the need for external decoupling capacitors, it significantly improves the ease of pcb design requirements to meet FCC,CE and other EMI related specifications.

- **Improved PreScaler Granularity**
  The previous version of the Prescaler supported division by (n + 0), (n + 0.125), (n + 0.25) and (n + 0.5) where n is an integer between 2 and 16,384 (2^{14}). To this we have added (n + 0.375), (n + 0.625), (n + 0.75) and (n+ 0.875) which can be used to improve the accuracy of some baud rates and generate new baud rates which were previously impossible (especially with higher baud rates).

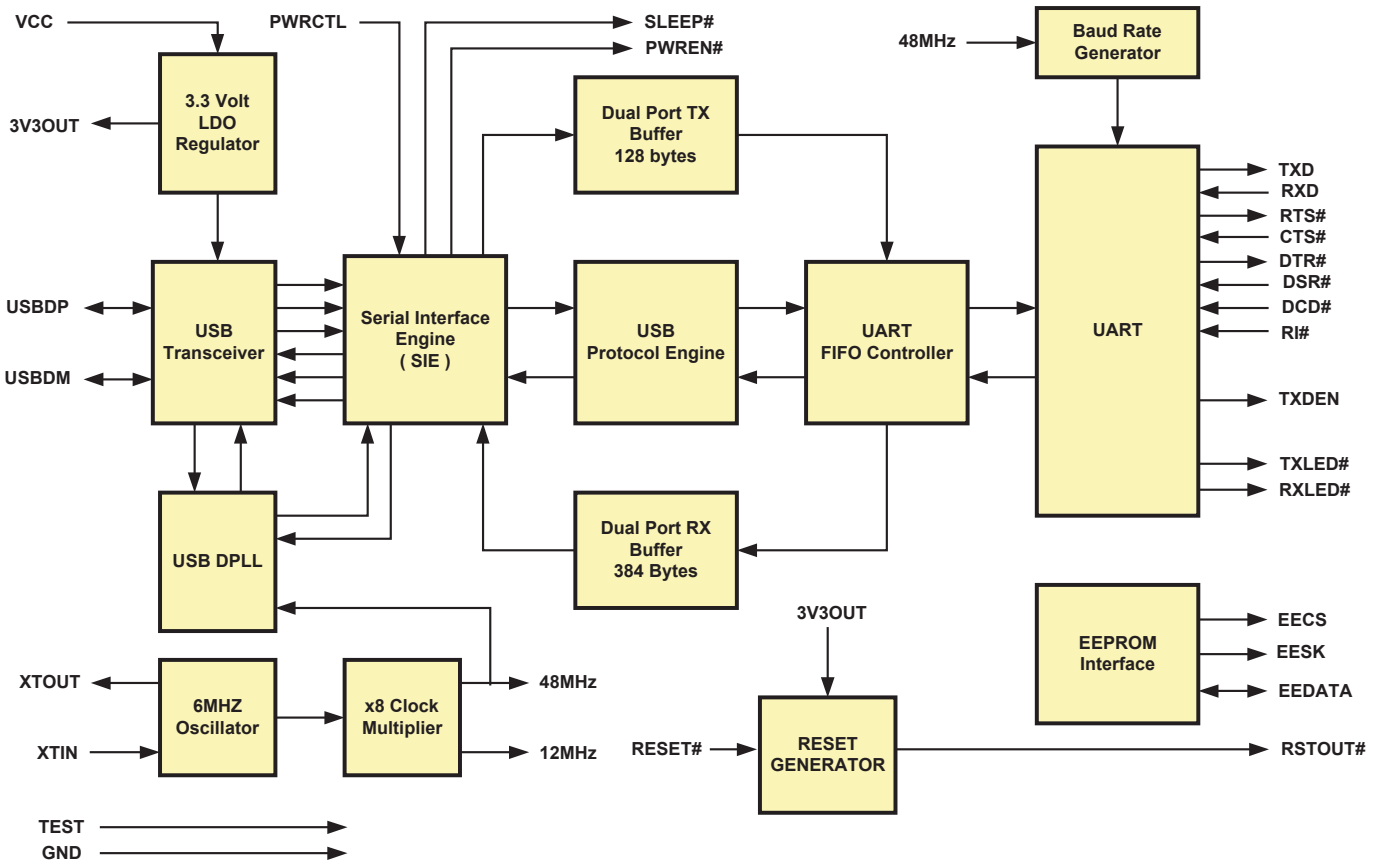
- **Bit Bang Mode**
  The 2nd generation device has a new option referred to as “Bit Bang” mode. In Bit Bang mode, the eight UART interface control lines can be switched between UART interface mode and an 8-bit Parallel IO port. Data packets can be sent to the device and they will be sequentially sent to the interface at a rate controlled by the prescaler setting. As well as allowing the device to be used stand-alone as a general purpose IO controller for example controlling lights, relays and switches, some other interesting possibilities exist. For instance, it may be possible to connect the device to an SRAM configurable FPGA as supplied by vendors such as Altera and Xilinx. The FPGA device would normally be un-configured (i.e. have no defined function) at power-up. Application software on the PC could use Bit Bang Mode to download configuration data to the FPGA which would define it’s hardware function, then after the FPGA device is configured the FT232BM can switch back into UART interface mode to allow the programmed FPGA device to communicate with the PC over USB. This approach allows a customer to create a “generic” USB peripheral who’s hardware function can be defined under control of the application software. The FPGA based hardware can be easily upgraded or totally changed simply by changing the FPGA configuration data file. Application notes, software and development modules for this application area will be available from FTDI and other 3rd parties.

- **PreScaler Divide By 1 Fix**
  The previous device had a problem when the integer part of the divisor was set to 1. In the 2nd generation device setting the prescaler value to 1 gives a baud rate of 2 million baud and setting it to zero gives a baud rate of 3 million baud. Non-integer division is not supported with divisor values of 0 and 1.

- **Less External Support Components**
  As well as eliminating the RCCLK RC network, and for most applications the need for an external reset circuit, we have also eliminated the requirement for a 100k pull-up on EECS to select 6MHz operation. When the FT232BM is being used without the configuration EEPROM, EECS, EESK and EEDATA can now be left n/c. For circuits requiring a long reset time (where the device is reset externally using a reset generator i.c., or reset is controlled by the IO port of a MCU, FPGA or ASIC device) an external transistor circuit is no longer required as the 1k5 pull-up resistor on USB DP can be wired to the RESETO# pin instead of to 3.3v. Note : RESETO# drives out at 3.3v level, not at 5v VCC level. This is the preferred configuration for new designs. In some other configurations, RSTO# can be used to reset external logic / MCU circuitry.
- **Extended EEROM Support**
  The previous generation of devices only supported EEPROM of type 93C46 (128 x 16 bit). The new devices will also work with EEPROM type 93C56 (256 x 16 bit) and 93C66 (512 x 16 bit). The extra space is not used by the device, however it is available for use by other external MCU / logic whilst the FT232BM is being held in reset.
- **USB 2.0 (full speed option)**
  A new EEPROM based option allows the FT232BM to return a USB 2.0 device descriptor as opposed to USB 1.1. Note : The device would be a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 High Speed device (480Mb/s).
- **Multiple Device Support without EEPROM**
  When no EEPROM (or a blank or invalid EEPROM) is attached to the device, the FT232BM no longer gives a serial number as part of it's USB descriptor. This allows multiple devices to be simultaneously connected to the same PC. However, we still highly recommend that EEPROM is used, as without serial numbers a device can only be identified by which hub port in the USB tree it is connected to which can change if the end user re-plugs the device into a different port.



3.0 Block Diagram (simplified)



3.1 Functional Block Descriptions

- **3.3V LDO Regulator**

  The 3.3V LDO Regulator generates the 3.3 volt reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3v power to the RSTOUT# pin. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring 3.3v nominal at a current of not greater than 5mA could also draw it's power from the 3V3OUT pin if required.

- **USB Transceiver**

  The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3 volt level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SEO and USB Reset condition detection.

- **USB DPLL**

  The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and data signals to the SIE block.

- **6MHz Oscillator**

  The 6MHz Oscillator cell generates a 6MHz reference clock input to the X8 Clock multiplier from an external 6MHz crystal or ceramic resonator.

- **x8 Clock Multiplier**
  The x8 Clock Multiplier takes the 6MHz input from the Oscillator cell and generates a 12MHz reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks. It also generates a 48MHz reference clock for the USB DPPL and the Baud Rate Generator blocks.
- **Serial Interface Engine (SIE)**
  The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 1.1 specification, it performs bit stuffing / un-stuffing and CRC5 / CRC16 generation / checking on the USB data stream.
- **USB Protocol Engine**
  The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB host controller and the commands for controlling the functional parameters of the UART.
- **Dual Port TX Buffer (128 bytes)**
  Data from the USB data out endpoint is stored in the Dual Port TX buffer and removed from the buffer to the UART transmit register under control of the UART FIFO controller.
- **Dual Port RX Buffer (384 bytes)**
  Data from the UART receive register is stored in the Dual Port RX buffer prior to being removed by the SIE on a USB request for data from the device data in endpoint.
- **UART FIFO Controller**
  The UART FIFO controller handles the transfer of data between the Dual Port RX and TX buffers and the UART transmit and receive registers.
- **UART**
  The UART performs asynchronous 7 / 8 bit Parallel to Serial and Serial to Parallel conversion of the data on the RS232 (RS422 and RS485) interface. Control signals supported by the UART include RTS, CTS, DSR , DTR, DCD and RI. The UART provides a transmitter enable control signal (TXDEN) to assist with interfacing to RS485 transceivers. The UART supports RTS/CTS, DSR/DTR and X-On/X-Off handshaking options. Handshaking, where required, is handled in hardware to ensure fast response times. The UART also supports the RS232 BREAK setting and detection conditions.
- **Baud Rate Generator**
  The Baud Rate Generator provides a x16 clock input to the UART from the 48MHz reference clock and consists of a 14 bit prescaler and 3 register bits which provide fine tuning of the baud rate (used to divide by a number plus a fraction). This determines the Baud Rate of the UART which is programmable from 183 baud to 3 million baud.
- **RESET Generator**
  The Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. An additional RESET# input and RSTOUT# output are provided to allow other devices to reset the FT232BM or the FT232BM to reset other devices respectively. During reset, RSTOUT# is high-impedance otherwise it drives out at the 3.3v provided by the onboard regulator. RSTOUT# can be used to control the 1k5 pull-up on USB DP directly where delayed USB enumeration is required. It can also be used to reset other devices. RSTOUT# will stay high-impedance for approximately 5ms after VCC has risen above 3.5v AND the device oscillator is running AND RESET# is high. RESET# should be tied to VCC unless it is a requirement to reset the device from external logic or an external reset generator i.c.

- **EEPROM Interface**

  Though the FT232BM will work without the optional EEPROM, an external 93C46 (93C56 or 93C66) EEPROM can be used to customise the USB VID, PID, Serial Number, Product Description Strings and Power Descriptor value of the FT232BM for OEM applications. The EEPROM is also required for applications where multiple FT232BM's are connected to a single PC as the drivers rely on a unique serial number for each device to bind a unique virtual COM port to each individual device. Other parameters controlled by the EEPROM include Remote Wake Up, Isochronous Transfer Mode, Soft Pull Down on Power-Off and USB 2.0 descriptor modes. The EEPROM should be a 16 bit wide configuration such as a MicroChip 93LC46B or equivalent capable of a 1Mb/s clock rate at VCC = 4.4v to 5.25v. The EEPROM is programmable-on board over USB using a utility available from FTDI's web site (http://www.ftdichip.com). This allows a blank part to be soldered onto the PCB and programmed as part of the manufacturing and test process.

  If no EEPROM is connected (or the EEPROM is blank), the FT232BM will use it's built-in default VID, PID Product Description and Power Descriptor Value. In this case, the device will not have a serial number as part of the USB descriptor.

4.0 Device Pin-Out

Figure 1

Pin-Out (LQFP-32 Package)

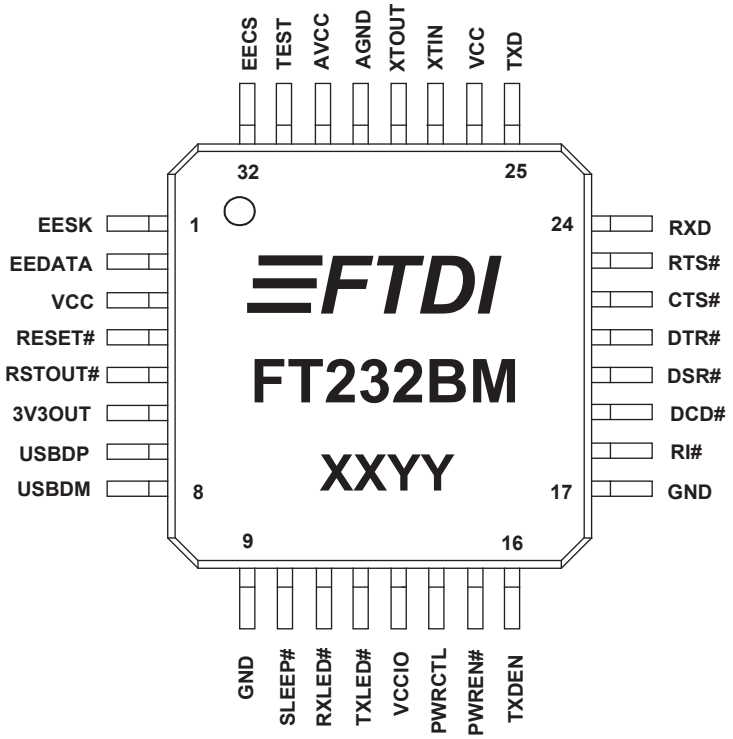
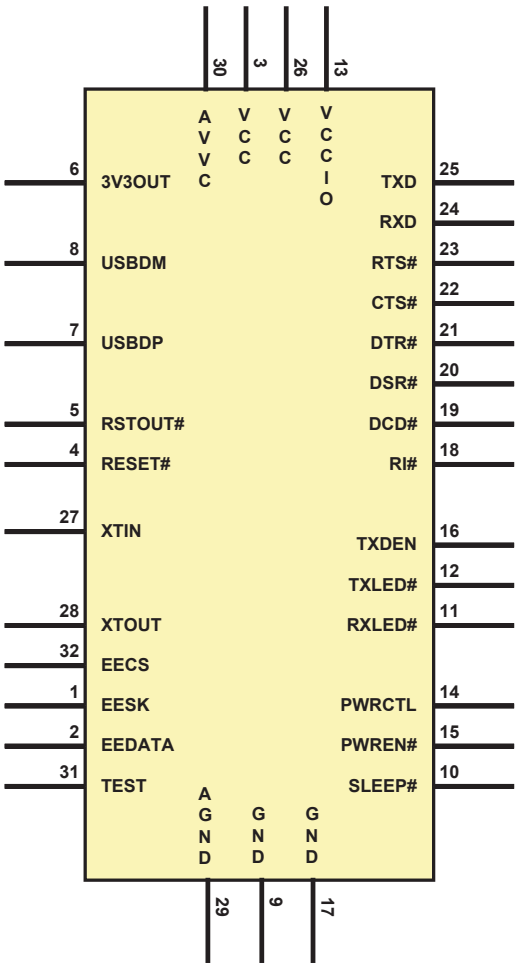


Figure 2

Pin-Out (Schematic Symbol)



4.1 Signal Descriptions

Table 1 - FT232BM - PINOUT DESCRIPTION

UART INTERFACE GROUP

Pin#	Signal	Type	Description
25	TXD	OUT	Transmit Asynchronous Data Output
24	RXD	IN	Receive Asynchronous Data Input
23	RTS#	OUT	Request To Send Control Output / Handshake signal
22	CTS#	IN	Clear To Send Control Input / Handshake signal
21	DTR#	OUT	Data Terminal Ready Control Output / Handshake signal
20	DSR#	IN	Data Set Ready Control Input / Handshake signal
19	DCD#	IN	Data Carrier Detect Control Input
18	RI#	IN	Ring Indicator Control Input. When the Remote Wakeup option is enabled in the EEPROM, taking RI# low can be used to resume the PC USB Host controller from suspend.
16	TXDEN	OUT	Enable Transmit Data for RS485

USB INTERFACE GROUP

Pin#	Signal	Type	Description
7	USBDP	I/O	USB Data Signal Plus (Requires 1.5k pull-up to 3V3OUT or RSTOUT#)
8	USBDM	I/O	USB Data Signal Minus

EEPROM INTERFACE GROUP

Pin#	Signal	Type	Description
32	EECS	I/O	EEPROM – Chip Select. For 48MHz operation pull EECS to GND using a 10k resistor. For 6MHz operation no resistor is required. Tri-State during device reset.
1	EESK	OUT	Clock signal to EEPROM. Tri-State during device reset, else drives out.
2	EEDATA	I/O	EEPROM – Data I/O Connect directly to Data-In of the EEPROM and to Data-Out of the EEPROM via a 2k2 resistor. Also pull Data-Out of the EEPROM to VCC via a 10k resistor for correct operation. Tri-State during device reset.

POWER CONTROL GROUP

Pin#	Signal	Type	Description
10	SLEEP#	OUT	Goes Low during USB Suspend Mode. Typically used to power-down an external TTL to RS232 level converter i.c. in USB -> RS232 converter designs.
15	PWREN#	OUT	Goes Low after the device is configured via USB, then high during USB suspend. Can be used to control power to external logic using a P-Channel Logic Level MOSFET switch. Enable the Interface Pull-Down Option in EEPROM when using the PWREN# pin in this way.
14	PWRCTL	IN	Bus Powered – Tie Low / Self Powered – Tie High

MISCELLANEOUS SIGNAL GROUP

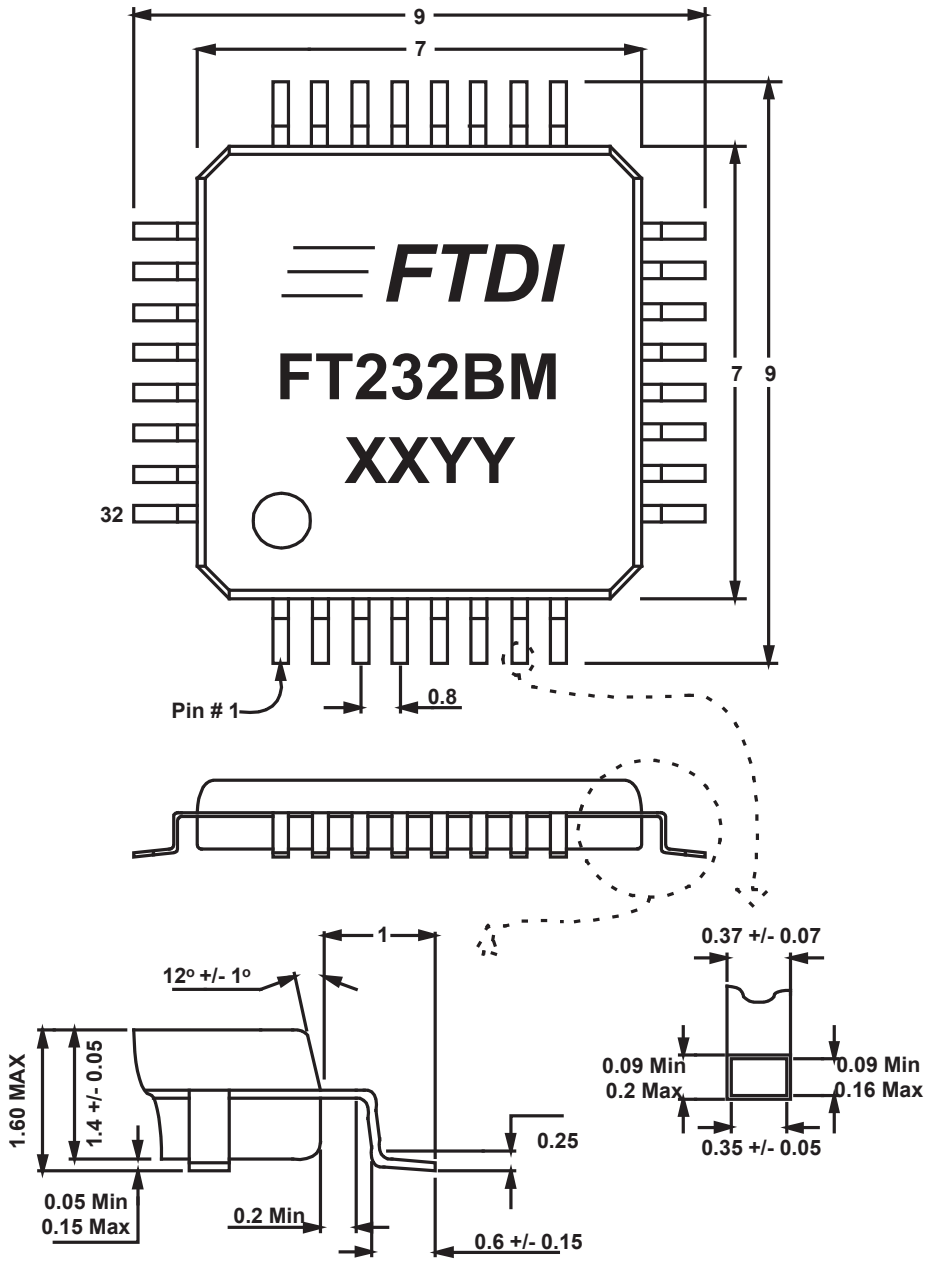
Pin#	Signal	Type	Description
4	RESET#	IN	Can be used by an external device to reset the FT232BM. If not required, tie to VCC.
5	RSTOUT#	OUT	Output of the internal Reset Generator. Stays high impedance for ~ 2ms after VCC > 3.5v and the internal clock starts up, then clamps it's output to the 3.3v output of the internal regulator. Taking RESET# low will also force RSTOUT# to go high impedance. RSTOUT# is NOT affected by a USB Bus Reset.
12	TXLED#	O.C.	LED Drive - Pulses Low when Transmitting Data via USB
11	RXLED#	O.C.	LED Drive - Pulses Low when Receiving Data via USB
27	XTIN	IN	Input to 6MHz Crystal Oscillator Cell. This pin can also be driven by an external 6MHz clock if required. Note : Switching threshold of this pin is VCC/2, so if driving from an external source, the source must be driving at 5V CMOS level or a.c. coupled to centre around VCC/2.
28	XTOUT	OUT	Output from 6MHz Crystal Oscillator Cell. XTOUT stops oscillating during USB suspend, so take care if using this signal to clock external logic.
31	TEST	IN	Puts device in i.c. test mode – must be tied to GND for normal operation.

POWER AND GND GROUP

Pin#	Signal	Type	Description
6	3V3OUT	OUT	3.3 volt Output from the integrated L.D.O. regulator This pin should be decoupled to GND using a 33nF ceramic capacitor in close proximity to the device pin. It's prime purpose is to provide the internal 3.3v supply to the USB transceiver cell and the RSTOUT# pin. A small amount of current (<= 5mA) can be drawn from this pin to power external 3.3v logic if required.
3,26	VCC	PWR	+4.4 volt to +5.25 volt VCC to the device core, LDO and and none-UART interface pins.
13	VCCIO	PWR	+3.0 volt to +5.25 volt VCC to the UART interface pins 10..12, 14..16 and 18..25. When interfacing with 3.3v external logic connect VCCIO to the 3.3v supply of the external logic, otherwise connect to VCC to drive out at 5v CMOS level.
9,17	GND	PWR	Device– Ground Supply Pins
30	AVCC	PWR	Device - Analog Power Supply for the internal x8 clock multiplier
29	AGND	PWR	Device - Analog Ground Supply for the internal x8 clock multiplier

5.0 Package Outline

Figure 3 – 32 LD LQFP Package Dimensions



The FT232BM is supplied in a 32 LD LQFP package as standard. This package has a 7mm x 7mm body (9mm x 9mm including leads) with leads on a 0.8mm pitch. An alternative 5mm x 5mm leadless chip scale package is available on special request for projects where package area is critical.

The above drawing shows the LQFP-32 package – all dimensions are in millimetres.

XXYY = Date Code (XX = 2 digit year number, YY = 2 digit week number.

6.0 Absolute Maximum Ratings

These are the absolute maximum ratings for the FT232BM device in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

- Storage Temperature ..... –65°C to + 150°C
- Ambient Temperature (Power Applied)..... 0°C to + 70°C
- VCC Supply Voltage ..... -0.5v to +6.00v
- DC Input Voltage - Inputs ..... -0.5v to VCC + 0.5v
- DC Input Voltage - High Impedance Bidirectionals ..... -0.5v to VCC + 0.5v
- DC Output Current – Outputs 24mA
- DC Output Current – Low Impedance Bidirectionals 24mA
- Power Dissipation (VCC = 5.25v) 500mW
- Electrostatic Discharge Voltage (I < 1uA) ..... +/- 2000v
- Latch Up Current (Vi < 0 or Vi > Vcc) 100mA

6.1 D.C. Characteristics

DC Characteristics (Ambient Temperature = 0 .. 70°C)

Operating Voltage and Current

Parameter	Description	Min	Typ	Max	Units	Conditions
Vcc1	VCC Operating Supply Voltage	4.4	5.0	5.25	V	
Vcc2	VCCIO Operating Supply Voltage	3.0	-	5.25	V	
Icc1	Operating Supply Current	-	25	-	mA	Normal Operation
Icc2	Operating Supply Current	-	180	200	uA	USB Suspend ** Note 1

Note 1 – Supply current excludes the 200uA nominal drawn by the external pull-up resistor on USB DP.

UART IO Pin Characteristics (VCCIO = 5.0v)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	4.4	-	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
Vin	Input Switching Threshold	1.1	1.5	1.9	V	Note 2
VHys	Input Switching Hysteresis		200		mV	

UART IO Pin Characteristics (VCCIO = 3.3v)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.7	-	3.2	V	I source = 2mA
Vol	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
Vin	Input Switching Threshold	1.0	1.4	1.8	V	Note 2
VHys	Input Switching Hysteresis		200		mV	

Note 2 – Inputs have an internal 200k pull-up resistor to VCCIO.

XTIN / XTOUT Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	4.0	-	5.0	V	Fosc = 6MHz
Vol	Output Voltage Low	0.1	-	1.0	V	Fosc = 6MHz
Vin	Input Switching Threshold	1.8	2.5	3.2	V	

RESET#, TEST, EECS, EESK, EEDATA, IO Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	4.4	-	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
Vin	Input Switching Threshold	1.1	1.5	1.9	V	Note 3
VHys	Input Switching Hysteresis		200		mV	

Note 3 – EECS and EEDATA pins have an internal 200k pull-up resistor to VCC

RSTOUT Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	3.0	-	3.6	V	I source = 2mA
Iol	Leakage Current Tri-State	-	-	5	uA	

USB IO Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
UVoh	IO Pins Static Output (High)	2.8		3.6v	V	RI = 1k5 to 3V3Out (D+) RI = 15k to GND (D-)
UVol	IO Pins Static Output (Low)	0		0.3	V	RI = 1k5 to 3V3Out (D+) RI = 15k to GND (D-)
UVse	Single Ended Rx Threshold	0.8		2.0	V	
UCom	Differential Common Mode	0.8		2.5	V	
UVDif	Differential Input Sensitivity	0.2			V	
UDrvZ	Driver Output Impedance	29		44	ohm	Note 4

Note 4 – Driver Output Impedance includes the external 27R series resistors on USBDP and USBDM pins.

7.0 Device Configuration Examples

7.1 Oscillator Configurations

Figure 4
3 Pin Ceramic Resonator Configuration

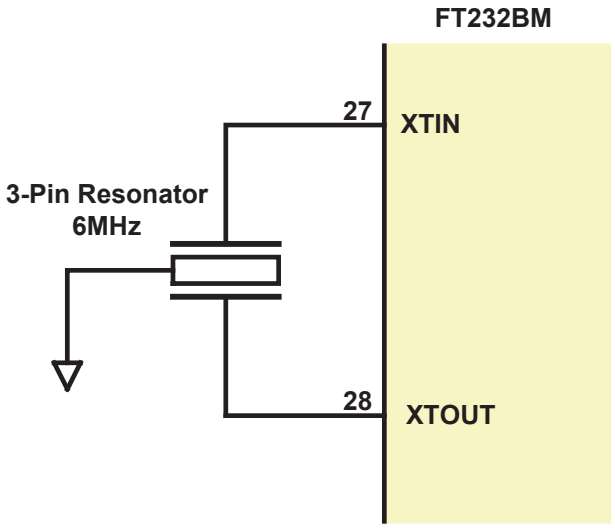


Figure 4 illustrates how to use the FT232BM with a 3-Pin Ceramic Resonator such as Murata Part # CSTLS6M00G53 or equivalent. 3-Pin resonators have the load capacitors built into the resonator so no external loading capacitors are required. This makes for an economical configuration. Though the typical accuracy of such a resonator is +/- 0.5% and is technically out-with the USB specification, it has been calculated that using such a device will work satisfactorily in practice with the FT232BM design.

Figure 5
Crystal or 2-Pin Ceramic Resonator Configuration

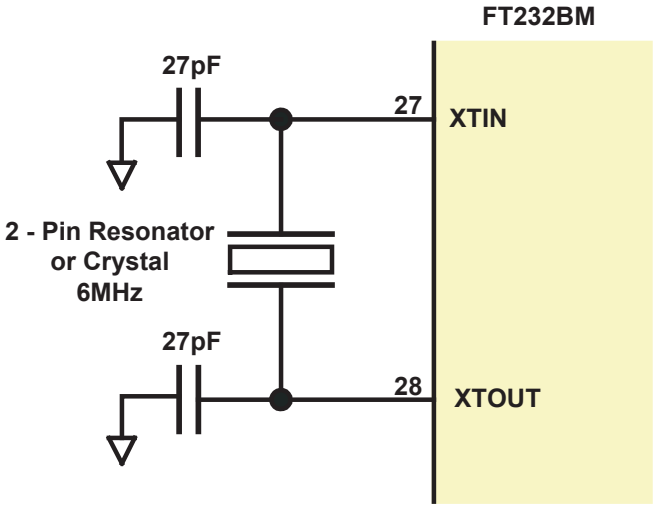


Figure 5 illustrates how to use the FT232BM with a 6MHz Crystal or 2-Pin Ceramic Resonator. In this case, these devices do not have in-built loading capacitors so these have to be added between XTIN, XTOUT and GND as shown. A value of 27pF is shown as the capacitor in the example – this will be good for many crystals and some resonators but do select the value based on the manufacturers recommendations wherever possible. If using a crystal, use a parallel cut type. If using a resonator, see the previous note on frequency accuracy.

7.2 EEPROM Configuration

Figure 6
EEProm Configuration

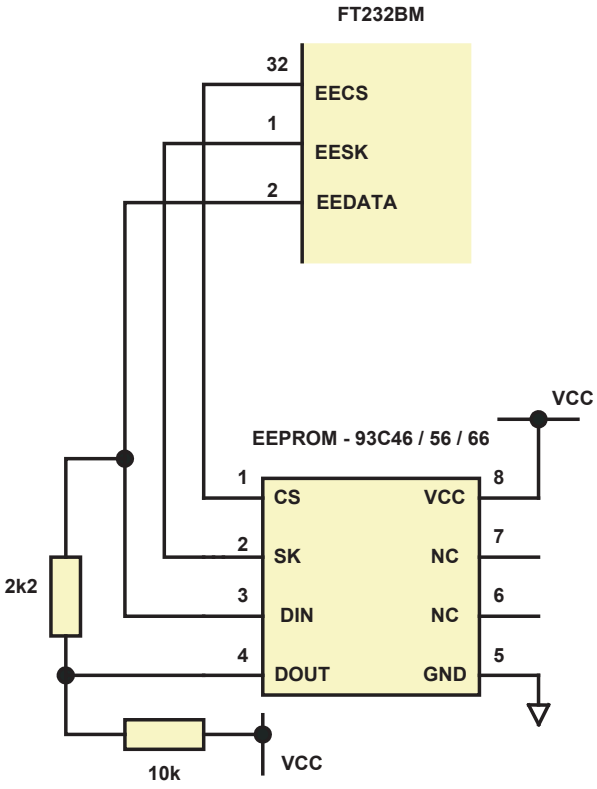


Figure 6 illustrates how to connect the FT232BM to the 93C46 (93C56 or 93C66) EEPROM. EECS (pin 32) is directly connected to the chip select (CS) pin of the EEPROM. EESK (pin 1) is directly connected to the clock (SK) pin of the EEPROM. EEDATA (pin 2) is directly connected to the Data In (Din) pin of the EEPROM. There is a potential condition whereby both the Data Output (Dout) of the EEPROM can drive out at the same time as the EEDATA pin of the FT232BM. To prevent potential data clash in this situation, the Dout of the EEPROM is connected to EEDATA of the FT232BM via a 2k2 resistor.

Following a power-on reset or a USB reset, the FT232BM will scan the EEPROM to find out a) if an EEPROM is attached to the Device and b) if the data in the device is valid. If both of these are the case, then the FT232BM will use the data in the EEPROM, otherwise it will use it's built-in default values. When a valid command is issued to the EEPROM from the FT232BM, the EEPROM will acknowledge the command by pulling it's Dout pin low. In order to check for this condition, it is necessary to pull Dout high using a 10k resistor. If the command acknowledge doesn't happen then EEDATA will be pulled high by the 10k resistor during this part of the cycle and the device will detect an invalid command or no EEPROM present.

There are two varieties of these EEPROMs on the market – one is configured as being 16 bits wide, the other is configured as being 8 bits wide. These are available from many sources such as Microchip, ST, SIS etc. The FT232BM requires EEPROMs with a 16-bit wide configuration such as the Microchip 93LC46B device. The EEPROM must be capable of reading data at a 1Mb clock rate at a supply voltage of 4.4v to 5.25v. Most available parts are capable of this.

Check the manufacturers data sheet to find out how to connect pins 6 and 7 of the EEPROM. Some devices specify these as no-connect, others use them for selecting 8 / 16 bit mode or for test functions. Some other parts have their pinout rotated by 90° so please select the required part and it's options carefully.

It is possible to "share" the EEPROM between the FT232BM and another external device such as an MCU. However, this can only be done when the FT232BM is in it's reset condition as it tri-states it's EEPROM interface at that time. A typical configuration would use four bit's of an MCU IO Port. One bit would be used to hold the FT232BM reset (using RESET#) on power-up, the other three would connect to the EECS, EESK and EEDATA pins of the FT232BM in order to read / write data to the EEPROM at this time. Once the MCU has read / written the EEPROM, it would take RESET# high to allow the FT232BM to configure itself and enumerate over USB.

7.3 USB Bus Powered and Self Powered Configuration

Figure 7
USB Bus Powered Configuration

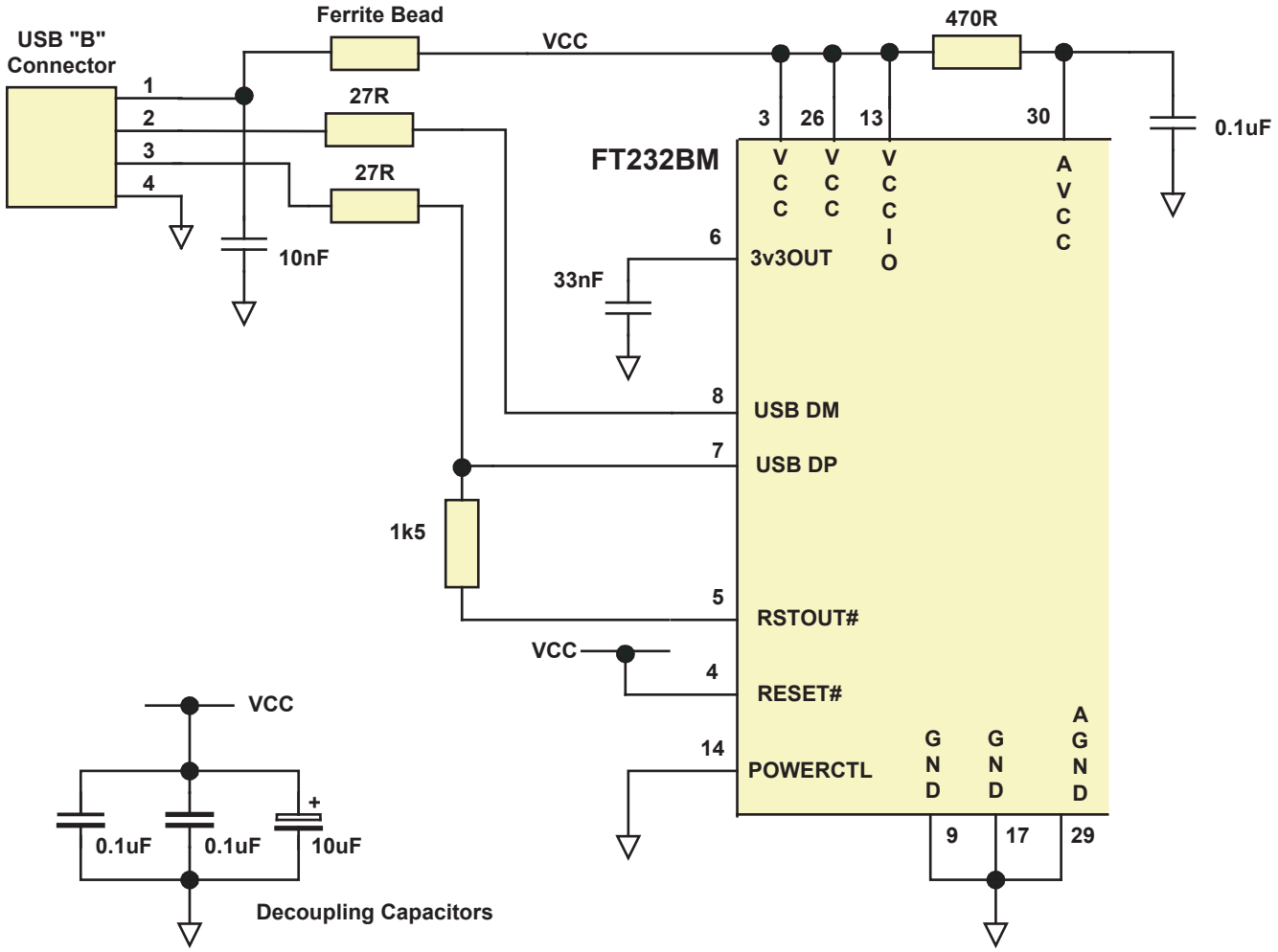


Figure 7 illustrates a typical USB bus powered configuration. A USB Bus Powered device gets its power from the USB bus. Basic rules for USB Bus power devices are as follows –

a) On plug-in, the device must draw no more than 100mA
b) On USB Suspend the device must draw no more than 500uA.
c) A Bus Powered High Power Device (one that draws more than 100mA) should use the PWREN# pin to keep the current below 100mA on plug-in and 500uA on USB suspend.
d) A device that consumes more than 100mA can not be plugged into a USB Bus Powered Hub
e) No device can draw more that 500mA from the USB Bus.

POWERCTL (pin 14) is pulled low to tell the device to use a USB Bus Power descriptor. The power descriptor in the EEPROM should be programmed to match the current draw of the device.

A Ferrite Bead is connected in series with USB power to prevent noise from the device and associated circuitry (EMI) being radiated down the USB cable to the Host. The value of the Ferrite Bead depends on the total current required by the circuit – a suitable range of Ferrite Beads is available from Steward (www.steward.com) for example Steward Part # **MI0805K400R-00 also available as DigiKey Part #** 240-1035-1.

Figure 8
USB Self Powered Configuration (1)

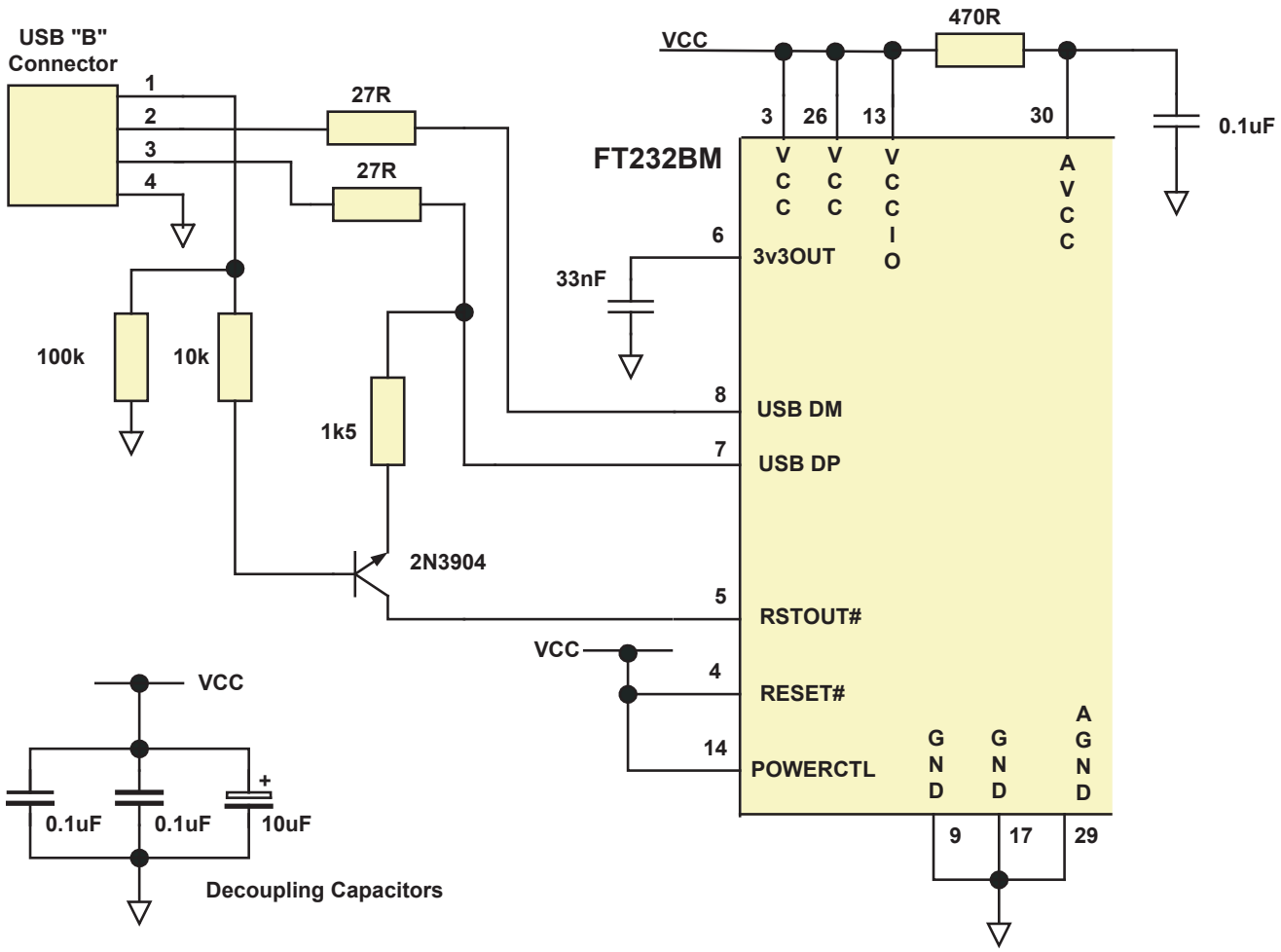


Figure 8 illustrates a typical USB self powered configuration. A USB Self Powered device gets its power from its own POWER SUPPLY and does not draw current from the USB bus. Basic rules for USB Self power devices are as follows –

a) A Self-Powered device should not force current down the USB bus when the USB Host or Hub Controller is powered down.

b) A Self Powered Device can take as much current as it likes during normal operation and USB suspend as it has its own POWER SUPPLY.

c) A Self Powered Device can be used with any USB Host and both Bus and Self Powered USB Hubs

POWERCTL (pin 14) is pulled high to tell the device to use a USB Bus Power descriptor. The power descriptor in the EEPROM should be programmed to a value of zero.

To meet requirement a), the 1k5 pull-up circuit on USB DP has to be modified to prevent the device forcing current into the USB DP line via the 1k5 pull-up when the host or hub is powered down. Failure to do this may cause some USB host or hub controllers to power up erratically. A NPN small signal transistor (2N3906) is used to sense the power on the USB bus. It is connected as an emitter-follower circuit so that when there is power on the USB bus the transistor will saturate and pull the 1k5 resistor to the voltage of RSTOUT#. When the USB power is off, the transistor will turn off thus preventing current flow into the USB DP line.

Figure 9
USB Self Powered Configuration (2)

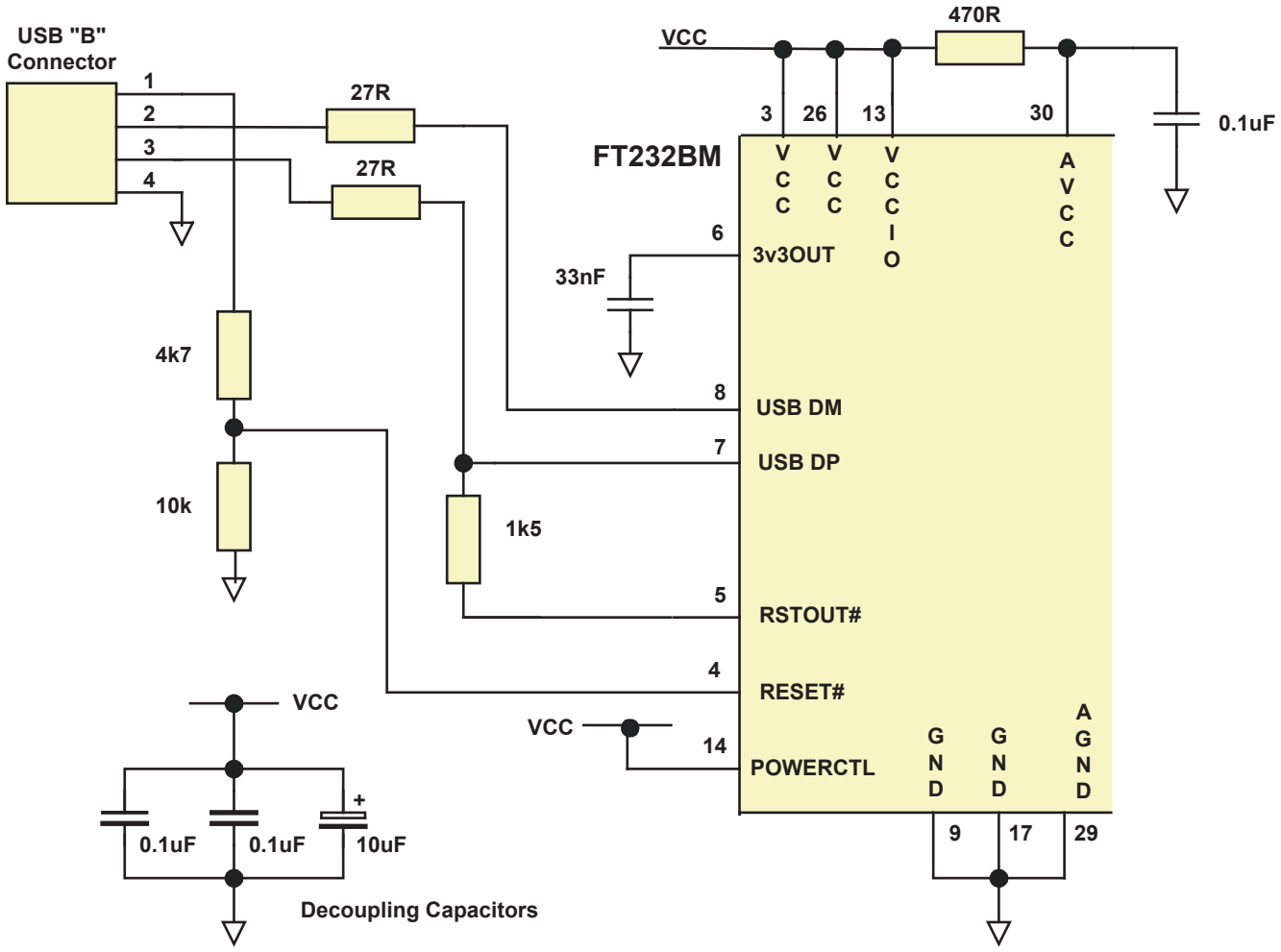


Figure 9 illustrates a variant of the circuit shown in Figure 8. This time, the 1k5 pull-up resistor on USB DP is connected to RSTOUT# as per the bus-power circuit. However, the USB Bus Power is used to control the RESET# Pin of the FT232BM device. When the USB Host or Hub power is off, RESET# will go low and the device will be held in reset. As RESET# is low, RSTOUT# will also be low, so no current will be forced down USB DP via the 1k5 pull-up resistor.

Note : When the FT232B is in reset, the UART interface pins all go tri-state. These pins have internal 200k pull-up resistors to VCC-IO so they will gently pull high unless driven by some external logic.

Which of the two configurations to use depends on the nature of the peripheral design. With the configuration of Figure 8, the FT232BM is “live” – when power to the USB port is shut off, there will be no activity on the USB bus and the device will enter low power sleep mode after a few milliseconds. In this configuration, the RESET# pin is still available if required.

In the Figure 9 configuration, the FT232BM is held in reset when the USB power is off. In reset, the FT232BM 6MHz oscillator will still be running and the device will not be in low power mode.

7.4 UART Interface Configuration

Figure 10
USB => RS232 Converter Configuration

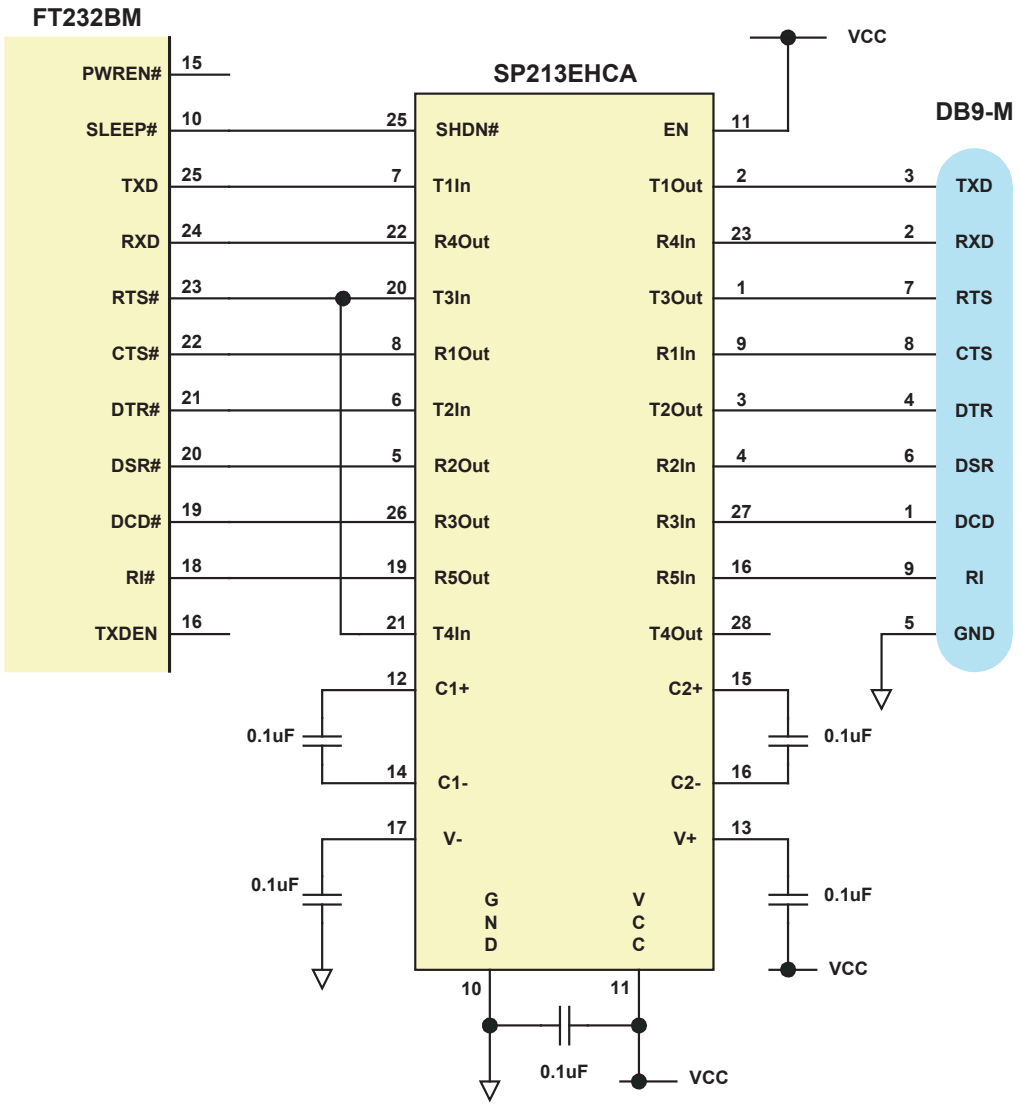


Figure 10 illustrates how to connect the UART interface of the FT232BM to a TTL – RS232 Level Converter I.C. to make a USB -> RS232 converter using the popular “213” series of TTL to RS232 level converters. These devices have 4 transmitters and 5 receivers in a 28 LD SSOP package and feature an in-built 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 the example 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. 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#.

Figure 11
USB => RS422 Converter Configuration

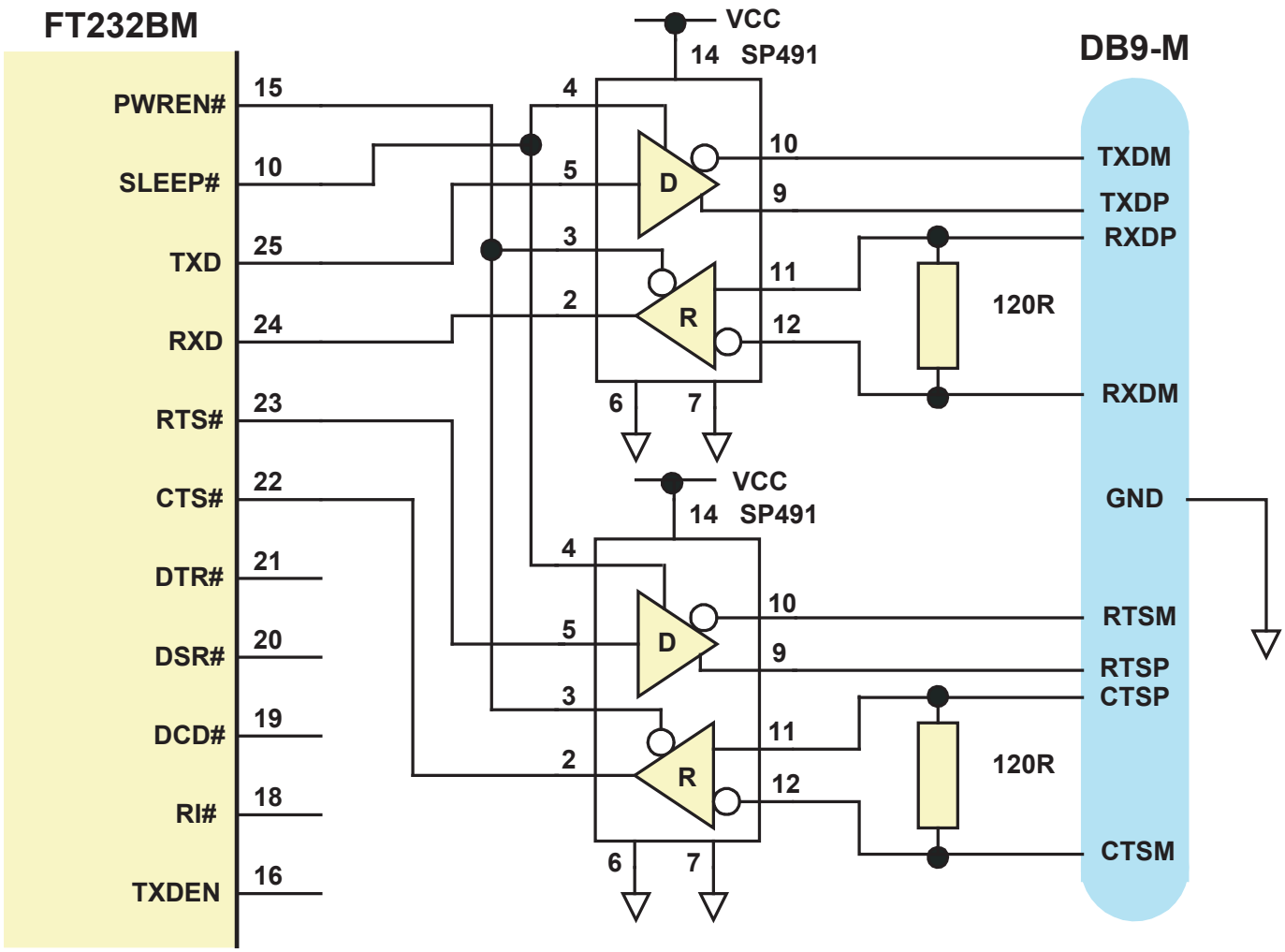


Figure 11 illustrates how to connect the UART interface of the FT232BM to a TTL – RS422 Level Converter I.C. to make a USB -> RS422 converter. There are many such level converter devices available – this example uses Sipex SP491 devices which have enables on both the transmitter and receiver. Because the transmitter enable is active high, it is connected to the SLEEP# pin. The receiver enable is active low and is connected to the PWREN# pin. This ensures that both the transmitters and receivers are enabled when the device is active, and disabled when the device is in USB suspend mode. If the design is USB BUS powered, it may be necessary to use a P-Channel logic level MOSFET (controlled by PWREN#) in the VCC line of the SP491 devices to ensure that the USB standby current of 500uA is met.

The SP491 is good for sending and receiving data at a rate of up to 5M Baud – in this case the maximum rate is limited to 3M Baud by the FT232BM.

Figure 12
USB => RS485 Converter Configuration

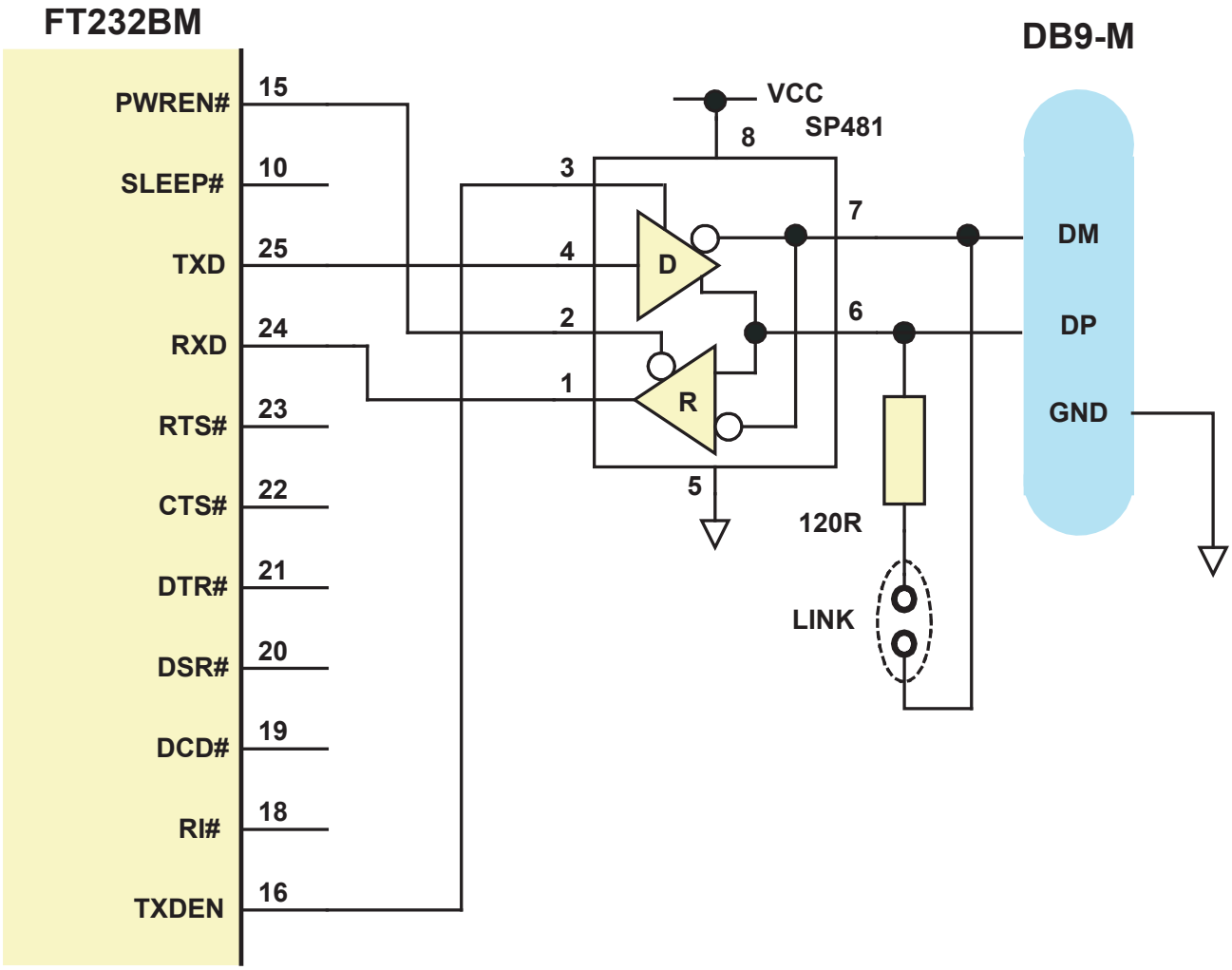


Figure 12 illustrates how to connect the UART interface of the FT232BM to a TTL – RS485 Level Converter I.C. to make a USB => RS485 converter. This example uses the Sipex SP491 device but there are similar parts available from Maxim and Analog Devices amongst others. The SP491 is a RS485 device in a compact 8 pin SOP package. It has separate enables on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. The TXDEN pin on the FT232BM is provided for exactly that purpose and so the transmitter enable is wired to TXDEN. The receiver enable is active low, so it is wired to the PWREN# pin to disable the receiver when in USB suspend mode.

RS485 is a multi-drop network – i.e. many devices can communicate with each other over a single two wire cable connection. The RS485 cable requires to be terminated at each end of the cable. A link is provided to allow the cable to be terminated if the device is physically positioned at either end of the cable.

In this example the data transmitted by the FT232BM is also received by the device that is transmitting. This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT232BM it is possible to do this entirely in hardware – simply modify the schematic so that RXD of the FT232BM is the logical OR of the SP481 receiver output with TXDEN using an HC32 or similar logic gate.

7.5 LED Interface

Figure 13
Dual LED Configuration

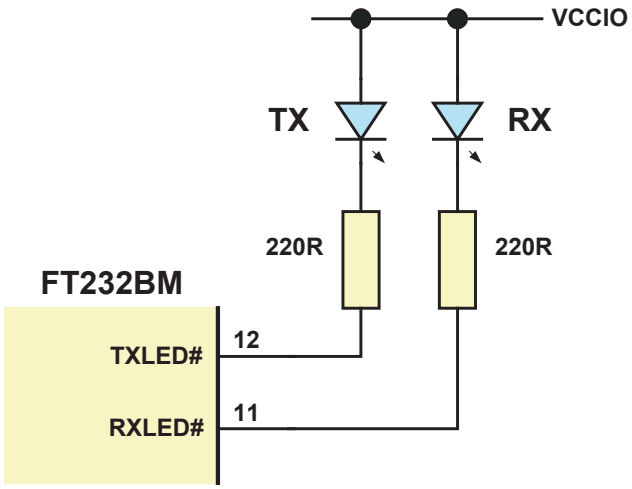
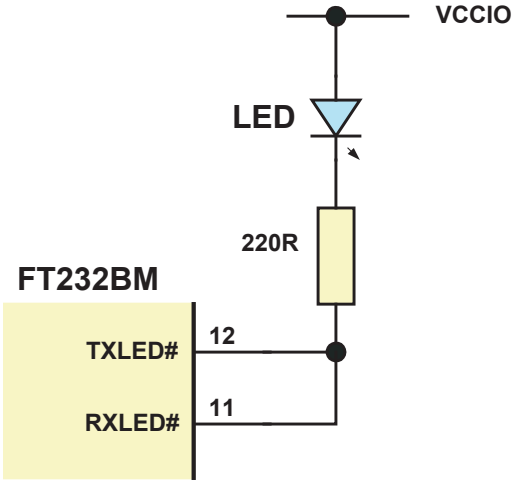


Figure 14
Single LED Configuration



The FT232BM has two IO pins dedicated to controlling LED status indicators, one for transmitted data the other for received data. When data is being transmitted / received the respective pins drive from tri-state to low in order to provide indication on the LEDs of data transfer. A digital one-shot timer is used so that even a small percentage of data transfer is visible to the end user. Figure 13 shows a configuration using two individual LED's – one for transmitted data the other for received data. In Figure 14, the transmit and receive LED indicators are wire-or'd together to give a single LED indicator which indicates any transmit or receive data activity.
Another possibility (not shown here) is to use a 3 pin common anode tri-color LED based on the circuit in Figure 13 to have a single LED that can display activity in a variety of colors depending on the ratio of transmit activity compared to receive activity.

7.6 Interfacing to 3.3v Logic

Figure 15
Bus Powered Circuit with 3.3v logic drive / supply voltage

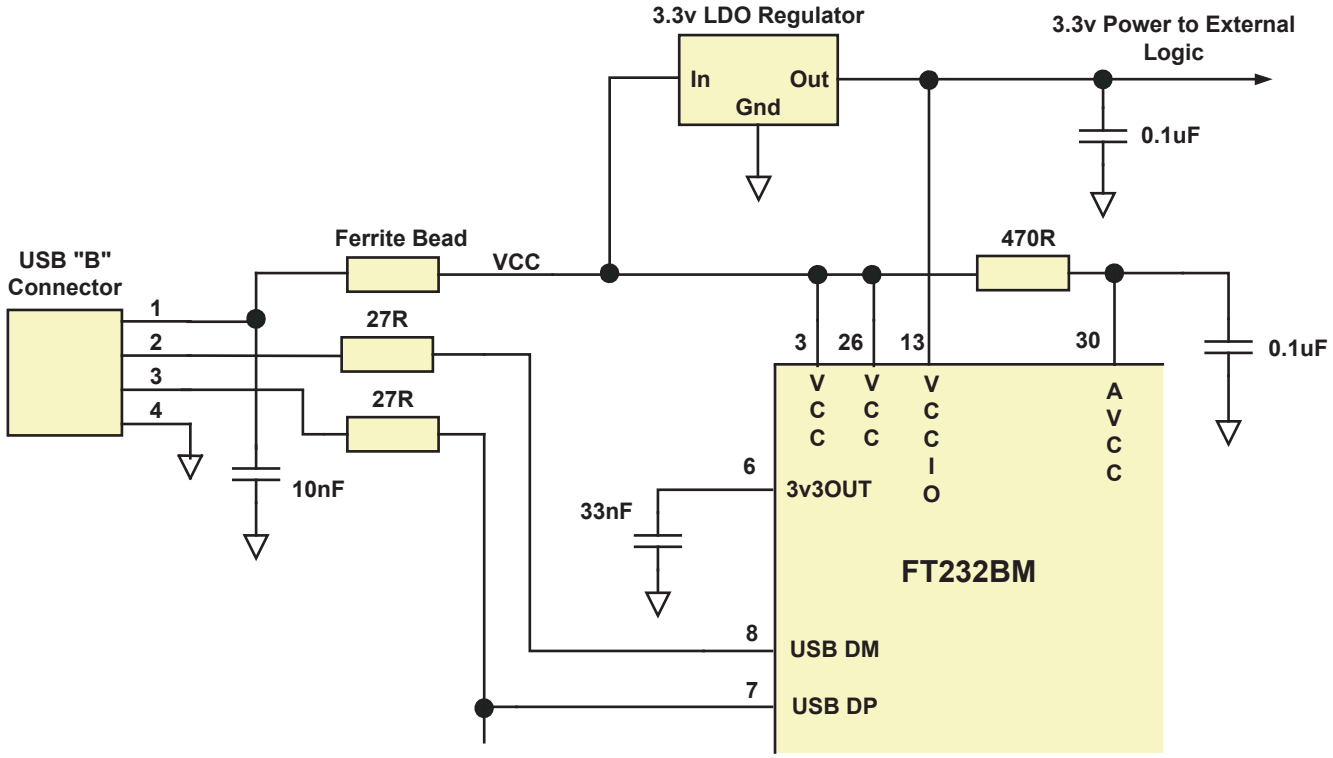


Figure 15 shows how to configure the FT232BM to interface with a 3.3v logic device. In this example, a discrete 3.3v regulator is used to supply the 3.3v logic from the USB supply. VCCIO is connected to the output of the 3.3v regulator, which in turn will cause the UART interface IO pins to drive out at 3.3v level. For USB bus powered circuits some considerations have to be taken into account when selecting the regulator –

a) The regulator must be capable of sustaining its output voltage with an input voltage of 4.4 volts. A Low Drop Out (LDO) regulator must be selected.

b) The quiescent current of the regulator must be low in order to meet the USB suspend total current requirement of <= 500uA during USB suspend.

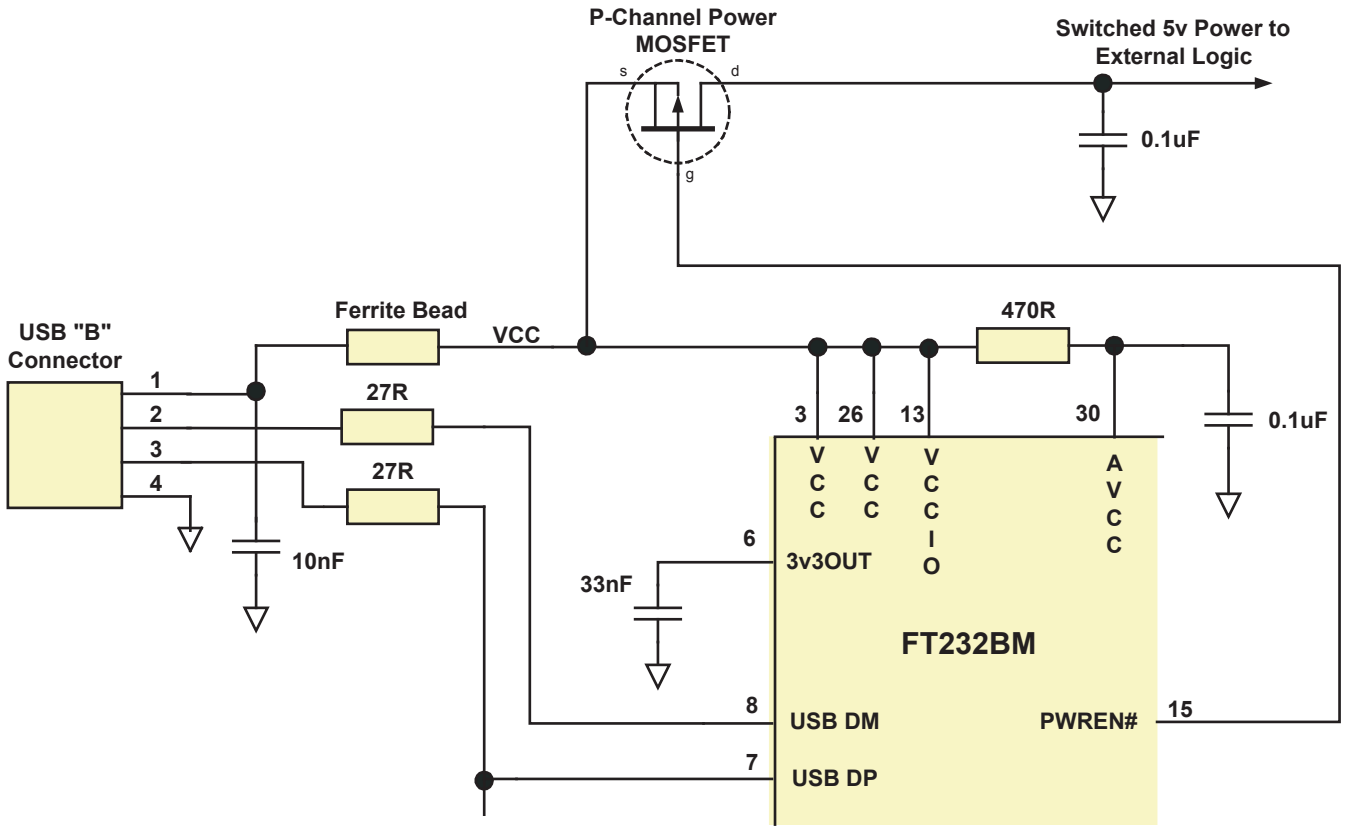
An example of a regulator family that meets these requirements is the MicroChip (Telcom) TC55 Series. These devices can supply up to 250mA current and have a quiescent current of under 1uA.

When using the FT232BM in a self powered USB design, simply connect VCCIO to the 3.3v supply rail of the external 3.3v logic. Suspend current is not a consideration for self powered designs.

In some cases, where only a small amount of current is required (< 5mA) , it may be possible to use the in-built regulator of the FT232BM to supply the 3.3v without any other components being required. In this case, connect VCCIO to the 3v3OUT pin of the FT232BM.

7.7 Power Switching

Figure 16
Bus Powered Circuit (<= 100mA) with Power Control



USB Bus powered circuits need to be able to power down in USB suspend mode in order to meet the <= 500uA total suspend current requirement (including external logic). Some external logic can power itself down into a low current state by monitoring the POWEREN# pin. For external logic that cannot power itself down in that way, the FT232BM provides a simple but effective way of turning off power to external circuitry during USB suspend.

Figure 16 shows how to use a discrete P-Channel Logic Level MOSFET to control the power to external logic circuits. A suitable device could be a Fairchild NDT456P or equivalent. This configuration is suitable for powering external logic where the normal supply current is <= 100mA and the logic to be controlled does not generate an appreciable current surge at power-up. For power switching external logic that takes over 100mA or generates a current surge on power-up we recommend that a dedicated power switch i.c with inbuilt “soft-start” is used instead of a MOSFET. A suitable power switch i.c. for such an application would be a Micrel (www.micrel.com) MIC2025-2BM or equivalent.

Please note the following points in connection with power controlled designs –

a) The logic to be controlled must have it’s own reset circuitry so that it will automatically reset itself when power is re-applied on coming out of suspend.
b) Set the soft pull-down option bit in the FT232BM EEPROM.
c) For 3.3v power controlled circuits VCCIO must not be powered down with the external circuitry (PWREN# gets it’s VCC supply from VCCIO). Either connect the power switch between the output of the 3.3v regulator and the external 3.3v logic OR if appropriate power VCCIO from the 3v3OUT pin of the FT232BM.

8.0 Document Revision History

DS232B Version 1.0 – Initial document created 30 April 2002.
DS232B Version 1.1 – Updated 04 August 2002

- RESET# Pin description corrected (RESET# does not have an internal 200k pull-up to VCC as previously stated).
- Figure 2 pin-out corrected (EECS = Pin 32).

9.0 Disclaimer

10.0 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.