Future Technology Devices Intl Ltd.

Application Note

Debugging FT232BM and FT245BM based designs

Created by Future Technology Devices Intl. (FTDI) Ltd.

Table of Contents

1 Introduction

1.1 Background

The Universal Serial Bus may be new to some developers. The information contained in this document is provided to help debug designs which use FTDI's FT232BM or FT245BM Integrated Circuit devices. Oscilloscope traces are provided to help with this.

2 Clock Circuit

2.1 6 MHz Clock

A **6 MHz crystal or ceramic resonator** can be used with FT232BM / FT245BM chips on the XTIN and XTOUT pins. Figure 1 shows the output of pin 28, XTOUT. This shows what the 6 MHz clock should look like when oscillating normally. In this mode the pin EECS (pin 32) does not need to be pulled high as it is pulled high via an internal 200K resistor.

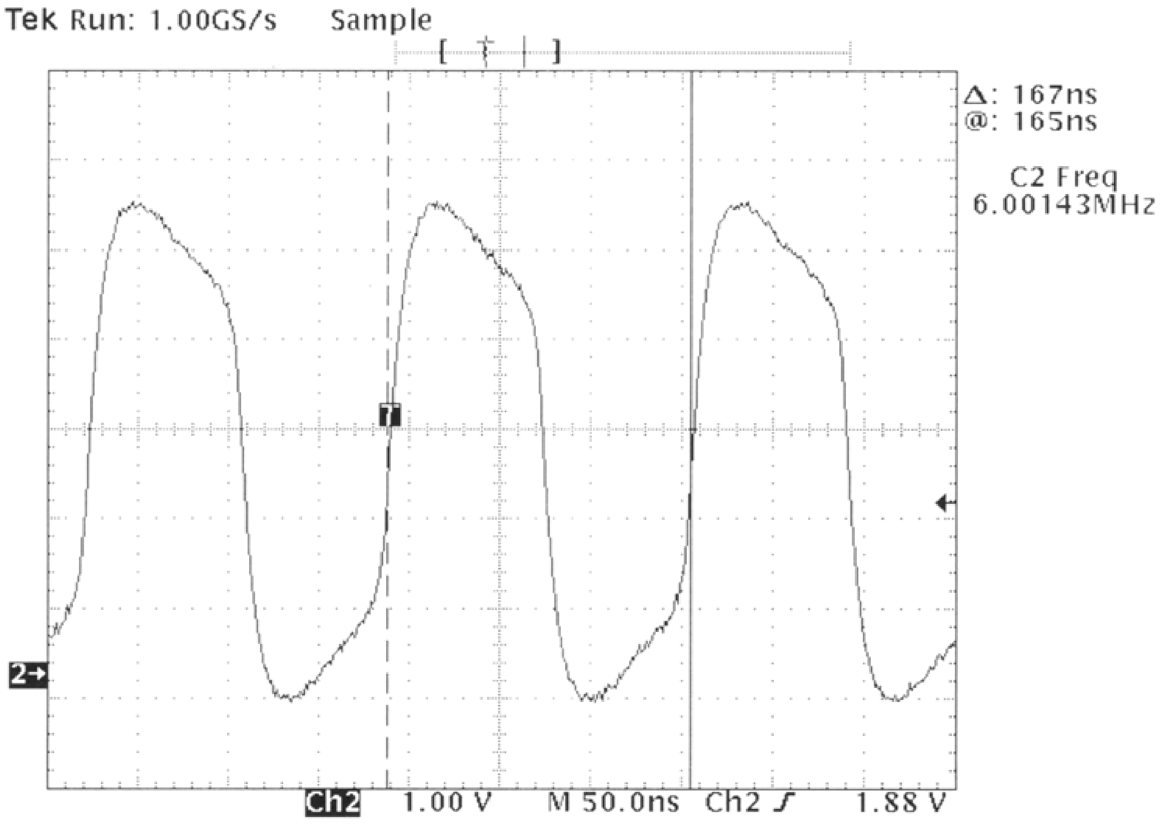


Figure 1. - 6MHz Clock signal

2.2 Debugging Clock Problems

2.2.1 Reasons For No Oscillation

1. **Lack of voltage** - For bus powered designs ensure that USB cable is plugged in and 5 Volts is seen at the chip. For self powered designs ensure that 5 Volts is applied to the FT232BM / FT245BM chip.

2. **Bad Communication / Enumeration** - The chip will stop the oscillator if the device is put into suspend by USB host. The system does this by stopping the Start Of Frame (SOF) packets which are normally sent on the USB bus every 1 millisecond. To check that the oscillator circuit is working correctly pin 4, RESET# should be held at GND (0V). This will stop the chip going into suspend so that the circuit can be looked at with an oscilloscope.

3. **Crystal is not Oscillating** - If a crystal (or resonator) without internal loading capacitors is being used, they should be fitted. Consult the manufacturers specification for a suitable capacitor value.

2.2.2 Possible Reasons For a Device Oscillating, But Not Working

1. Pin 4, RESET# being held low or floating. While RESET# is held low, the RSTOUT# signal will drive low to prevent the 1.5K ohm pull up resistor on USBDP from going high. This 1.5 K ohm pull up resistor on USBDP to the 3.3 Volts on RSTOUT# is used by the system to detect that a device is present. The system can detect that a high speed device has been connected by looking for USBDP being pulled high (USBDM is pulled high for a low speed device). If RESET# is tied to VCC then RSTOUT# will go high when the oscillator has become stable. You can tell when the chip has come out of reset as RTSOUT# will go high (3.0 - 3.6V). The chip will continue to stay in reset for 5.6 milliseconds after RSTOUT# goes high. This is shown in Figure 2.

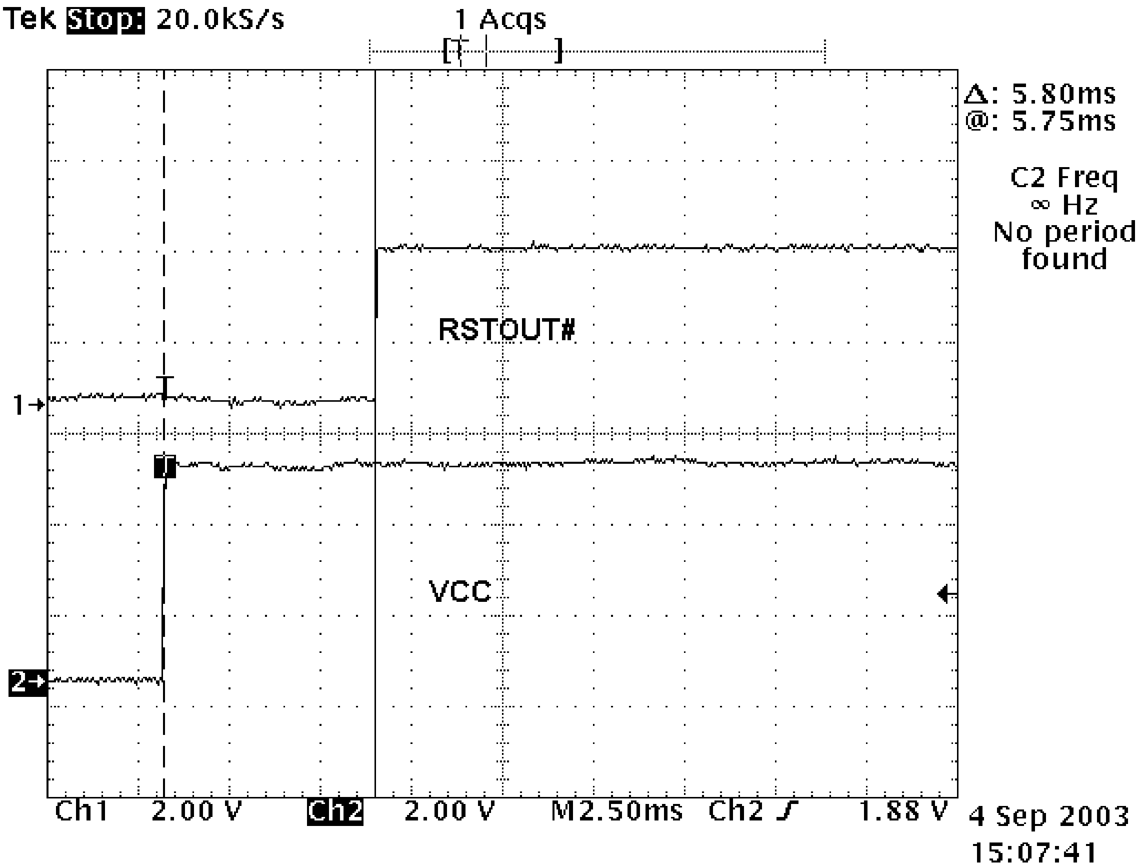


Figure 2. - RSTOUT# will go high 5.6 milliseconds after Vcc is applied.

After the 5.6 millisecond delay time the chip will start to try talk to the external EEPROM. This can be seen by looking at the signal on EESK, pin 1. See Figures 3 and 4, below.

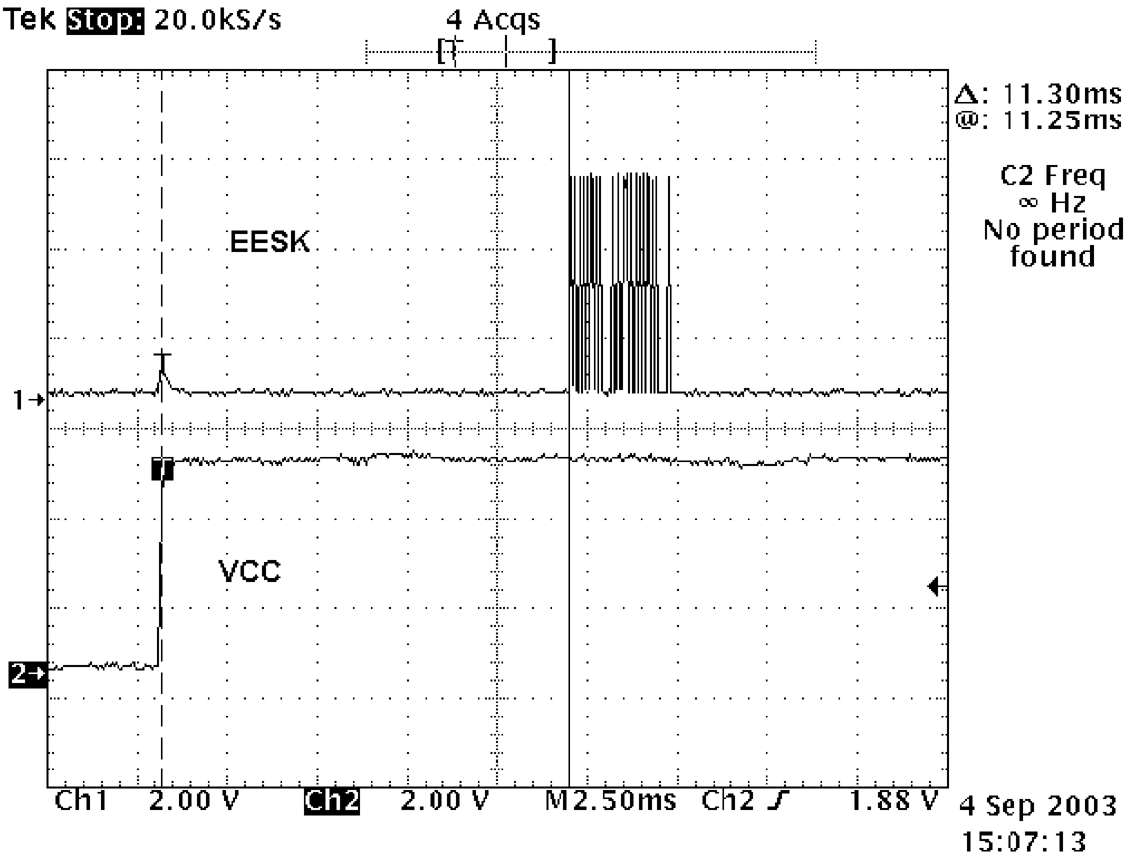


Figure 3. - EESK can be seen talking to the EEPROM after the chip comes out of reset.

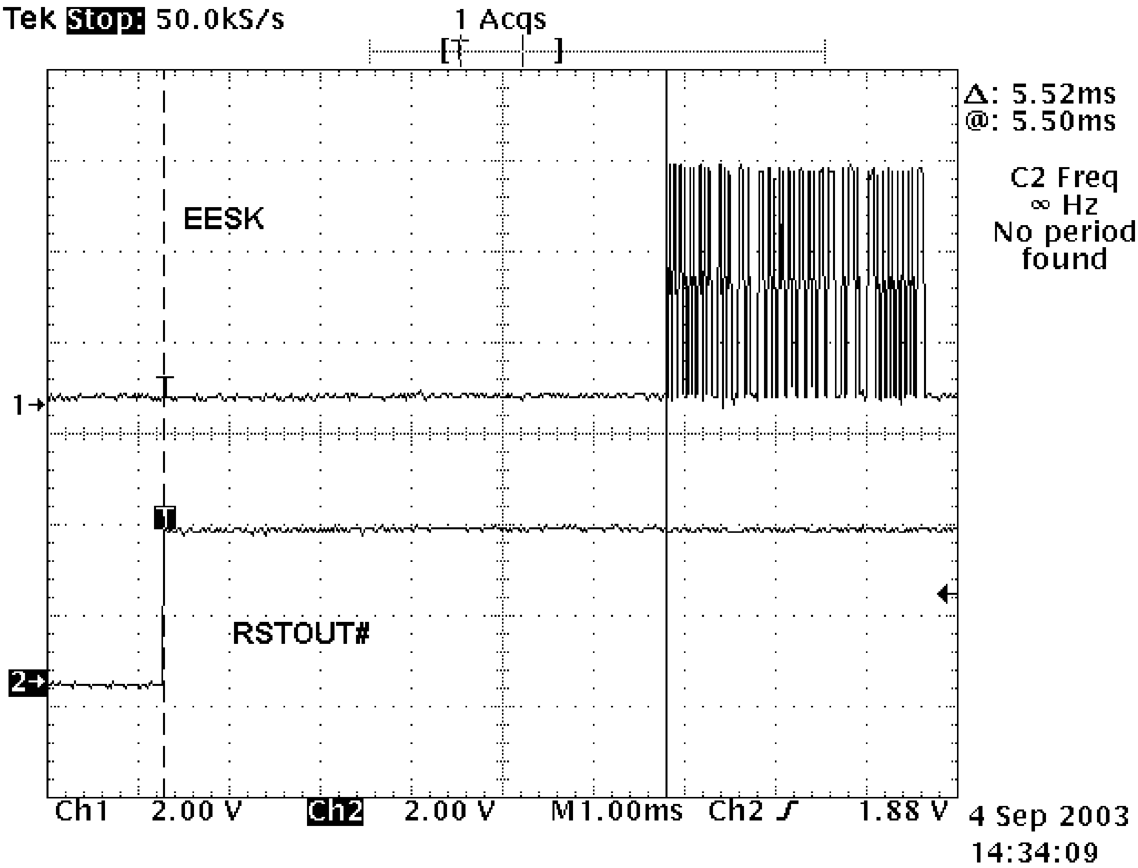


Figure 4. - EESK can be seen talking to the EEPROM after the chip comes out of reset.

2. Chip responds to first SETUP packet but always NAKs the response. - This is generally because there is a fault in the EEPROM interface. Usually the 10 K ohm pull up resistor on EEDATA (pin 2) is missing, or EEDATA is held at GND. This can be seen by continuous clocking of the EESK pin.

(NAK - USB handshake packet indicating a negative acknowledgement)

3. APLL bypass set to wrong value for speed. - In order to use the chip with a 6 MHz resonator or crystal pin 32 EECS should be high.

First generation AM devices required a 100K ohm pull up resistor to be fitted to EECS. Second generation BM chips have incorporate this pull up resistor onto the chip. The chip's APLL can be turned off by fitting a 100 K ohm pull down resistor on EECS. This is to allow BM chips to be used with a 48 MHz crystal, instead of a 6 MHz resonator / crystal.

To check for the correct speed, measure the clock period on the EESK line. EESK is used to interrogate the EEPROM when the device comes out of reset. It should have a period of just less than 1.6 microseconds. See Figure 5 in the next section.

2.2.3 Checking the Internal Chip Frequency

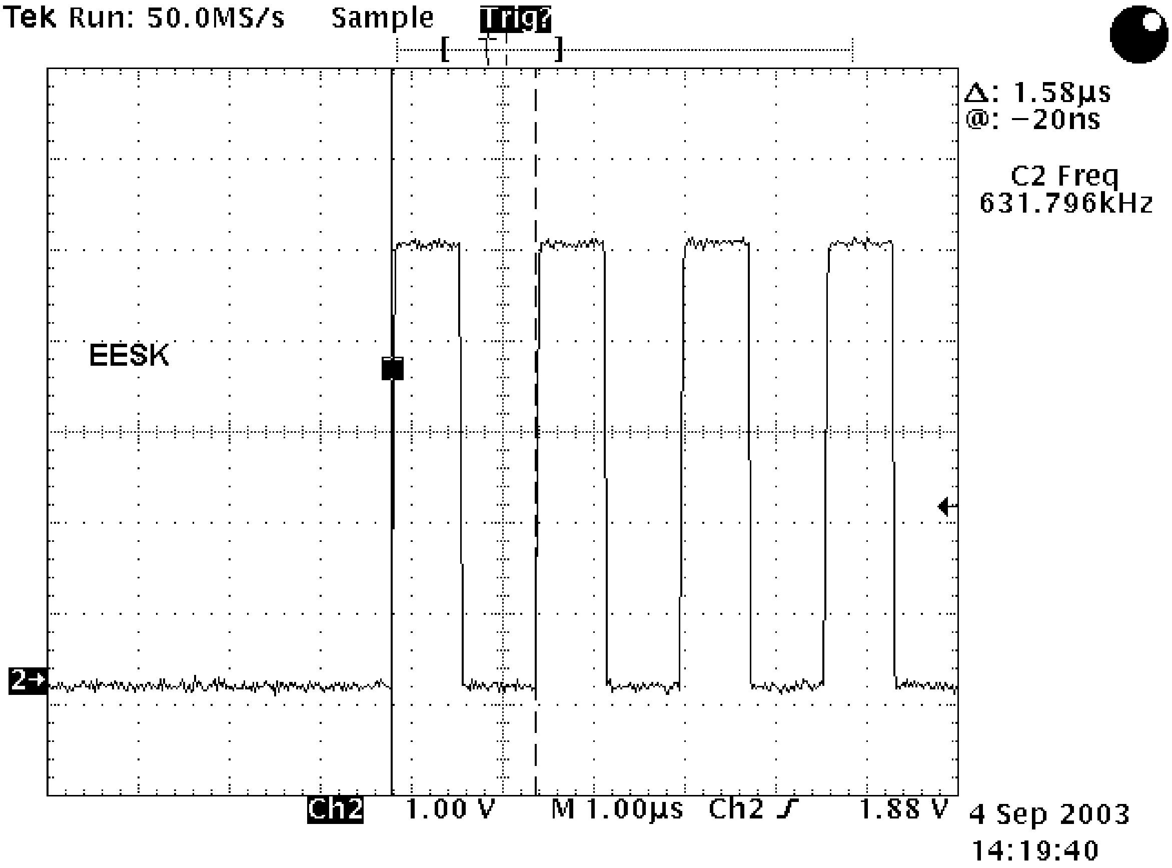


Figure 5. - The clock signal on EESK should have a period of 1.6 microseconds

A combination of faults has been seen where an FT232BM device appeared to work properly, but the baud rate selection was out by a factor of 8. This came about by the wrong frequency mode being selected by EECS, and the D+ and D- pins swapped around. The board was fitted with a 6MHz crystal but was set to bypass the 8 times internal clock multiplier APLL. This made the chip run internally at 6 instead of 48 MHz. The D+ and D- lines were swapped at the connector which caused the host PC's D- line to be pulled up instead of the D+ line. This made the host PC think that a low speed device was present. The 6 MHz frequency is the 4 times over-sampling frequency for the internal DPLL that would be used for a normal Low Speed device. Hence everything appeared to be correct except that the device baud rate was an 8th of the rate set. Either fault on its own would have caused the device not to work. When debugging, a good sanity check is to look for the above waveform on the EESK pin and check that the period is approximately 1.6 microseconds.

3 USB Signals

3.1 USBDP and USBDM

The two USB signal lines **USBDP** and **USBDM** (or D+ and D-), should not have any ferrite beads or inductors fitted in series as this will make the bus ring and oscillate. They should only have the 27 ohm series resistors in their path. These resistors should be located close to the FT232BM / FT245BM chip. Try to keep the lengths of the USBDP and USBDM tracks the same (within reason).

A SOF (Start of Frame) packet as seen at the USB connector. The top signal is D- and the bottom signal is D+.

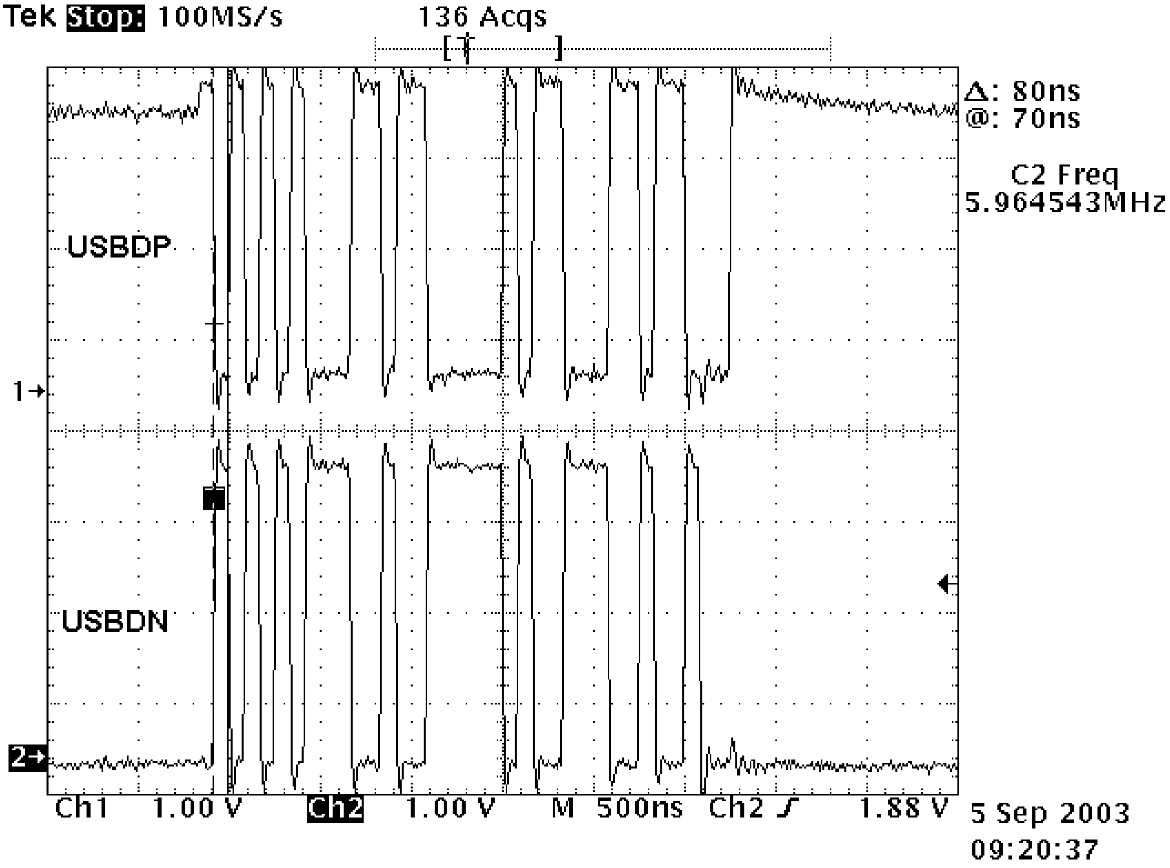


Figure 6. - USBDP and USBDM signals.

3.2 Vusb, USB Signal Ground and USB Cable Shield

Vusb - 5 Volt supply which can be used to supply bus powered designs. Up to 100 mA can be sourced from USB in low power designs, and up to 500 mA can be sourced in High power designs. High power designs should follow the advice given in the device data sheet and Designers Guide. The amount current to be sourced by the device should be written into the external EEPROM.

It is recommended that a ferrite is used on Vusb in bus powered designs.

USB Signal Ground - Should be connected to the ground for the FT232BM / FT245BM chip, but should not be connected to the USB cable shield. Inductors or ferrites should not be placed between USB signal ground and the USB cable shield.

For designs where EMC is a concern it has been seen that a ceramic capacitor in the range 0.01uF and 0.47uF between USB signal ground and the USB cable shield is effective.

USB Cable Shield - Connect to the device case or RS232 / RS422 / RS485 connector housing.

4 RESET Circuit

4.1 RESET#

The chip contains a voltage comparator to generate an internal reset signal. The **RESET#** pin can be tied to VCC or can be driven low to extend the reset period if required. While the reset is active, the pin RSTOUT# will be driven low. RSTOUT# can be used to drive the 1.5K pull up on USBDP. If a longer reset period is required then RSTOUT# will prevent the system detecting the chip until the RESET# pin is driven high.

Internal power-on reset circuit specification:-

1. The chip will be held in reset while the input to RESET# is less than 3.3V.
2. The chip will be reset mode if Vcc drops below 3.3V, provided it remains below 3.3V for a period greater than 250ns.
3. The chip is not released from reset until the RESET# input has been greater than 3.3V for at least 5.6 milliseconds.
4. The chip is not released from reset until clock is running.

5 USB Enumeration

5.1 Enumeration on USB

The presence of a device and it's USB speed is detected by the host by the presence of a pull up resistor on USBDP (for full speed) or on USBDM (for low speed). The FT232BM and FT245BM are full speed USB devices.

When the chip is detected on the host computer, and the drivers have loaded then the PWREN# signal will be driven low in order to power up any external logic on the users board. This should be done using a soft start circuit to avoid sudden glitches in the VCC rail causing the chip to stop working.

6 USB Suspend and Resume

6.1 Suspend and Resume Process

The device may be put into **suspend** by the USB host controller if USB traffic stops for more than 3 milliseconds and the USB bus is idle. For the FT232BM device, the SLEEP# pin will go low to indicate the device has been suspended. For both the FT232BM and the FT245BM, the PWREN# signal will go high when in suspend. This is to reduce overall power consumption to less than 500 micro Amps, which is a requirement of the USB specification. An option in the EEPROM configuration bits will use internal pull down resistors instead of pull up resistors on the I/O pins during suspend. This is to avoid powering up the external logic from the pull ups.

When the device wakes up to an external event such as USB **resume** or reset, it will start a timer and wait for around 5.6 milliseconds before enabling the chip. This delay is to allow time for the oscillator is stabilise.

7 Signals Unique to the FT232BM

7.1 TXDEN, RXLED# and TXLED#

TXDEN - This pin will go high whenever the device transmits a character. It is used for systems where multiple devices can be driving a cable. Its purpose is to control the output enable of a RS485 level converter. It is turned off at the same time as the last STOP bit is sent. See Figure 7 for an oscilloscope trace showing the output of TXDEN.

TXDEN is specified as follows :-

1.TXDEN will go high 1 bit time before the start bit.

2. TXDEN will go low at the end of the stop bit

3. The exact length of the bit time depends on the baud rate selected.

4. If there are consecutive bytes of data TXDEN will go high, and stay high until all data has been passed through.

If 2 stop bits are not being used, then a small delay can be added using logic or an RC network to ensure the TXDEN drives the single STOP bit.

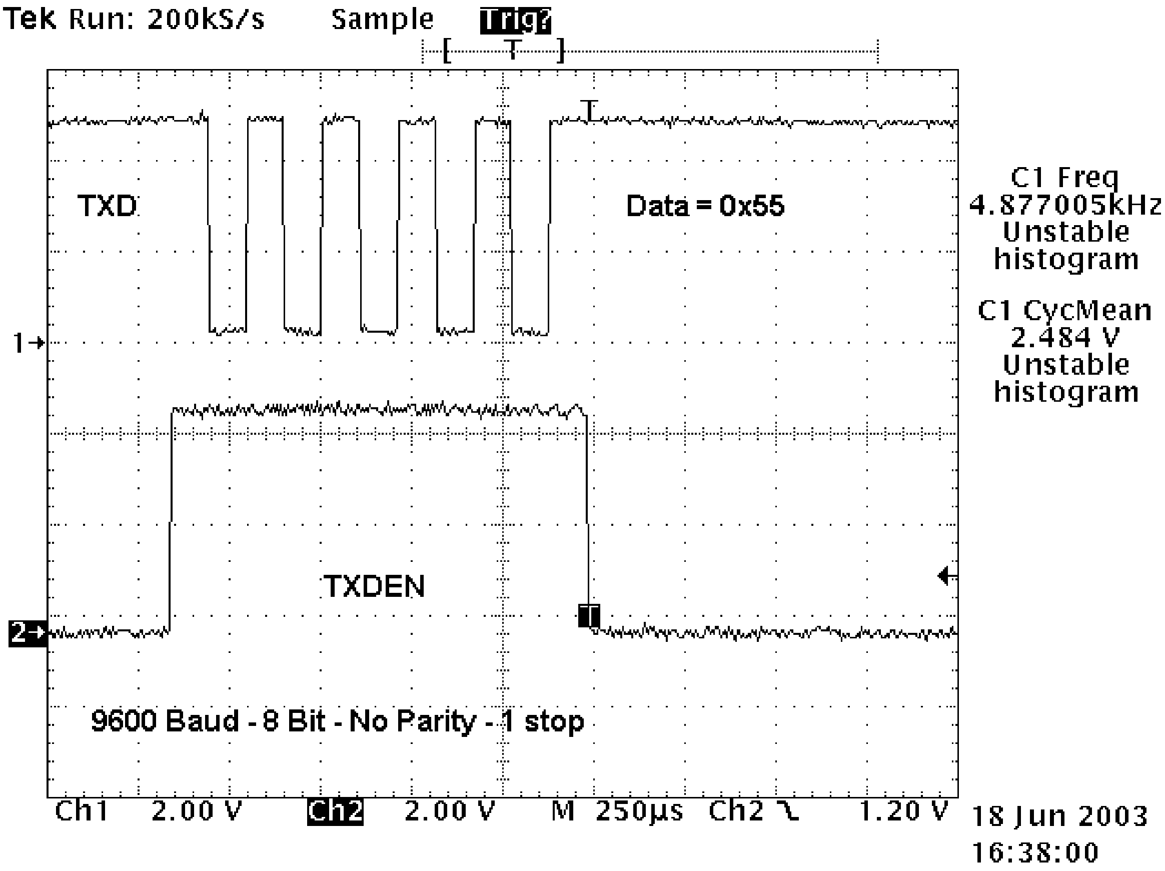


Figure 7. Example TXDEN output.

RXLED# - This is pulsed low for a maximum of 1 millisecond when a character is Received from RS232. It has a recovery time of approximately 1 millisecond before it can be pulsed low again.

TXLED# - This is pulsed low for a maximum of 1 millisecond when a character is Transmitted to RS232. It has a recovery time of approximately 1 millisecond before it can be pulsed low again.

7.2 SLEEP#, PWRCTL, and PWREN#

SLEEP# - This pin will go low when the device is in suspend. When this happens the oscillator will stop. This pin is typically used to power down an external TTL to RS232 level converter I.C. in USB to RS232 converter designs.

PWRCTL - This is an input pin used to tell the system the type of power source for a GET_STATUS USB command. The device will use the values from the EEPROM for the data returned in a CONFIG_DESCRIPTOR for Bus powered or Self powered and remote wakeup. The state of PWRCTL will override the EEPROM setting. If PWRCTL is low then a GET_STATUS command will see the device as bus powered. If PWRCTL is high then a GET_STATUS command will see the device as self powered. This is useful for a system where the device can be Self powered or Bus powered. The Config descriptor will return a value indicating that the device is self powered, and thus the actual source can be read by the GET_STATUS command. If the PWRCTL pin is connected to the external power supply with a 10K ohm pull down, then the power source will be monitored by the system with the GET_STATUS command.

PWREN# - This goes low when the device has been configured over USB by a SET_CONFIGURATION command. This pin will go high during USB suspend.

It is useful in a system where there is a choice of RS232 source. In a modem, for example, there could be two connectors. One would be USB the other normal RS232. This gives the user a choice of which port to connect to. The RS232 lines could be buffered at the 5 volt side using a '244 device. The PWREN# signal can then be used to select the FT232BM chip's RS232 lines when USB is connected.

It can also used to switch the power to external logic to meet the suspend / enumeration current limits. The internal pull down resistors option should be set in the EEPROM when using PWREN# this way.

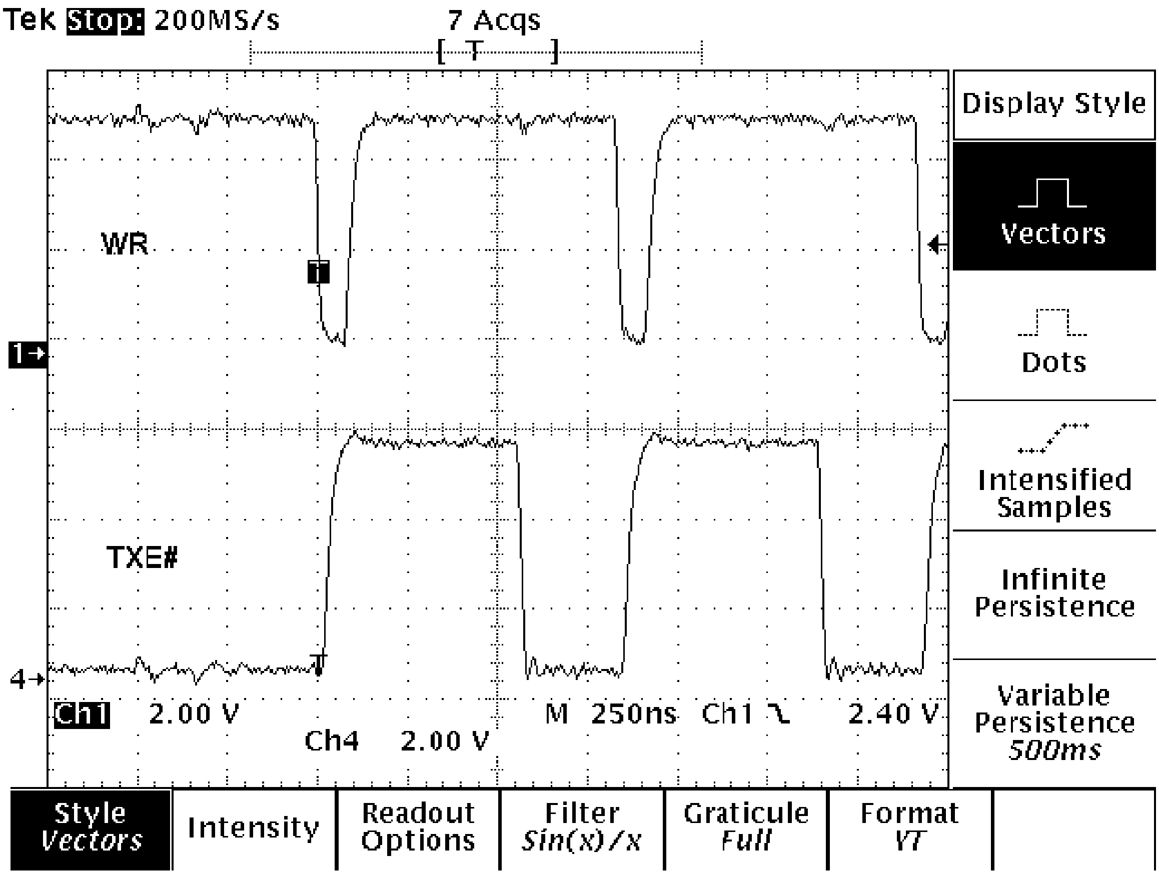
8 Signals Unique to the FT245BM

8.1 WR and TXE#

WR - This input is used to **Write** the data Byte on the D[0...7] lines into the Transmit FIFO Buffer when WR goes from high to low.

TXE# - Active low output indicating **Transmitter Empty**. When high do not write data into the FIFO. When low data can be written into the FIFO by strobing WR high then low.

The oscilloscope trace in Figure 7 shows a typical burst of data to the device. The data is latched on the negative edge of WR. The TXE# signal must be monitored before the next byte can be written.

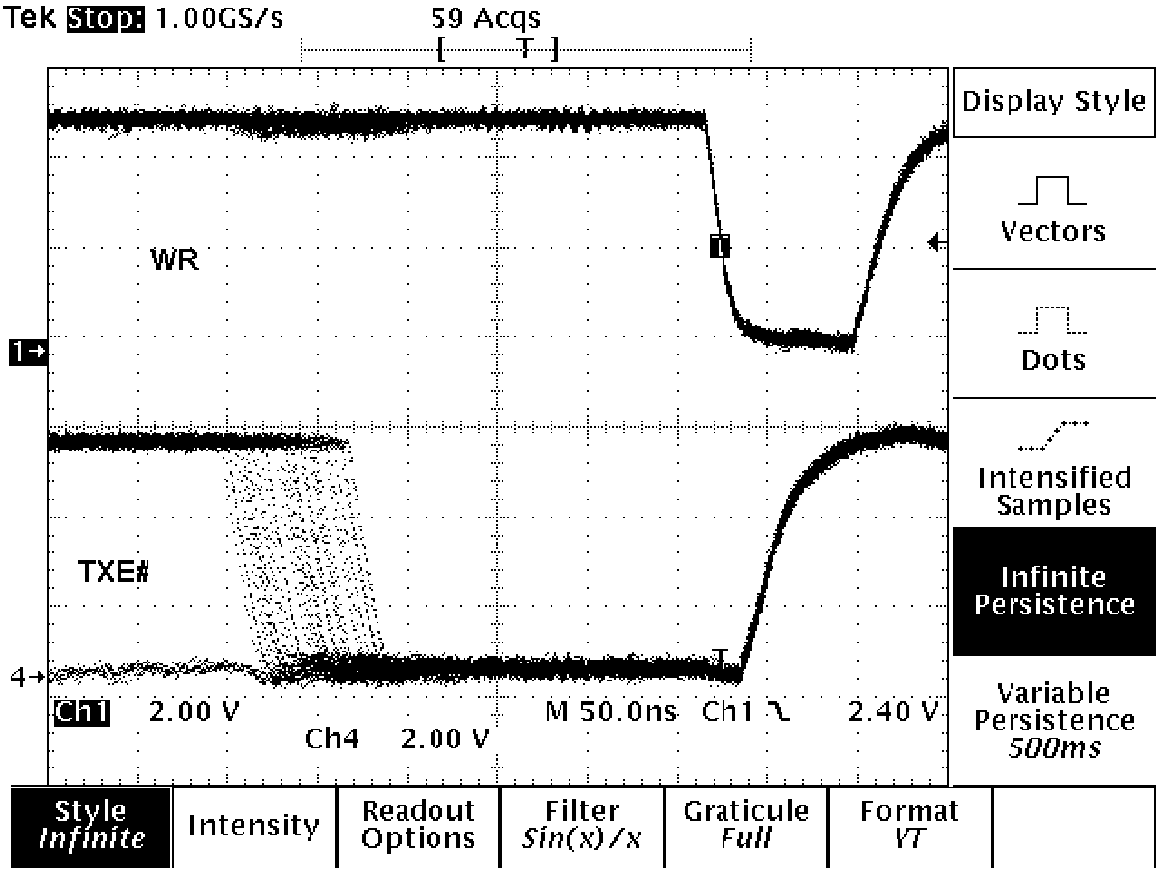


Tek Stop: 200MS/s 7 Acqs

Ch1 2.00 V Ch4 2.00 V M 250ns Ch1 ⌄ 2.40 V

Figure 7. WR and TXE# while writing data.

As can be clearly seen from the following trace (figure 8), the TXE# signal becomes inactive on the falling edge of WR. This example is using one 12 MHz clock period for the WR inactive period, which gives a negative pulse of approx. 80 nanoseconds.


Tek Stop: 1.00GS/s 59 Acqs

Ch1 2.00 V M 50.0ns Ch1 ⟍ 2.40 V

Ch4 2.00 V

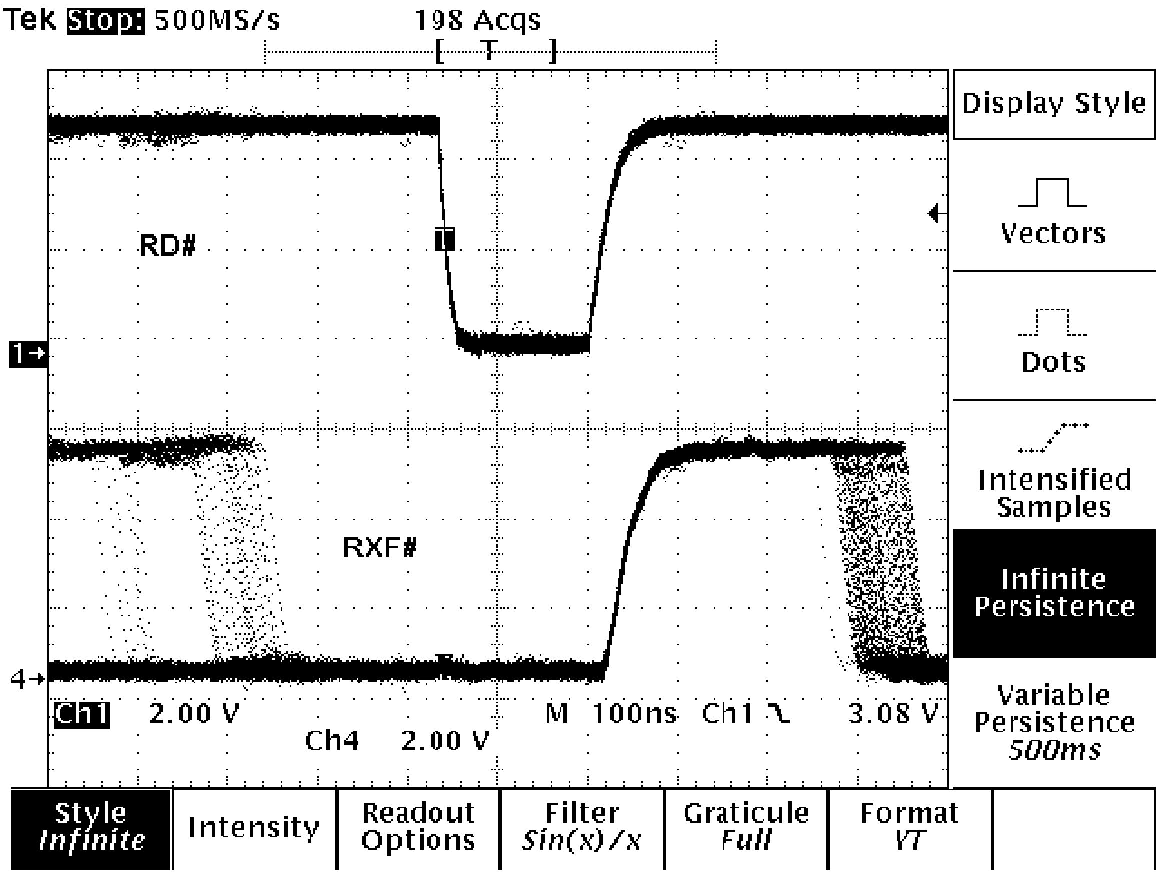
Display Style: Infinite Persistence

Figure 8. TXE# becomes inactive on the falling edge of WR.

8.2 RD# and RXF#

RD# - This active low input is used to **Read** the current data byte from the FIFO Receive buffer onto the D[0...7] lines. If a data Byte is available it will be read when RD# is taken from low to high.

RXF# - Active low output indicating **Receiver full**. When high do not read data from the FIFO. When low data is available in the FIFO which can be read by strobing RD# low then high again.



Style Infinite

Filter Sin(x)/x

Figure 9. shows data being read from the device. The device indicates buffer empty on the rising edge of RD#.

The oscilloscope trace in figure 10 shows the data being driven by the device when RD# goes low. The RD# signal enables the devices data buffer drivers. The data will be valid within a maximum of 50nS from the falling edge of RD#. This example is using two 12 MHz clock periods for the read pulse,.which gives a read time of approx 160 nanoseconds.

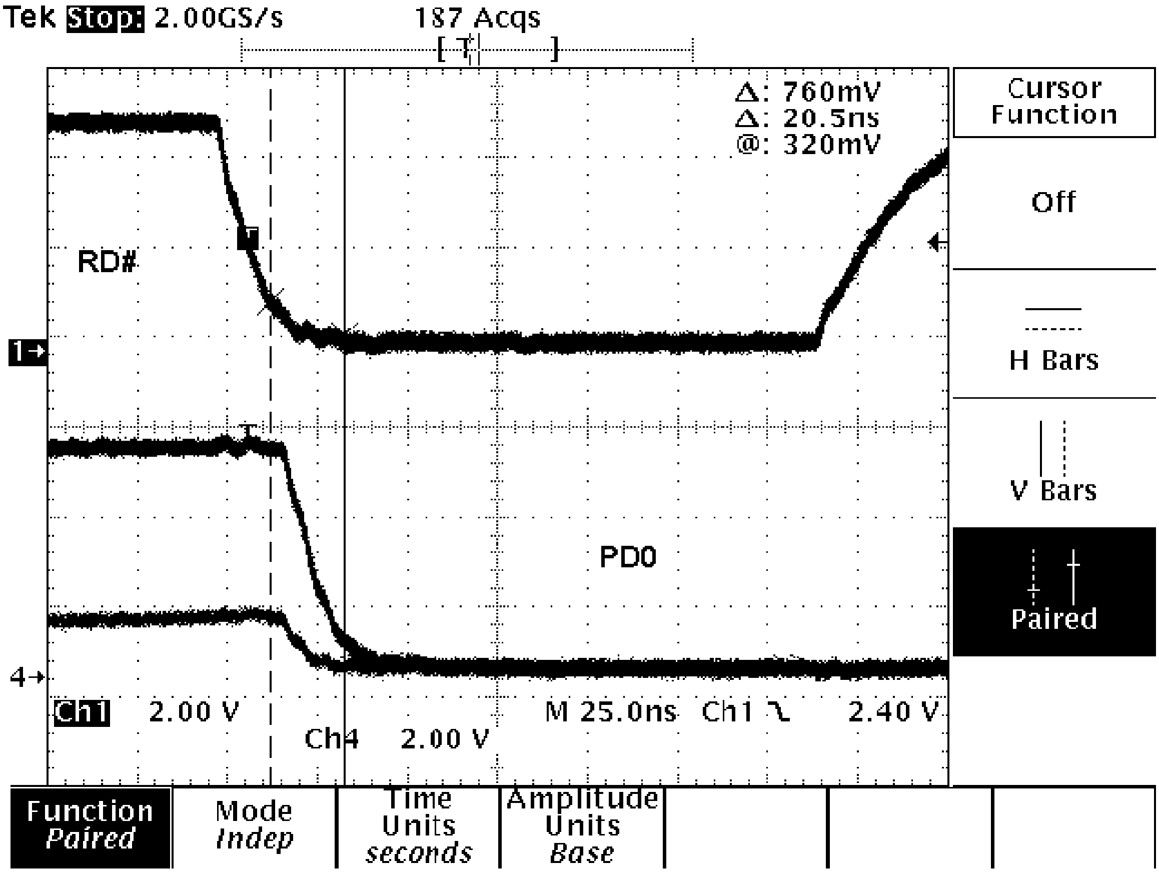


Figure 10. Data being written when RD# goes low.

8.3 PWREN# and SI/WU#

PWREN# - This goes low when the device has been configured over USB by a SET_CONFIGURATION command. This pin will go high during USB suspend.

It can also used to switch the power to external logic to meet the suspend / enumeration current limits. The internal pull down resistors option should be set in the EEPROM when using PWREN# this way.

SI/WU# - Send Immediate / Wake up. This can be used to pass on a short packet of data to the host computer to avoid waiting for the latency timer to trigger. It can also be used to resume the host system if remote wakeup has been enabled to in the EEPROM.

The Send Immediate function activated on a falling edge.

To use SI/WU# to wake up a suspended device the pin should be taken low and held low for 20 milliseconds.

8.4 Data Loss and Data Corruption Problems with the FT245BM

Care must be taken to ensure that the WR and RD# signals have no ringing on them. The lines are edge sensitive so ringing on these lines can cause data loss or data corruption. To avoid this, use a small value of series resistor 27 - 47 Ohms in the lines at the chip that is driving these lines.

Another reason for data loss is not observing the RXF# and TXE# lines. You MUST ONLY write when TXE# is low and you MUST ONLY read when RXF# is low. Not observing these conditions will result in data loss / corruption.

9 Disclaimer and contact information

9.1 Disclaimer

Neither the whole nor any part of the information contained in this document, or the products described, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.
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 these products. Your statutory rights are not affected.
These products, or any variant of them, are 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.

9.2 Contact Information

Future Technology Devices International Limited
373 Scotland Street
Glasgow G5 8QB,
United Kingdom.

Tel : +44 (0)141 429 2777
Fax : +44 (0)141 429 2758

E-Mail (Sales) : sales@ftdichip.com
E-Mail (Support) : support@ftdichip.com
E-Mail (General Enquiries) : admin@ftdichip.com
Web Site URL : www.ftdichip.com

Agents and Sales Representatives

At the time of writing our Sales Network covers over 50 different countries world-wide. Please visit the Sales Network page of our Web site for the contact details our distributor(s) in your country.

Index