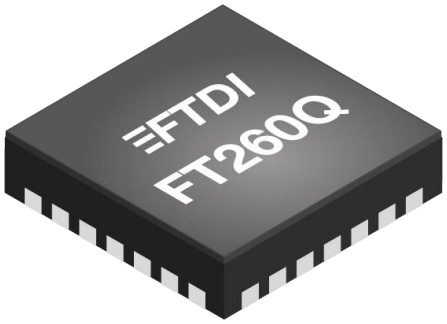
Future Technology Devices International Ltd.

FT260

HID-class USB to UART/I²C Bridge IC



FT260 is a HID-class USB to I²C/UART interface Device Controller with the following advanced features:

- Single chip USB to UART/I²C bridge with standard Human Interface Device (HID) class support
- USB2.0 compliant Full Speed device with entire USB protocol handled on the chip
- Support 2 USB HID Interfaces, each corresponding to the on-chip physical interfaces, I²C and UART
- Pin configuration of enabling HID interface for variety of application
- HID-over-I²C specification support
- Configurable I²C Master Interface controller conforming to I²C v2.1 and v3.0 specification
- Support 4 speed modes defined in I²C-bus specification, Standard mode (SM), Fast mode (FM), Fast mode plus (FM+), and High-Speed mode (HS)
- Robust FTDI UART controller with hardware and software flow control
- Data transfer rate from 1.2Kbaud to 12Mbaud (RS422, RS485, RS232) at TTL levels
- Configurable GPIOs can be easily controlled by software applications under HID class via the USB bus
- I²C bit rate is up to 3Mbps
- With the latest Linux kernels, a new HID-FT260 driver allows use of built in I²C-Tools commands.
- Report Descriptor length of HID-over-I²C is up to 512bytes
- HID-over-I²C supports only the Standard Mode communication
- Fully integrated oscillator PLL with no external crystal required
- On-chip eFUSE for USB Vendor ID (VID), Product ID (PID), and other vendor specific parameters
- Unique USB serial number generation engine and programming path to external EEPROM
- Integrated 5V-3.3V-1.8V level converter for USB I/O
- +5V USB VBUS detection engine
- USB Power Configurations; supports bus-powered, self-powered and bus-powered with power switching
- USB2.0 Low operating and suspend current; 24mA (active-typ) and 405µA (suspend-typ).
- True 3.3V CMOS drive output and TTL input. (operates down to 1V8 with external pull-ups)
- Multiple I/O operating voltage level +3.3V, +2.5V, +1.8V
- Pin output drive strength; 4mA(min) and 16mA(max)
- Integrated power-on-reset circuit
- USB Battery Charger Detection
- UHCI/OHCI/EHCI/xHCI host controller compatible
- Extended operating temperature range; -40 to 85ºC
- Available in compact Pb-free 28 Pin WQFN or TSSOP packages (RoHS compliant)

1 Typical Applications

- HID class Device controller
- USB to HID-over-I^{2}C Bridge
- USB to I^{2}C master controller
- USB to RS232/RS422/RS485 Converters
- Interfacing MCU/PLD/FPGA based designs to USB
- USB Instrumentation

1.1 Driver Support

The USB Human Interface Device (HID) class is natively supported by most operation systems. A custom driver is not required to be installed for the FT260.

- Windows 11, 64-bit
- Windows 10 32, 64-bit
- Windows 8.1 32, 64-bit
- Windows 8 32, 64-bit
- Windows 7 32, 64-bit
- Windows Vista and Vista 64-bit
- Windows XP and XP 64-bit
- Windows CE 4.2, 5.0, 5.2, 6.0
- Windows Server 2008, 2003, 2000
- Windows Embedded Operating Systems
- Mac OS X
- Linux
- Android

1.2 Part Numbers

Part Number	Package
FT260Q-x	28 Pin WQFN
FT260S-x	28 Pin TSSOP

Note: Packing codes for x are:

- R: Taped and Reel, 2,500pcs per reel
- T: Tray packing, 490pcs per tray (WQFN only)
- U: Tube packing, 50pcs per tube (TSSOP only)

For example: FT260Q-R is 2,500pcs taped and reel packing.

1.3 USB Compliant

The FT260 is fully compliant with the USB 2.0 specification and has been given the USB-IF Test-ID (TID) 40001720.

2 FT260 Block Diagram

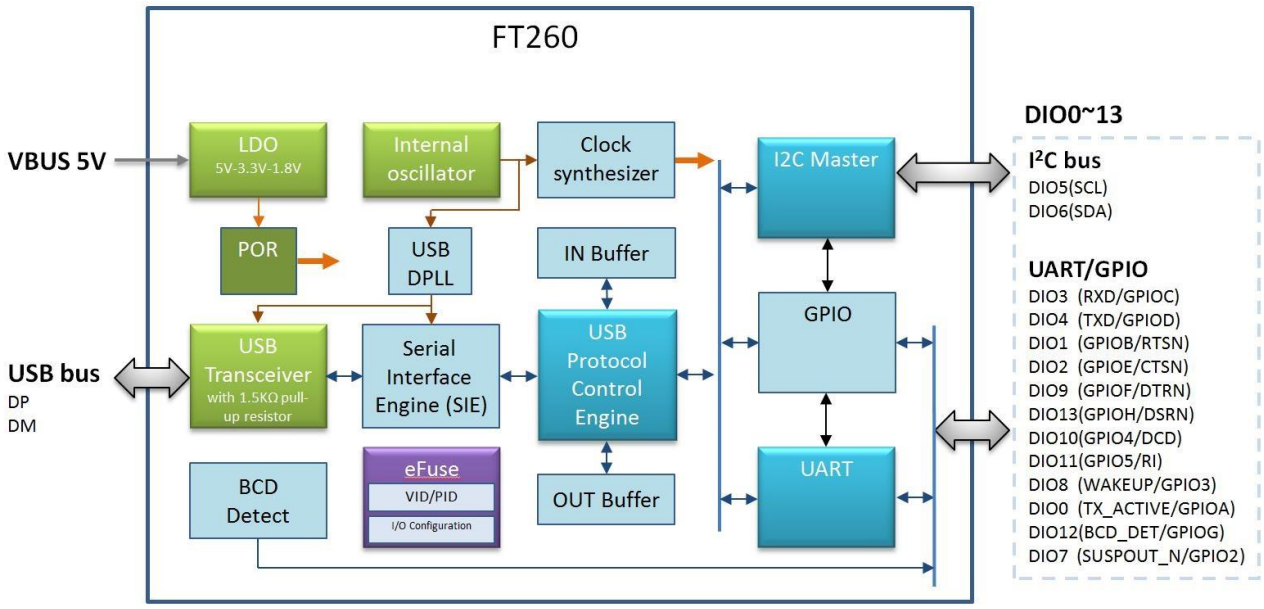


Figure 2-1 FT260 Block Diagram

For a description of each function please refer to Function Description.

Table of Contents

3 Device Pin Out and Signal Description

3.1 WQFN-28 Package Pin Out

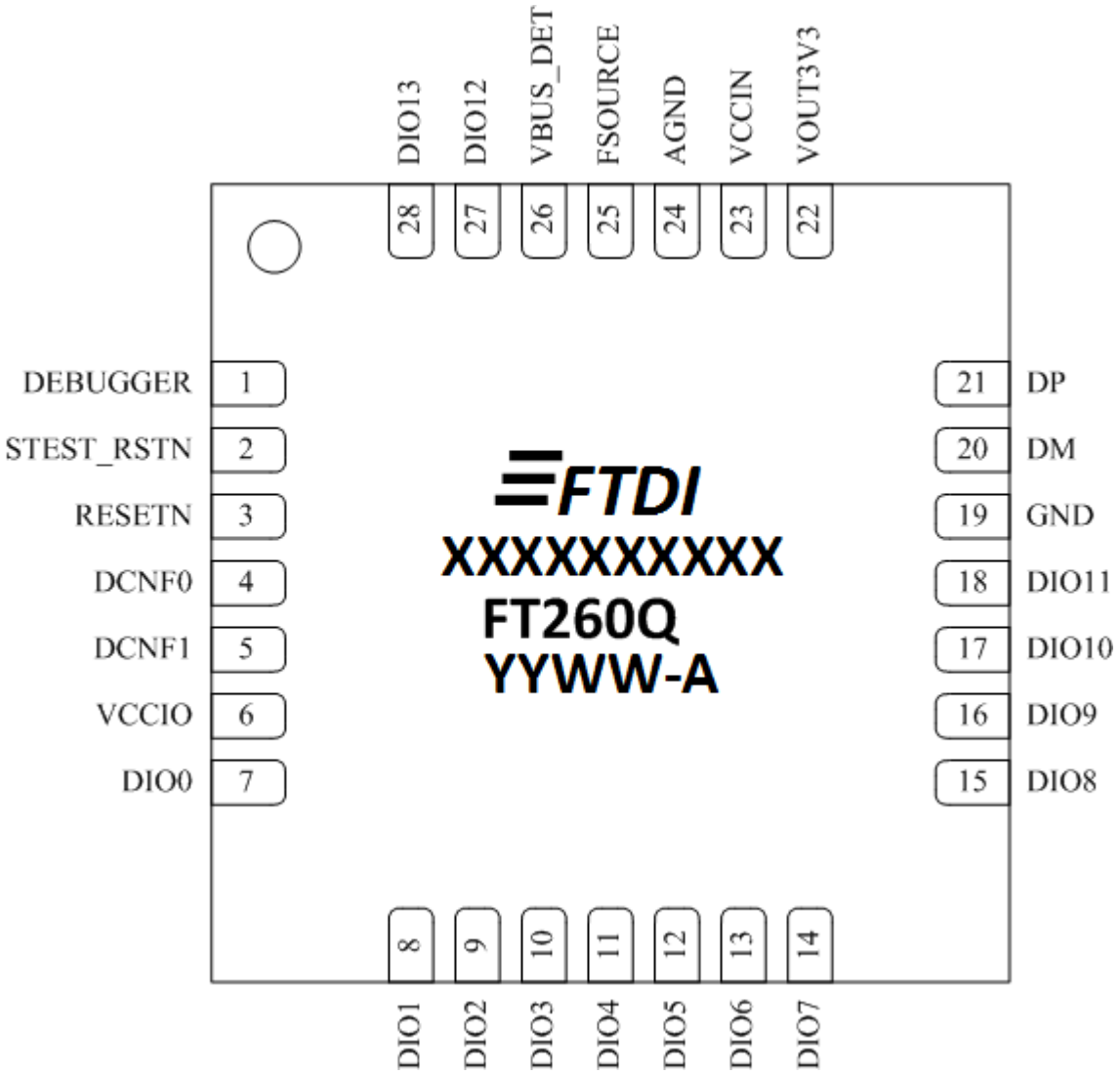


Figure 3-1 Pin Configuration WQFN-28 (top-down view)

3.2 TSSOP-28 Package Pin Out

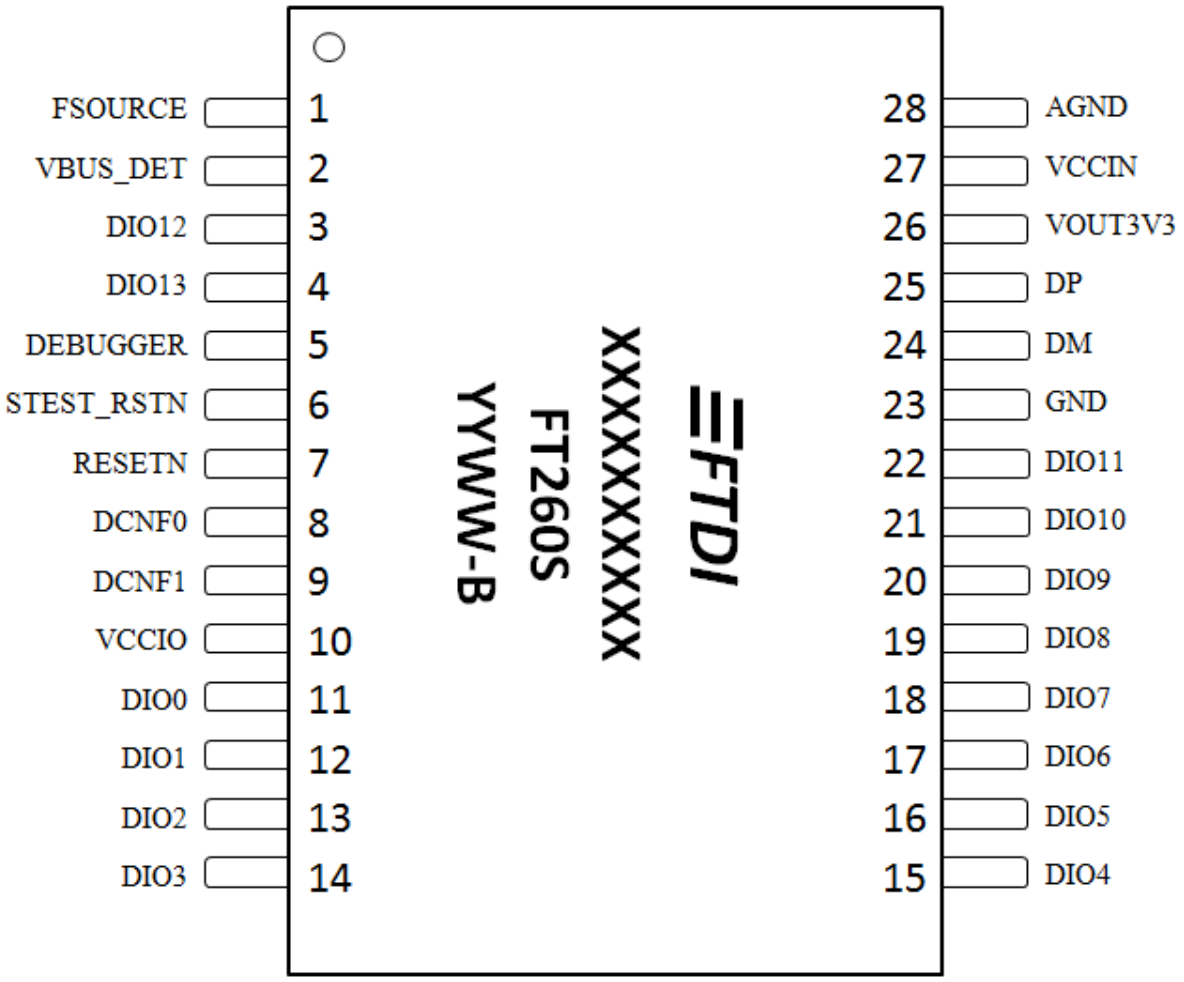


Figure 3-2 Pin Configuration TSSOP-28 (top-down view)

3.3 Pin Description

FT260Q Pin No.	FT260S Pin No.	Pin Name (Function)	Type	Description
1	5	**DEBUGGER**	**I/O**	Debugging pin. Should be reserved and tied to high
2	6	**STEST_RSTN**	**I**	Chip reset input for test mode. Active low. Should be reserved and tied to high.
3	7	**RESETN**	**I**	Chip reset input. Active low. Can be tied to high if external reset function is not required.
4	8	**DCNF0**	**I**	Device Interface Configuration Selection bit-0 for the HID interface selection. Refer to Section 5.1.
5	9	**DCNF1**	**I**	Device Interface Configuration Selection bit-1 for the HID interface selection. Refer to Section 5.1.
6	10	**VCCIO**	**P** **	+3.3V/2.5V/1.8V supply voltage. This is the supply voltage for all the I/O ports. This pin shall be connected to VOUT3V3(pin 22/26) when I/O ports are working at 3.3V.
7	11	**DIO0** (TX_ACTIVE / TX_LED / GPIOA) **See Note4.**	**I/O** O O I/O	DIO0, Digital Input/Output Pin 0. This pin can be configured as one of the following three functions via embedded eFUSE or external EEPROM. TX_ACTIVE is set as the default function to indicate the UART transmitting is active. TX_LED is set as the LED driving source when data is transmitted on UART TX port. GPIOA, General Purpose I/O. GPIOA is another optional function.
8	12	**DIO1** (GPIOB / RTSN)	**I/O** I/O O	DIO1, Digital Input/Output Pin 1. GPIOB, General Purpose I/O. is set as the default function. RTSN, Request To Send Handshake, can be enabled via a USB command for the UART interface.
9	13	**DIO2** (GPIOE / CTSN)	**I/O** I/O I	DIO2, Digital Input/Output Pin 2. GPIOE, General Purpose I/O. is set as the default function. CTSN, Clear To Send Handshake, can be enabled via a USB command for the UART interface.
10	14	**DIO3** (RXD / GPIOC)	**I/O** I I/O	DIO3, Digital Input/Output Pin 3. RXD, Receive Asynchronous Data Input, is set as default function when the UART interface is selected via {DCNF1, DCNF0}. GPIOC, General Purpose I/O, is set as the default function when UART interface is not configured.
11	15	**DIO4** (TXD / GPIOD)	**I/O** O I/O	DIO4, Digital Input/Output Pin 4. TXD, Transmit Asynchronous Data Output, is set as default when the UART interface is selected via {DCNF1, DCNF0}. GPIOD, General Purpose I/O, is set as default when the UART interface is not configured.
12	16	**DIO5** (SCL /	**I/O** I/O	DIO5, Digital Input/Output Pin 5. SCL, Serial clock for I^2C bus with open

FT260Q Pin No.	FT260S Pin No.	Pin Name (Function)	Type	Description
		GPIO0) **See Note 2**	I/O	drain output, is set as the default function. GPIO0, General Purpose I/O. GPIO0 is another optional function and can be enabled via a USB command.
13	17	**DIO6** (SDA / GPIO1) **See Note 2**	**I/O** I/O I/O	DIO6, Digital Input/Output Pin 6. SDA, Serial data for I^2C mode with open drain output, is set as the default function. GPIO1, General Purpose I/O. GPIO1 is another optional function and can be enabled via a USB command.
14	18	**DIO7** (SUSPOUT_N / PWREN_N / TX_LED / GPIO2)	**I/O** O O I/O	DIO7, Digital Input/Output Pin 7. This pin can be configured as one of the following three functions via embedded eFUSE or external EEPROM. SUSPOUT_N is set as the default function as the indicator when entering the USB suspending state. _N means active low. This indicator can also be configured as active high via EEPROM and symbolled as SUSPOUT. PWREN_N is as the power enable indicator when the FT260 is USB enumerated. Active low. TX_LED is set as the LED driving source when data is transmitted on UART TX port. GPIO2, General Purpose I/O. GPIO2 is another optional function and can be enabled.
15	19	**DIO8** (INTRIN / WAKEUP / GPIO3)	**I/O** I I I/O	DIO8, Digital Input/Output Pin 8. INTRIN is the default function as the external interrupt input source. WAKEUP functions as the USB remote wakeup input source. GPIO3, General Purpose I/O. GPIO3 is another optional function and can be enabled via a USB command.
16	20	**DIO9** (GPIOF / DTRN) **See Note 1**	**I/O** I/O O	DIO9, Digital Input/Output Pin 9. GPIOF, General Purpose I/O. is set as the default function. DTRN, Data Terminal Ready, can be enabled via a USB command for the UART interface.
17	21	**DIO10** (GPIO4 / DCD)	**I/O** I/O I	DIO10, Digital Input/Output Pin 10. GPIO4, General Purpose I/O, is set as the default function. DCD, Data Carrier Detection, can be enabled via a USB command for the UART interface.
18	22	**DIO11** (GPIO5 / RI)	**I/O** I/O I	DIO11, Digital Input/Output Pin 11. GPIO5, General Purpose I/O, is set as the default function. RI, Ring Indicator, can be enabled via a USB command for the UART interface. RI may be used as an alternative to WAKEUP for waking up the USB host. WAKEUP feature accompanied with RI can be enabled via the parameter defined in an

FT260Q Pin No.	FT260S Pin No.	Pin Name (Function)	Type	Description
				external EEPROM.
19	23	**GND**	**P**	Ground
20	24	**DM**	**AI/O**	USB peripheral bidirectional DM line.
21	25	**DP**	**AI/O**	USB peripheral bidirectional DP line.
22	26	**VOUT3V3**	**P** **	+3.3V voltage Out May be used to power VCCIO. When VCCIN is supplied with 3.3V, this pin is a power input pin and should be connected to pin 23 / 27.
23	27	**VCCIN**	**P** **	+5.0V (or 3.3V) supply voltage In Power source-in to embedded regulator.
24	28	**AGND**	**P**	Analog Ground
25	1	**FSOURCE**	**AP**	+3.8V supply voltage In Power source for programming embedded eFUSE. It should be kept floating or 0V when not in programming mode
26	2	**VBUS_DET**	**I**	VBUS detection input. It is a +5.0V tolerant pin
27	3	**DIO12** (BCD_DET / RX_LED / PWREN_N / GPIOG)	**I/O** O O O I/O	DIO12, Digital Input/Output Pin 12. This pin can be configured as one of the following three functions via embedded eFUSE or external EEPROM. BCD_DET is the default function as the battery charger detection indicator output when the device is connected to a dedicated battery charger port. Polarity can be defined. RX_LED is as the LED driving source when data is received on UART RX port. PWREN_N is as the power enable indicator when FT260 is USB enumerated. Low active. GPIOG, General Purpose I/O, is another optional function.
28	4	**DIO13** (GPIOH / DSRN)	**I/O** I/O I	DIO13, Digital Input/Output Pin 13. GPIOH, General Purpose I/O, is set as the default function. DSRN, Data Set Ready, can be enabled via USB command for UART interface.

Table 3.1 FT260 Pin Description

**If VCCIN is supplied with 3.3V power input:

-For 3.3V I/O, VOUT3V3 and VCCIO must also be driven with this 3.3V power source.

-For 1.8V/2.5V I/O, VOUT3V3 does not require connection to VCCIN. VCCIO should be connected to the 1.8V/2.5V source.

For details refer to Section 7, for some examples.

Important Notes

NOTE 1: The DTRN pin must be high when the device is powered up and comes out of reset. If this pin is low during start-up, the device will enter test mode which is reserved for FTDI use only. Device pins will behave differently in test mode compared to the normal user mode shown in the table above and may drive out signals. There is an internal weak pull-up on DTRN pin, so it is not necessary to pull this pin high with external circuitry.

NOTE 2: The GPIO0 and GPIO1 pins act as SCL and SDA to check for, and to read from, the external EEPROM on start-up. If using as GPIO, the user should consider that there will be activity on these pins when designing their external circuit.

NOTE 3: Pins (DIO0~13) can be left floating if not used except from DIO5 and DIO6 pins when I2C function is enabled.

NOTE 4: GPIOA pin function configuration differs when using eFuse or I2C EEPROM. GPIOA functionality is not available when using eFuse. See Section 9 for more details.

4 Function Description

The FT260 is a USB device which supports I²C and UART communication through standard USB HID class interfaces. The USB HID class is natively supported by most operating systems. A custom driver is not required to be installed for the FT260.

4.1 Key Features

Highly Functional Integration. FT260 is highly integrated, with a USB2.0 compliant full-speed transceiver, oscillator PLL as the source of the operating clock, LDO regulator for full chip operating power source, eFUSE for basic customization and automatic scanning mechanism of EEPROM for advanced customization. It also includes Power-On-Reset (POR) and VBUS detection input with 5V-tolerance. These embedded functions simplify external circuit design and reduce external component count.

HID class USB to I²C/UART Bridge. FT260 provides the bridge function between standard a USB HID class driver and an I²C slave device and/or UART device. The standard USB HID class driver is natively supported by most operating systems meaning the FT260, does not need a customized driver to be installed. The USB HID class exchanges data between a host and a device by HID reports, which are the actual data blobs follow HID format, and the application developers must communicate with the FT260 via the HID reports. Please refer to Application Notes for detail formats. To help the developers, FTDI also provides a Windows DLL with easy-to-use API for FT260 application development.

There are 2 USB interfaces corresponding to HID class in the FT260. One is for the I²C bus and the other is for the UART bus. These two interfaces can exist concurrently and can be selected independently according to the application. For each interface, there exists one Interrupt IN pipe and Interrupt OUT pipe with a max packet size equal to 64 bytes. With the fastest polling frequency, one time in 1ms, the Interrupt pipes can operate with maximum data throughput up to 64kB/sec. Users can also utilize the HID class commands through the Control pipe to configure the setting and to control the functions in the FT260. Digital function pins can be programmed as GPIO and can be controlled by HID class commands through Endpoint 0.

The I²C bus can run at common I²C bus speeds, Standard mode (SM), Fast mode (FM), Fast mode plus (FM+), and High-Speed mode (HS). A higher bit rate on the I²C bus is also configurable up to 3Mbit/s. Clock stretching is supported to conform to v2.1 and v3.0 of the I²C specification. The default configuration is for standard mode speed (SM). All the configurable settings can be changed over USB before the I²C bus starts any transferring.

The robust FTDI UART bus is embedded in the FT260. The baud rate can be supported from 1200 baud to 12M baud. RTSN/CTSN, DSRN/DTRN and XON/XOFF handshaking options are also supported and can be enabled by associated APIs defined in the DLL for the FT260. Data can be received from the RX pin and delivered to the HID driver via the Interrupt IN pipe. Data can also be delivered from the USB host through the Interrupt OUT pipe and transmitted on to the TX pin.

A remote wake up function is also supported. If the operating system supports remote wake up and allows external hardware to wake it, the FT260 can be resumed by the pin DIO8 which is set by default as WAKEUP triggering a resume signal on USB bus to wake up USB host. DIO11 can also be a remote wake up source when the pin function is set as RI and the parameter, RI as Wake-Up; defined in external EEPROM is enabled.

HID over I²C Bridge. Human Interface Device (HID) is one of the most popular USB devices. It was a protocol developed to simplify the process of connecting accessories such as mouse, keyboard, and touchpad to the PC. HID was originally developed to run over USB or Bluetooth. For Windows 8, Microsoft created a new device type called "HID-over-I²C", which allows the device to communicate HID protocol over an Inter-Integrated Circuit (I²C) bus. The new "HID-over-I²C" devices are only supported natively by Microsoft Windows 8 or above.

The FT260 provides a bridge which connects a "HID-over-I²C" device via an I²C bus, helps to translate USB HID requests from a PC to the device, and makes it work as a normal USB HID class device. With the FT260, an I²C slave function compliant to HID-over-I²C protocol can directly communicate to USB HID class driver through the USB connection.

Configurable Settings for Customization. An electrical poly-fuse (eFUSE) is embedded in the FT260. This embedded eFUSE provides the configurable settings of the Vendor Specific Parameters for basic customization. These Vendor Specific Parameters are the settings about USB, I/O and HID-over-I^2C. Users can utilize this embedded eFUSE to achieve basic customization.

For advanced settings, the FT260 also reserves the programming interface of an external EEPROM via an I^2C interface to record the Vendor Specific Parameters. The FT260 will automatically scan for the presence of an EEPROM. (See Supported EEPROM Spec for suitable devices). When the FT260 is powered up, these Vendor Specific Parameters will be automatically loaded and the FT260 will operate with the parameter setting. When both eFUSE and EEPROM exist at the same time, the Vendor Specific Parameters in the EEPROM will dominate. Both eFUSE and EEPROM can be programmed using the FTDI utility software called FT_PROG, which can be downloaded from the FTDI Utilities page on the FTDI website (https://ftdichip.com/utilities/#ft_prog).

Configurable Digital I/Os. There are 14 digital pins in the FT260 that can be configured for different purposes, such as UART/I^2C bus signals, General Purpose Input/Output (GPIO), LED indicator for data transfer over UART, a USB suspend indicator output, remote wake up input, an interrupt input or power enable indicator. Functions for each pin will be determined during Chip Configuration, with parameters from the eFUSE or EEPROM, or via USB commands.

The signal drive strength of these Digital I/Os can be configured via the FT_Prog utility for different design needs.

Power management. The operating clock for the FT260 can be set as 48MHz, 24MHz, 12MHz. Higher operating frequencies allow higher data throughput. And lower operating frequencies allow lower power consumption. IDLE mode is also supported and can be enabled via the parameters in eFUSE or EEPROM. The system operating clock will be switched to 30 kHz when no data is transferred between USB and I^2C/UART bus for a period of 5 seconds. Any UART RX signalling will trigger the whole chip exiting from the IDLE mode to normal operating status.

USB suspend/resume and remote wakeup are fully supported. The FT260 will be set to a power saving status and the clock to most of the digital circuits will be stopped when the device is suspended.

Source Power and Power Consumption. The FT260 can operate with a voltage supply of +3.3V or +5.0V with a nominal operational mode current of 24mA, a nominal idle mode current of 5.6mA and a nominal USB suspend mode current of 405µA. This allows greater margin for peripheral designs to meet the USB suspend mode current limit of 2.5mA. An integrated level converter within the FT260 allows the device to interface with logic running at +1.8V, 2.5V or +3.3V. (Note: External pull-ups are recommended for IO <3V3).

4.2 Functional Block Descriptions

The following paragraphs detail each function within the FT260. Please refer to the block diagram shown in Figure 2-1 .

Internal Oscillator. The Internal Oscillator cell generates a 48MHz reference clock. With internal trimming mechanisms and an adaptive algorithm, this oscillator provides a stable clock source to the USB DPLL block for generating a recovered clock to Clock Synthesizer block for functional operating.

Clock Synthesizer. The Clock Synthesizer takes the 48MHz clock from the Internal Oscillator and generates 48MHz, 24MHz and 12MHz as reference clocks. The user can select one of these reference clocks as the system operating clock through software over USB. The system operating clock will be the clock source for embedded functions to generate the required interface clock. Higher frequencies should be chosen for higher data throughput demand and lower frequencies for lower power operation. Users can choose the system operating frequency based on the application.

USB Transceiver. The USB Transceiver cell provides the USB 1.1 / USB 2.0 full-speed physical interface. Output drivers provide +3.3V level slew rate control, while a differential input and two single ended input receivers provide data in, Single-Ended-0 (SE0) and USB reset detection conditions respectfully. A 1.5kΩ pull up resistor on USBDP is incorporated.

USB DPLL. The DPLL cell locks on to the incoming NRZI USB data and generates recovered clock and data signals for the Serial Interface Engine (SIE) block.

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 with the USB2.0 specification, it performs bit stuffing/un-stuffing and CRC5/CRC16 generation. It also checks the CRC on the USB data stream.

USB HID Protocol Engine. The USB HID Protocol Engine manages the standard commands from the device control pipe when enumerating. It also handles the Human Interface Device (HID) class commands between the standard HID host driver and the device with I^2C or/and UART functions. With the Device Interface Configuration pins, DCNF0 and DCNF1, it can easily connect HID functions via UART or I^2C interfaces to a PC host driver. Additionally, it can simultaneously support 2 HID functions via UART and I^2C interfaces. This Protocol Engine also includes an IN and OUT Buffer management memory unit which handles the data between USB endpoints and function interfaces such as UART and I^2C.

The USB HID Protocol Engine includes:

- Endpoint-0 for a control pipe with max packet size 64 Bytes
- 2 endpoints for interrupt-in pipe with max packet size equal to 64 Bytes
- 2 endpoints for interrupt-out pipe with max packet size up to 64 Bytes
- Multiple interfaces configuration support
- HID class-specific request parsing and transporting to I^2C/UART bus interface
- Command suspend detection and power management
- Remote wake-up support
- Fully compatible to USB2.0 specification requirement in full speed mode

OUT Buffer. Data sent from the USB host controller to the FT260 via the USB data OUT endpoint is stored in the OUT buffer. Data is removed from the OUT buffer to function interfaces under the control of the USB HID protocol engines. The endpoint buffer size is 64 bytes as the maximum packet size defined for full speed transferring. For the interrupt pipe, the buffer is double buffered for increased throughput.

IN Buffer. Data from the function interfaces is stored in the IN buffer. The USB host controller removes data from the IN buffer by sending a USB request for data from the device data IN endpoint. The endpoint buffer size is 64 bytes as the maximum packet size defined for full speed transferring. For the interrupt pipe, the buffer is double buffered for increased throughput.

UART Controller. When the data and control bus are configured in UART mode, the interface implements a standard asynchronous serial UART port with flow control. 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, DTR, DSR, DCD and RI. The UART provides a transmitter enable control signal (TX_ACTIVE) on the pin DIO0 to assist with interfacing to RS485 transceivers. The UART can support baud rates from 1.2Kbaud to 12Mbaud.

UART in the FT260 functions include:

- Full RS232 support
- 7 or 8 data bits, an optional parity bit and 1 or 2 stop bits support
- Baud rate from 1.2Kbaud to 12Mbaud support
- Baud rate accuracy within +-1.5%
- Optional hardware flow control via RTS / CTS and DTR / DSR
- Optional software flow control via XON / XOFF characters

I^2C Master Controller. I^2C (Inter Integrated Circuit) is a multi-master serial bus invented by Philips. I^2C uses two bi-directional open-drain wires called serial data (SDA) and serial clock (SCL). Common I^2C bus speeds are the standard mode (SM) with bit rate up to 100 Kbit/s, fast mode (FM) with the bit rate up to 400 Kbit/s, Fast mode plus (FM+) with the bit rate up to 1 Mbit/s, and High-Speed mode (HS) with the bit rate up to 3.4 Mbit/s. Refer to the I^2C specification for more information on the protocol.

The FT260 device can operates as I^2C master, and the major functions include:

- Fully compatible to v2.1 and v3 specification
- 7-bit address support
- Support 4 speed configurations defined in I^2C-bus specification
- Support bit rate up to 3Mbit/s

- Clock stretching support

GPIOs. The FT260 contains 14 digital function pins. Each pin can be set as I^2C/UART related function or GPIO (General Purpose Input/Output). Some GPIO functions are implemented in the FT260 for various applications like TX_ACTIVE, TX_LED, RX_LED for UART; SUSPOUT_N, WAKEUP for USB; PWREN and BCD_DET indicator for power management. GPIO functions can also be directly controlled by applications over USB via the Control pipe. The drive strength, slew rate control and pull high/low resistors can be configured in the Vendor Specific Parameters defined in embedded eFUSE or external EEPROM by FT_PROG.

GPIO functions for each pin in the FT260 include:

- DIO0 (pin 7 @ WQFN28) can be configured as TX_ACTIVE, TX_LED, GPIOA
- DIO1 (pin 8 @ WQFN28) will be set as GPIOB function by default
- DIO2 (pin 9 @ WQFN28) will be set as GPIOE function by default
- DIO3 (pin 10 @ WQFN28) will be set as GPIOC function by default when the UART interface in not enabled
- DIO4 (pin 11 @ WQFN28) will be set as GPIOD function by default when the UART interface in not enabled
- DIO5 (pin 12 @ WQFN28) can be set as GPIO0 function when the I^2C interface and external EEPROM are not supported
- DIO6 (pin 13 @ WQFN28) can be set as GPIO1 function when the I^2C interface and external EEPROM are not supported
- DIO7 (pin 14 @ WQFN28) can be configured as SUSPOUT_N, SUSPOUT, PWREN_N, GPIO2
- DIO8 (pin 15 @ WQFN28) can be configured as WAKEUP, GPIO3
- DIO9 (pin 16 @ WQFN28) will be set as GPIOF function by default
- DIO10 (pin 17 @ WQFN28) will be set as GPIO4 function by default
- DIO11 (pin 18 @ WQFN28) will be set as GPIO5 function by default
- DIO12 (pin 27 @ WQFN28) can be configured as BCD_DET, RX_LED, PWREN_N and GPIOG
- DIO13 (pin 28 @ WQFN28) will be set as GPIOH function by default

eFUSE Controller + Internal eFUSE. The internal eFUSE (electrical poly fuse) provides storage for the Vendor Specific Parameters. These Vendor specific Parameters are for the purpose of cost-effective customization. When FT260 is powered up, all the parameters will be automatically loaded into and taken effective before operation. The embedded eFUSE can be programmed over USB with an external voltage requirement on the pin FSOURCE with 3.8V power source. These parameters can be programmed using the FTDI utility software called FT_PROG, which can be downloaded from FTDI Utilities on the FTDI website (https://ftdichip.com/utilities/#ft_prog).

Vendor Specific Parameters in eFUSE include:

- USB Vendor ID (VID), Product ID (PID), power type selection
- DIO0, DIO7, DIO12 function selection
- Digital pins driving strength selection (4mA, 8mA, 12mA, 16mA)
- HID-over-I^2C Slave Address
- HID-over-I^2C Interrupt Type
- HID-over-I^2C SET_/GET_IDLE, SET_/GET_PROTOCOL, SET_POWER enable control

For further details refer to Section 9.1.

5V-3.3V-1.8V LDO regulator. The LDO will regulate out 2 reference voltages for use within the FT260. The +3.3V LDO regulator generates the +3.3V reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the VOUT3V3 regulator output pin. Another +1.8V LDO regulator generates the +1.8V reference voltage for driving the internal core of the IC.

POR RESET Generator. POR is the integrated Power on Reset Generator Cell providing a reliable power-on reset to the device internal circuitry at power up. There is also a RESETN input pin allowing an external device to reset the FT260. RESETN can be tied to VCCIO (+3.3v) if not being used.

Embedded BCD Detection. Supports Battery Charger Detection. When the pin DIO12 is set as BCD_DET function, it will be active if the device is connected to a dedicated charger instead of a standard USB Host.

5 FT260 Configuration and Bus Interfaces

5.1 Device Interface Configuration

The FT260 has 2 HID interfaces and can be selected by {DCNF1, DCNF0}. The first HID Interface is for the bridge function from USB HID driver to I^2C bus interface. And the second HID Interface is for the bridge function from USB HID driver to UART bus interface. The following table shows the USB interfaces corresponding to the chip configuration mode.

DCNF1	DCNF0	HID Interfaces
0	0	Both interfaces for I^2C and UART are enabled. Interfaces will be created as: – Interface-0 is set as the interface for I^2C to send and receive data via I^2C connection – Interface-1 is set as the interface for UART to send and receive data via UART connection – DIO3 and DIO4 are set as RXD and TXD for UART by default
0	1	Only the interface for I^2C is enabled. The interface will be created as: – Interface-0 is set as the interface for I^2C to send and receive data via an I^2C connection – DIO3 and DIO4 are set as GPIO functions by default.
1	0	Only the interface for UART is enabled. The interface will be created as: – Interface-0 is set as the interface for UART to send and receive data via a UART connection – DIO3 and DIO4 are set as RXD and TXD for UART by default
1	1	Both Interfaces for I^2C and UART are enabled. Interfaces will be created as: – Interface-0 is set as the interface for I^2C to send and receive data via an I^2C connection – Interface-1 is set as the interface for UART to send and receive data via a UART connection – DIO3 and DIO4 are set as RXD and TXD for UART by default

Table 5.1 FT260 USB Device Interface Configuration

Note that the default functions for the pins, GPIOC and GPIOD, will be determined by Device Interface Configuration. When the interface for UART is enabled, the pin DIO3 is assigned as RXD for UART and DIO4 is assigned as TXD for UART.

DIO5 and DIO6 are default designed as SCL and SDA for the I^2C bus. It means that the I^2C master controller is enabled by default no matter if the interface for I^2C is enabled or not. Users can set the DIO5 and DIO6 as the GPIO functions via USB commands if the interface for the I^2C is disabled and connectivity to the external I^2C devices is not required.

5.2 I²C Bus Interface

I²C (Inter Integrated Circuit) is a multi-master serial bus invented by Philips. I²C uses two bi-directional open-drain wires called serial data (SDA) and serial clock (SCL). Common I²C bus speeds are standard mode (SM) with a bit rate up to 100Kbit/s, fast mode (FM) with a bit rate up to 400Kbit/s, Fast mode plus (FM+) with a bit rate up to 1Mbit/s, and High-Speed mode (HS) with the bit rate up to 3.0Mbit/s.

An I²C bus node can operate either as a master or a slave:

- Master node – issues the clock and addresses slaves
- Slave node – receives the clock line and address

The FT260 operates as an I²C master and is capable of being set to the speed modes defined in the I²C bus specification. Besides the speed mode defined in the I²C standard specification, the I²C controller of the FT260 can support flexible SCL frequencies defined by the following function.

$$SCL\ Freq = \frac{\text{Operating Clock Frequency}}{M*(N+1)} \quad M = 6\ or\ 8;\ \ N = 1, 2, 3, \ldots\ldots, 127$$

When the target frequency is below 100KHz or higher than 1MHz, M will be equal to 8; otherwise, M will be equal to 6. For example, to generate a 3MHz frequency on SCL, M will be selected as 8. With the operating clock frequency equal to 48MHz, the user can set N as 1. The SCL frequency of the I²C master mode for the FT260 can be set via USB commands. Details can be referenced in the FT260 Application Notes.

5.2.1 I²C Pin Definition

The I²C function in the FT260 is an I²C master device. It is enabled by default when the FT260 is powered up and the operating speed on the I²C bus is designed as 60KHz for connectivity to most of the external I²C slave devices. The I²C pins of the FT260 are:

- Clock – SCL (DIO5, pin 12 @ WQFN28), as clock output with open-drain design
- Data – SDA (DIO6, pin 13 @ WQFN28), command/address/data transfer between master and slave with open-drain design

5.2.2 I²C Bus Protocol

There are four potential modes of operation for a given bus device, although most devices only use a single role (Master or Slave) and its two modes (Transmit and Receive):

- Master transmit – sending data to a slave
- Master receive – receiving data from a slave
- Slave transmit – sending data to a master
- Slave receive – receiving data from the master

The following figure shows the basic I²C bus protocol.

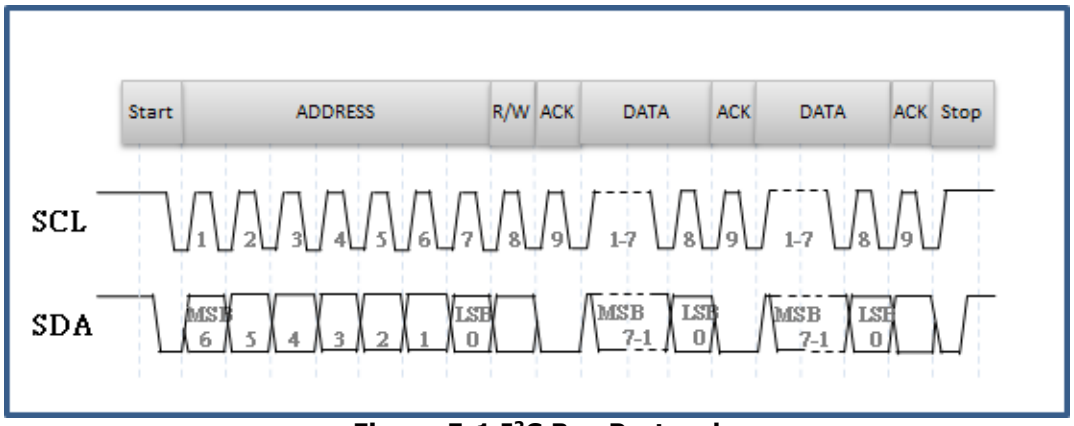


Figure 5-1 I²C Bus Protocol

The master is initially in master transmit mode by sending a start bit followed by the 7-bit address of the slave it wishes to communicate with, which is finally followed by a single bit representing whether it wishes to write(0) to or read(1) from the slave.

If the slave exists on the bus, then it will respond with an ACK bit (active low for acknowledged) for that address. The master then continues in either transmit or receive mode (according to the read/write bit it sent), and the slave continues in its complementary mode (receive or transmit, respectively).

The address and the data bytes are sent most significant bit first. The start bit is indicated by a high-to-low transition of SDA with SCL high; the stop bit is indicated by a low-to-high transition of SDA with SCL high.

If the master wishes to write to the slave, then it repeatedly sends a byte with the slave sending an ACK bit. (In this situation, the master is in master transmit mode and the slave is in slave receive mode.)

If the master wish to read from the slave, then it repeatedly receives a byte from the slave, the master sends an ACK bit after every byte but the last one. (In this situation, the master is in master receive mode and the slave is in slave transmit mode.). The master then ends transmission with a stop bit, or it may send another START bit if it wishes to retain control of the bus for another transfer (a "combined message").

I²C defines three basic types of messages, each of which begins with a START and ends with a STOP:

- Single message where a master writes data to a slave;
- Single message where a master reads data from a slave;
- Combined messages, where a master issues at least two reads and/or writes to one or more slaves

In a combined message, each read or write begins with a START and the slave address. After the first START, these are also called repeated START bits; repeated START bits are not preceded by STOP bits, which is how slaves know the next transfer is part of the same message.

Users can refer to the I²C specification for more information on the protocol.

5.2.3 I²C Slave Address

The FT260 is configured as a USB to I²C master bridge and can issue any value of 7-bits slave address. Users can issue I²C commands towards an I²C slave device to read or write data via the applications defined in USB host side. For details, refer to the FT260 Application Notes.

When the FT260 is powered up, the I²C master controller will start to scan the external I²C device. The scanning address range is from 50h to 57h for the types of EEPROM. For further details refer to Section 9.2.

5.2.4 I²C Timing

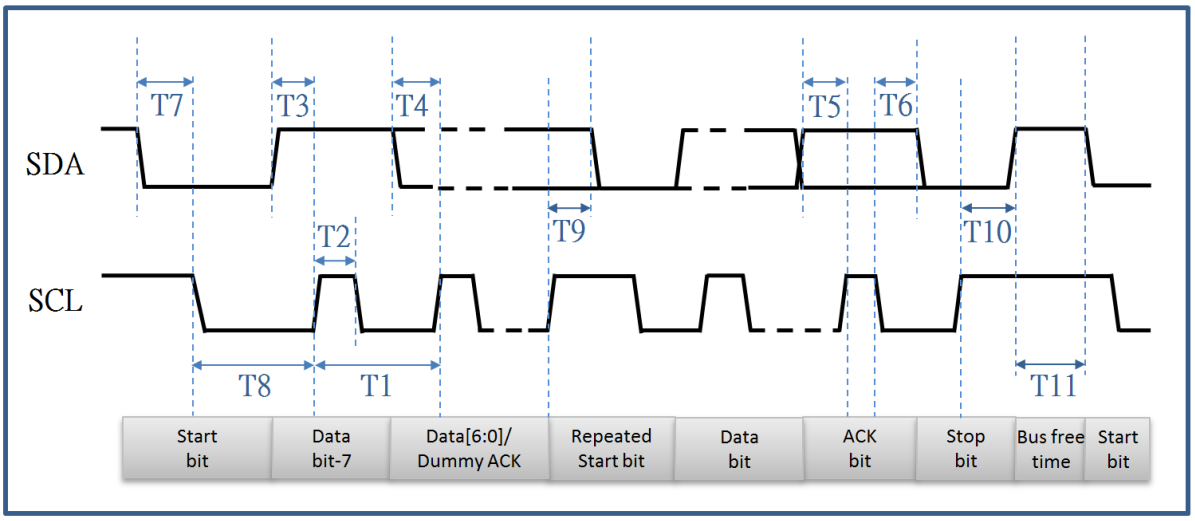


Figure 5-2 I²C Bus Timing

Parameter	Min(ns)	Typ(ns)	Max(ns)	Description
T0@12MHz		83.333		T0 is the period when operating clock=12MHz
T0@24MHz		41.666		T0 is the period when operating clock=24MHz
T0@48MHz		20.833		T0 is the period when operating clock=48MHz
T1@SM/HM	16*T0	8*(1+N)*T0		SCK Period when I²C as master with standard speed mode (SM) and HS speed mode
T1@FM/HM	12*T0	6*(1+N)*T0		SCK Period when I²C as master with FM, FM+ speed mode
T2	8*T0	4*(1+N)*T0		SCK high pulse width when I²C as master with standard speed mode (SM) and HS speed mode
T2	4*T0	2*(1+N)*T0		SCK high pulse width when I²C as master with FM, FM+ speed mode
T3		2*(1+N)*T0		SDA output setup time to SCL rising edge when I²C as master
T4		2*(1+N)*T0		SDA output hold time to SCL falling edge when I²C as master
T5			>=0	input setup time requirement from SDA to SCL rising edge when I²C as master
T6			>=0	input hold time requirement from SDA to SCL falling edge when I²C as master
T7		2*(1+N)*T0		Start bit setup time to SCL falling edge
T8		4*(1+N)*T0		Start bit hold time to SCL falling edge
T9		2*(1+N)*T0		Stop bit setup time to SCL rising edge
T10		2*(1+N)*T0		Stop bit hold time to SCL rising edge
T11	4*(1+N)*T0			Bus free time between Start and Stop bit

Table 5.2 I²C Timing for VCCIO=3.3V

NOTE: N can be ranged from 1 to 255.

5.3 UART Interface

A universal asynchronous receiver/transmitter (UART) is a computer hardware device that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as TIA (formerly EIA) RS-232, RS-422 or RS-485.

The UART can support baud rates from 1.2Kbaud to 12Mbaud defined by the following function.

$$\textbf{Baud Rate} = \frac{\boldsymbol{Operating\ Clock\ Frequency}}{\boldsymbol{Baud\ Divisor}}$$

The baud divisor is used to divide the operating clock frequency to the desired baud rate. It can take any value between 4 and 40000 with the added option of adding a fractional component in the order of 1/8ths.

Example: To generate an 115200 baud rate in the FT260, the operating clock frequency to the UART controller equals to 48MHz. The baud divisor can be calculated as shown in the below equation.

$$\textbf{Baud Divisor} = \frac{\boldsymbol{48MHz}}{\boldsymbol{115200Hz}} = \boldsymbol{416.667}$$

Due to the fractional component is the order of 1/8ths, the baud divisor must be selected as 416.625. It is obvious that the difference of baud divisors will produce a percentage error. A comparison of standard baud rates and the divisor values can be seen in Table 5.3. This shows the baud rate required, followed by the divisor value needed to achieve this if the UART is running off a 48MHz clock. Then it lists the actual baud rate achieved and finally the percentage error this produces.

Target Baud Rate	Ideal Baud Divisor	Actual Baud Divisor	Actual Baud Rate	Baud Error Rate
12,000,000	4	4	12,000,000	0.00%±0.25%*Note
9,600,000	5	5	9,600,000	0.00%±0.25%
8,000,000	6	6	8,000,000	0.00%±0.25%
6,000,000	8	8	6,000,000	0.00%±0.25%
3,000,000	16	16	3,000,000	0.00%±0.25%
2,000,000	24	24	2,000,000	0.00%±0.25%
1,500,000	32	32	1,500,000	0.00%±0.25%
1,000,000	48	48	1,000,000	0.00%±0.25%
921,600	$52.08\overline{3}$	52	923,076.9231	0.16%±0.25%
460,800	$104.1\overline{6}$	104.125	460,984.3938	0.04%±0.25%
230,400	$208.\overline{3}$	208.250	230,492.1969	0.04%±0.25%
115,200	$416.\overline{6}$	416.625	115,211.5212	0.01%±0.25%
57,600	$833.\overline{3}$	833.250	57,605.7606	0.01%±0.25%
38,400	1,250	1250	38,400	0.00%±0.25%
19,200	2,500	2500	19,200	0.00%±0.25%
9,600	5,000	5000	9,600	0.00%±0.25%
4,800	10,000	10000	4,800	0.00%±0.25%
2,400	20,000	20000	2,400	0.00%±0.25%

Target Baud Rate	Ideal Baud Divisor	Actual Baud Divisor	Actual Baud Rate	Baud Error Rate
1,200	40,000	40000	1,200	0.00%±0.25%

Table 5.3 Baud Rate Comparison

*Note that the baud error rate with ±0.25% is from the internal oscpll.

5.3.1 UART Pin Definition

The UART function in the FT260 can be configured as UART-only or I²C plus UART mode by DCNF0 and DCNF1 pins. The mode selection is as shown in the Table 5.2.

The pins of the FT260 will be mapped accordingly. The UART pins are:

- Receive Data (RXD) – DIO3 (pin-10 @ WQFN28)
  serial data input.
- Transmit Data (TXD) – DIO4 (pin-11 @ WQFN28)
  serial data output.
- Transmit Active signal (TX_ACTIVE) – DIO0 (pin-7 @ WQFN28)
  active high when data transmission is in progress. Asserted one clock cycle before start bit and de-asserted with final stop bit.
- Request To Send Signal (RTSN) – DIO1 (pin-8 @ WQFN28)
  active low handshaking bit. When low it indicates that the UART can start receiving Rx Data.
- Clear To Send Signal (CTSN) – DIO2 (pin-9 @ WQFN28)
  active low handshaking bit. When this bit is '1', the UART should stop sending TX Data.
- Data Terminal Ready (DTRN) – DIO9 (pin-16 @ WQFN28)
  active low and when '0', indicates that the UART can be connected and receive RX Data.
- Data Set Ready (DSRN) – DIO13 (pin-28 @ WQFN28)
  active low indicating an active connection. When this bit is '1', the UART should not send TX Data.
- Data Carrier Detect (DCD) – DIO10 (pin-17 @ WQFN28)
  asserted when a connection has been established with external device.
- Ring Indicator (RI) – DIO11 (pin-18 @ WQFN28)
  asserted when requested to wake up.

5.3.2 UART Bus Protocol

Data transferring uses NRZ (Non-Return to Zero) data format consisting of 1 start bit, 7 or 8 data bits, an optional parity bit, and one or two stop bits.

- Data Bits - 7 data bits or 8 data bits.
- Parity Bit - **No** parity.
  - **Odd** parity. This means that the parity bit is set to either '1' or '0' so that an odd number of 1's is sent.
  - **Even** parity. This means that the parity bit is set to either '1' or '0' so that an even number of 1's is sent.
  - **High** parity. This simply means that the parity bit is always High.
  - **Low** parity. This simply means that the parity bit is always Low.
- Stop Bits - one bit or two bits.

When transmitting the data bits, the least significant bit is transmitted first. UART transmitting and receiving waveforms are illustrated in the Figure 5-3, Figure 5-4, Figure 5-5, Figure 5-6 and Figure 5-7.

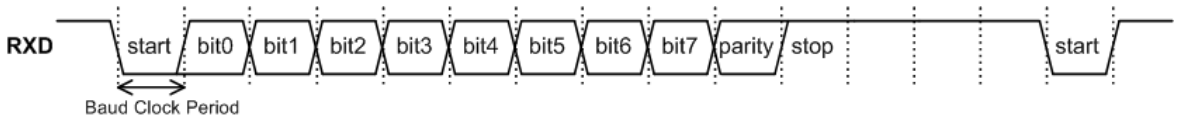


Figure 5-3 UART RX Waveform consist of 8 data bits, 1 optional parity bit and 1 stop bit

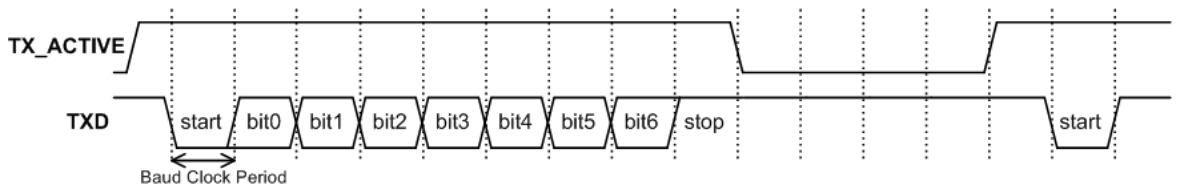


Figure 5-4 UART TX Waveform consist of 7 data bits, no parity bit and 1 stop bit

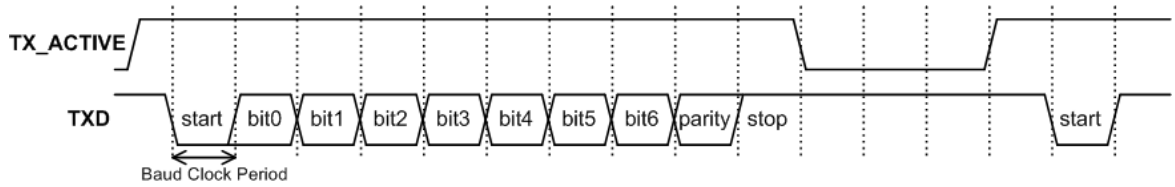


Figure 5-5 UART TX Waveform consist of 7 data bits, 1 optional parity bit and 1 stop bit

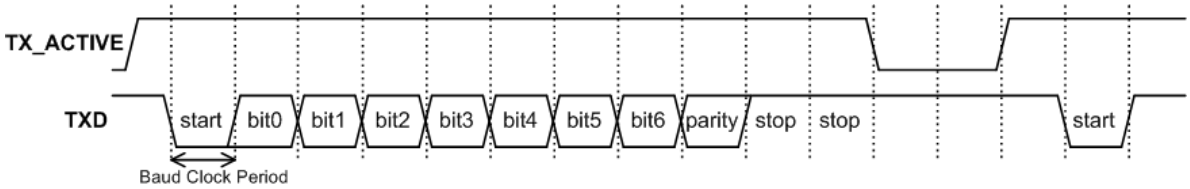


Figure 5-6 UART TX Waveform consisting 7 data bits, 1 optional parity bit and 2 stop bits

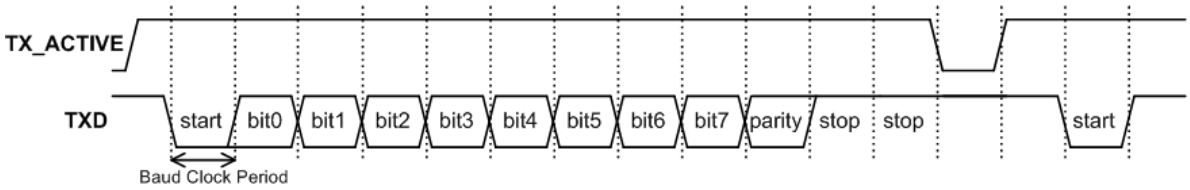


Figure 5-7 UART TX Waveform consisting 8 data bits, 1 optional parity bit and 2 stop bits

TX_ACTIVE is default function of the pin DIO0 as the transmitting indicator for UART; this output may be used in RS485 designs to control the transmitting of the line driver.

5.3.3 UART Flow Control

The UART interface needs to implement proper flow control to prevent data from being lost by the external device. This will be done using either hardware or software flow control. The FT260 UART supports the modes listed below.

- OFF, and switch UART pins to GPIO
- RTS_CTS mode (hardware flow control)
- DTR_DSR mode (hardware flow control)
- XON_XOFF (software flow control)
- No flow control mode

RTS/CTS Hardware Flow Control

When RTS / CTS flow control is used, the CTS input indicate to the FT260 that the data communications equipment (DCE) is ready to receive data. If it is active (low), then the FT260 is free to transmit data on the TX data line, otherwise it must hold the data until CTS goes low. The RTS output is used to indicate that The FT260 can receive data (active low). Thus, it should be set inactive by the FT260 when both the UART receive register and receive holding register are full.

DTR/DSR Hardware Flow Control

These signals are provided to give information about the status of each UART. When this mode is enabled and the DTS input is high, the FT260 UART should not send any data on the TX line. DTR will be enabled on reset and a register bit will allow the IO Bus to alter the state at any time.

Software Flow Control

When software flow control is enabled the XON character and XOFF character are used to stop and start the flow of data. The XON character tells the downstream device to start sending data. The XOFF character tells the downstream device to stop sending data. Typical defaults for XON are `11' and for XOFF is `13'.

5.3.4 UART Timing

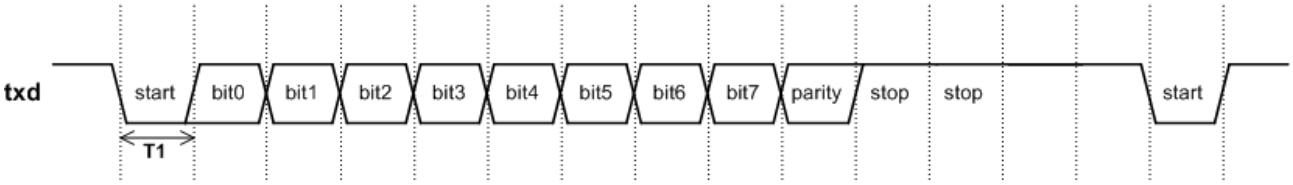


Figure 5.8 UART Timing

Parameter	Min(ns)	Typ(ns)	Max(ns)	Description
T0@48MHz	-2500ppm	20.833	+2500ppm	T0 is the period when operating clock=48MHz
T1	4*T0		40000*T0	Baud clock period of txd

Table 5.4 UART Timing

5.4 GPIOs

Most of the digital I/O pins of the FT260 have a GPIO function as an alternative function. For example, pin 10 can be RXD or GPIOC, and RXD is the main function of pin 10. Usually, if the main function of a pin is switched off, the pin will be switched to a GPIO pin. Therefore, if UART is turned off, all UART pins will become GPIO pins. The FT260 has 3 pins which have more than 2 functions. They are GPIO 2(pin 14), GPIOA (pin 7), and GPIOG (pin 27). The working function of these 3 pins can be configured by eFUSE, EEPROM, or via USB commands. Please refer to the eFUSE and EEPROM sections for more details.

The FT260 has two sets of GPIO pins: GPIO0~5 and GPIOA~H. After the pins are configured as GPIO, users can set or get the direction and pin status via the USB control pipe, i.e., HID SET_REPORT and GET_REPORT requests. The FT260 also provides an interrupt input source on GPIO3/Interrupt (pin 15). If the interrupt is triggered, the FT260 will generate an interrupt report with report ID 0xB1 via the interrupt IN pipe from the UART interface. The FT260 interrupt provides 4 trigger types: rising edge, falling edge, level-high, and level-low. By default, it is configured as level-high for 30ms. Note, that GPIO and interrupt are two different functions, which means if interrupt is the working function of pin 15, users cannot set and get GPIO3 status. Switching pin 15 to be either GPIO or interrupt and trigger settings can be done via EEPROM or USB commands.

The FT260 has flexible settings for suspend behaviour of all digital I/O pins. During suspend, these pins can perform pushing high, pushing low, tristate, or no-change. No-change means to keep the original function and value during suspend. The suspend behaviour can be configured in an external EEPROM.

6 Devices Characteristics and Ratings

6.1 Absolute Maximum Ratings

The absolute maximum ratings for the FT260 devices are as follows. These are in accordance with the Absolute Maximum Rating System (IEC 60134). Exceeding these may cause permanent damage to the device.

Parameter	Value	Unit	Conditions
Storage Temperature	-65°C to 150°C	Degrees C	
Floor Life (Out of Bag) At Factory Ambient (30°C / 60% Relative Humidity)	168 Hours (IPC/JEDEC J-STD-033A MSL Level 3 Compliant)*	Hours	
Ambient Operating Temperature (Power Applied)	-40°C to 85°C	Degrees C	
MTTF FT260	TBD	Hours	
VCCIN Supply Voltage	-0.3 to +5.5	V	
VCCIO IO Voltage	-0.3 to +4.0	V	
FSOURCE Supply Voltage	3.8±0.4	V	
DC Input Voltage – USBDP and USBDM	-0.5 to +3.63	V	
DC Input Voltage – High Impedance Bi-directional (powered from VCCIO)	-0.3 to +(VCCIO+0.5V)	V	
DC Output Voltage	3.3±0.15	V	VOUT3V3
DC Output Current – Outputs	100 **	mA	

Table 6.1 Absolute Maximum Ratings

* If devices are stored out of the packaging beyond this time limit the devices should be baked before use. The devices should be ramped up to a temperature of +125°C and baked for up to 17 hours.

** This DC output current on VOUT3V3 is also the power supply source for FT260 operation. If it must be the source for other components in the system, it can only supply 75mA or less.

6.2 ESD and Latch-up Specifications

Description	Specification
Human Body Mode (HBM)	> ± 2kV
Machine mode (MM)	> ± 200V
Charged Device Mode (CDM)	> ± 500V
Latch-up	> ± 200mA

Table 6.2 ESD and Latch-Up Specifications

6.3 DC Characteristics

DC Characteristics (Ambient Temperature = -40°C to +85°C)

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
VCC1	VCCIN Operating Supply Voltage	4.5	5	5.5	V	VCCIN is supplied with 5V
		2.97	3.3	3.63V	V	VCCIN is supplied with 3.3V
VCC2	VCCIO Operating Supply Voltage	2.97	3.3	3.63	V	VCCIO is supplied with 3.3V

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
		2.25	2.5	2.75	V	VCCIO is supplied with 2.5V
		1.62	1.8	1.98	V	VCCIO is supplied with 1.8V
I_{normal}	Operating Supply Current		9.6*		mA	Normal Operation at 12MHz
			14.5*		mA	Normal Operation at 24MHz
			23.6*		mA	Normal Operation at 48MHz
I_{idle}	Idle Supply Current		4.17		mA	IDLE Operation at 30KHz
I_{susp}	Suspend Supply Current		405		µA	USB Suspend
3V3	3.3v regulator output	2.97	3.3	3.63	V	VCCIN must be greater than 3V3 otherwise VOUT3V3 is an input which must be driven with 3.3V
$V_{FSOURCE}$	eFUSE Blowing Supply Voltage	3.4	3.8	4.2	V	

Table 6.3 Operating Voltage and Current

* The current measurement is with the regular data transferring speed between USB and I2C/UART interface. All the pins are with 4mA driving strength and without heavy loading.

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{oh}	Output Voltage High	2.97	VCCIO	VCCIO	V	Ioh = +/-2mA I/O Drive strength* = 4mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		2.97	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
V_{ol}	Output Voltage Low		0	0.4	V	Iol = +/-2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
V_{il}	Input low Switching Threshold			0.8	V	LVTTL

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{ih}	Input High Switching Threshold	2.0			V	LVTTL
V_t	Switching Threshold		1.49		V	LVTTL
V_{t-}	Schmitt trigger negative going threshold voltage		1.15		V	
V_{t+}	Schmitt trigger positive going threshold voltage		1.64		V	
R_{pu}	Input pull-up resistance	40	75	190	KΩ	Vin = 0
R_{pd}	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
I_{in}	Input Leakage Current	-10	+/-1	10	µA	Vin = 0
I_{oz}	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.4 I/O Pin Characteristics VCCIO = +3.3V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in eFUSE or external EEPROM

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{oh}	Output Voltage High	2.25	VCCIO	VCCIO	V	Ioh = +/-2mA I/O Drive strength* = 4mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		2.25	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
V_{ol}	Output Voltage Low		0	0.4	V	Iol = +/-2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
V_{il}	Input low Switching Threshold			0.8	V	LVTTL
V_{ih}	Input High Switching Threshold	1.7			V	LVTTL
V_t	Switching Threshold		1.1		V	LVTTL
V_{t-}	Schmitt trigger negative going threshold voltage		0.8		V	
V_{t+}	Schmitt trigger positive going threshold voltage		1.2		V	
R_{pu}	Input pull-up resistance	40	75	190	KΩ	Vin = 0
R_{pd}	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
I_{in}	Input Leakage	-10	+/-1	10	µA	Vin = 0

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
	Current					
I_{oz}	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.5 I/O Pin Characteristics VCCIO = +2.5V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in eFUSE or external EEPROM

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
V_{oh}	Output Voltage High	1.62	VCCIO	VCCIO	V	Ioh = +/-2mA I/O Drive strength* = 4mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 8mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 12mA
		1.62	VCCIO	VCCIO	V	I/O Drive strength* = 16mA
V_{ol}	Output Voltage Low		0	0.4	V	Iol = +/-2mA I/O Drive strength* = 4mA
			0	0.4	V	I/O Drive strength* = 8mA
			0	0.4	V	I/O Drive strength* = 12mA
			0	0.4	V	I/O Drive strength* = 16mA
V_{il}	Input low Switching Threshold			0.63	V	LVTTL
V_{ih}	Input High Switching Threshold	1.17			V	LVTTL
V_t	Switching Threshold		0.77		V	LVTTL
V_{t-}	Schmitt trigger negative going threshold voltage		0.557		V	
V_{t+}	Schmitt trigger positive going threshold voltage		0.893		V	
R_{pu}	Input pull-up resistance	40	75	190	KΩ	Vin = 0
R_{pd}	Input pull-down resistance	40	75	190	KΩ	Vin =VCCIO
I_{in}	Input Leakage Current	-10	+/-1	10	µA	Vin = 0
I_{oz}	Tri-state output leakage current	-10	+/-1	10	µA	Vin = 5.5V or 0

Table 6.6 I/O Pin Characteristics VCCIO = +1.8V (except USB PHY pins)

* The I/O drive strength and slow slew-rate are configurable in eFUSE or external EEPROM

6.4 USB Characteristics

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
DC Characteristics						
UR_o	Output impedance	3.5	7	14	Ω	

Parameter	Description	Minimum	Typical	Maximum	Units	Conditions
UR_{pu}	Internal Pull Up resistance	0.976	1.24	1.574	Ω	
UV_{OH}	High level Output on DP and DM	2.8			V	
UV_{OL}	Low level Output on DP and DM			0.3	V	
AC Characteristics						
UT_{rise}	Rise Time on DP/DM	4	10	20	ns	
UT_{fall}	Fall Time on DP/DM	4	10	20	ns	
UV_{cr}	Cross point	1.3		2.0	V	
UV_{th}	Single-ended receiver threshold	0.8		2.0	V	

Table 6.7 USB I/O Pin (DP, DM) Characteristics

7 FT260 Power Configurations

Section 7.1 to Section 7.4 illustrates possible USB power configurations for the FT260. Section 7.5 shows the configuration for system pins about DCNF0, DCNF1, XRESETN and other system pins. Section 7.6 shows the power for programming eFUSE.

7.1 USB Bus Powered Configuration with +3.3V I/O Voltage

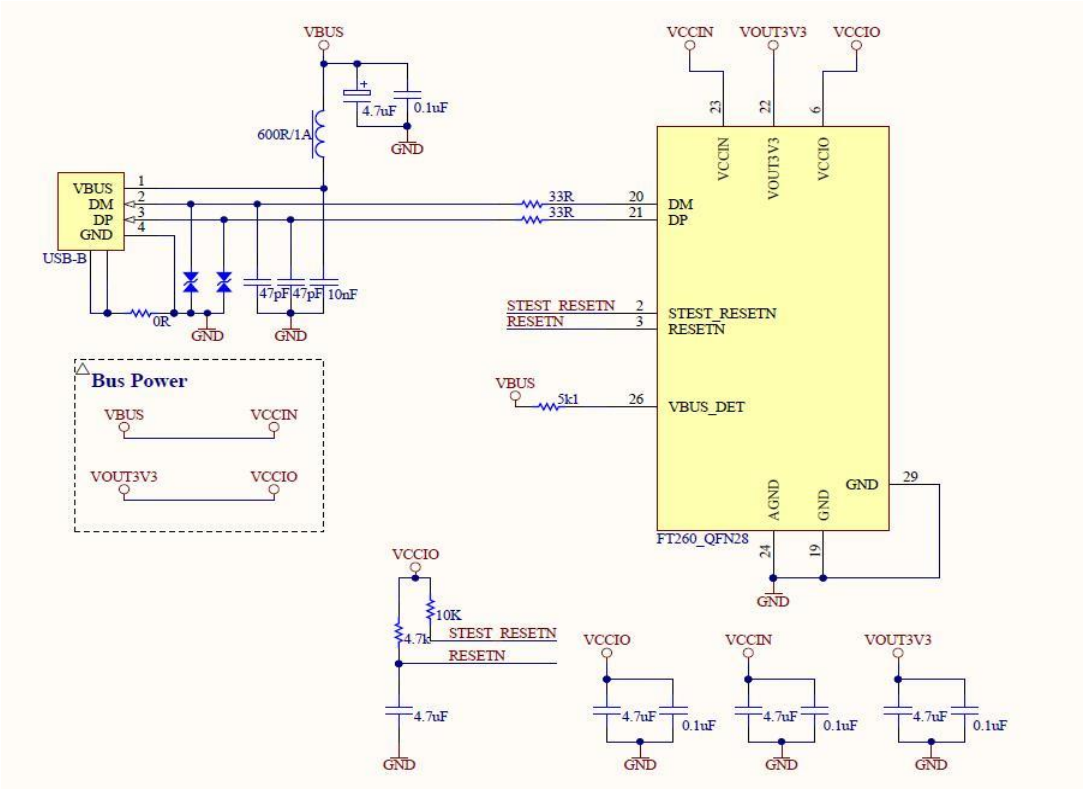


Figure 7-1 Bus Powered Configuration

Figure 7-1 illustrates the FT260 in a typical USB2.0 bus powered design configuration. A USB bus powered device gets its power from the USB bus. Basic rules for USB bus powered devices are as follows:

i) On plug-in to USB, the device should draw no more current than 100mA.
ii) In USB Suspend mode the device should draw no more than 2.5mA.
iii) A bus powered, high power USB device (one that draws more than 100mA) can use the SUSPOUT_N function on the pin DIO7 as a power disable function and use it to keep the current below 2.5mA on USB suspend.
iv) A device that consumes more than 100mA cannot be plugged into a USB bus powered hub.
v) No device can draw more than 500mA from the USB bus.

The VCCIN pin is the power source for the FT260 and can directly connect to VBUS. When the I/O voltage level is as +3.3V, users can directly connect VOUT3V3 to VCCIO without an external regulator. The VBUS_DET pin is a 5V-tolerant input pin and can directly connect to VBUS without an on-board voltage divider circuit. The power descriptors in the embedded eFUSE of the FT260 or in the external EEPROM should be programmed to match the current drawn by the device.

A ferrite bead is connected in series with the USB power supply to reduce EMI noise from the FT260 and associated circuitry being radiated down the USB cable to the USB host. The value of the Ferrite Bead depends on the total current drawn by the application. A suitable range of Ferrite Beads is available from Steward (www.steward.com), for example Laird Technologies Part # MI0805K400R-10.

7.2 Self Powered Configuration with 5V Source Input

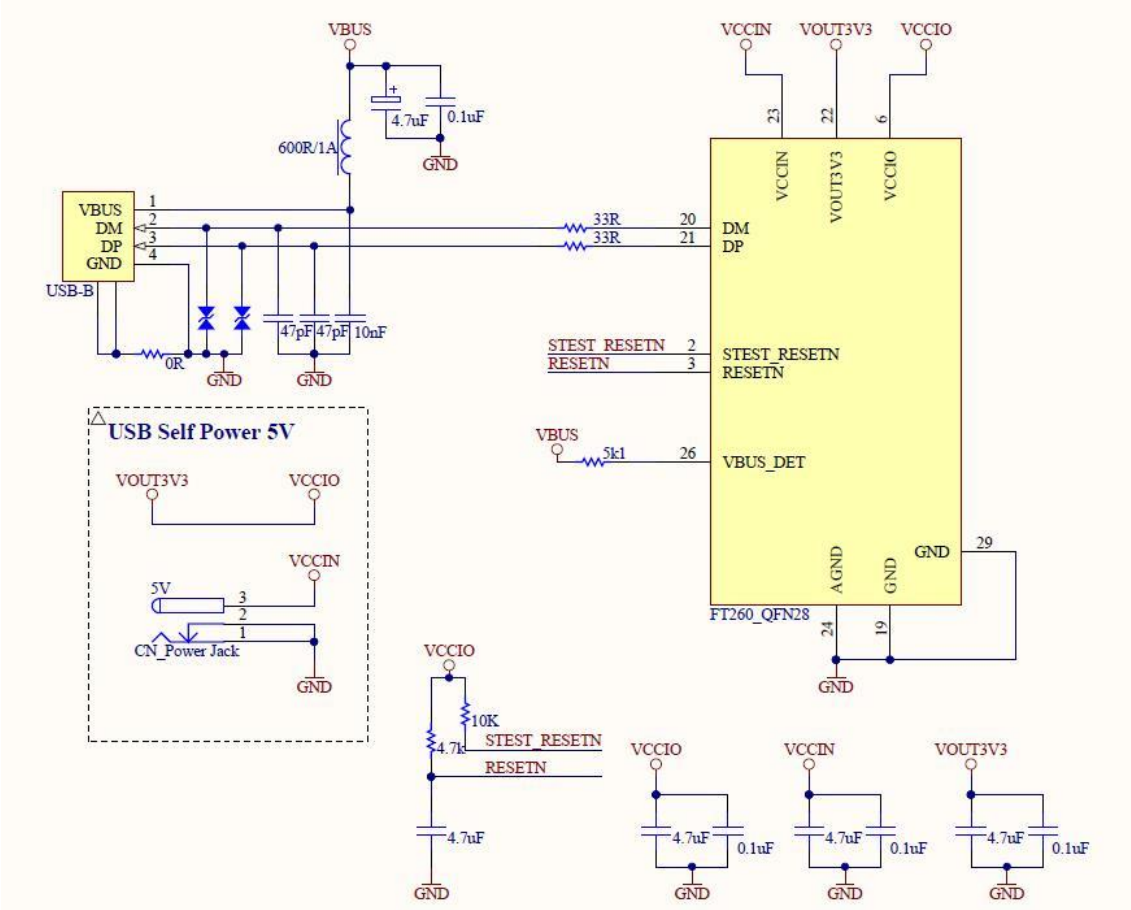


Figure 7-2 Self-Powered Configuration with 5V Source Input

Figure 7-2 illustrates the FT260 in a typical USB2.0 self-powered configuration. A USB self-powered device gets its power from its own power supply, 5V, and does not draw current from the USB bus. The basic rules for USB self-powered devices are as follows –

i) A self-powered device should not force current down the USB bus when the USB host or hub controller is powered down.
ii) A self-powered device can use as much current as it needs during normal operation and USB suspend as it has its own power supply.
iii) A self-powered device can be used with any USB host, a bus powered USB hub or a self-powered USB hub.

VCCIN is the power source for the FT260 and will source power from its own power supply. When the I/O voltage level is +3.3V, users can directly connect VOUT3V3 to VCCIO without an external regulator. The power descriptors in the embedded eFUSE of the FT260 or in the external EEPROM should be programmed as self-powered.

To comply with the first requirement above, the USB bus power (USB connector pin 1) is used to control the VBUS_DET pin of the FT260 device. VBUS_DET is a 5V-tolerant input pin and can directly connect to VBUS without an on-board voltage divider circuit. When the USB host or hub is powered up an internal 1.5kΩ resistor on DP is pulled up to +3.3V, thus identifying the device to the USB host or hub. When the USB host or hub is powered off, the VBUS_DET pin will be low and the FT260 is held in a suspend state. In this state the internal 1.5kΩ resistor is not pulled up to any power supply (hub or host is powered down), so no current flows down DP via the 1.5kΩ pull-up resistor. Failure to do this may cause some USB host or hub controllers to power up erratically.

7.3 Self Powered Configuration with 3.3V Source In

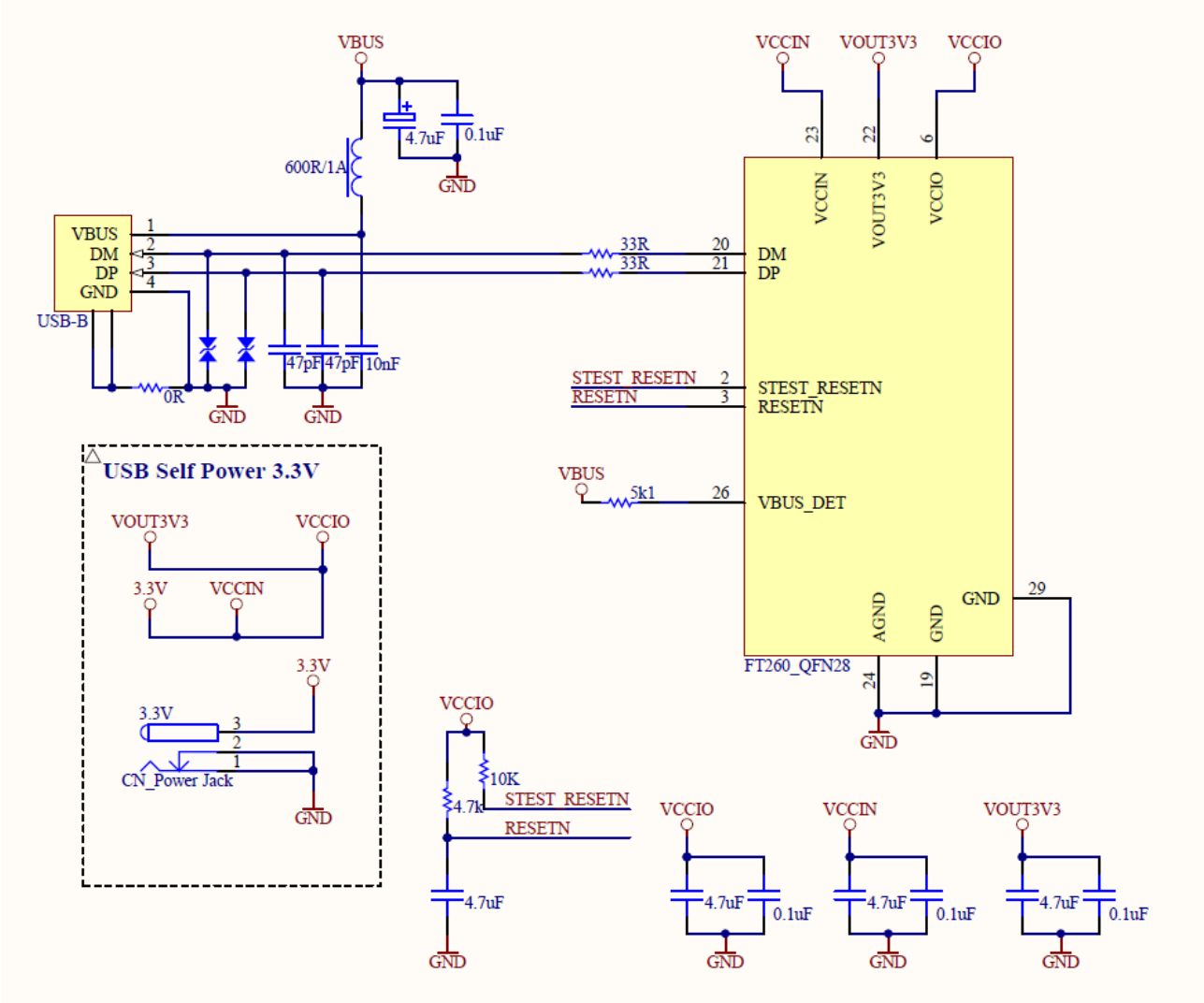


Figure 7-3 Self-Powered Configuration with 3.3V Source Input

Figure 7-3 illustrates the FT260 in a typical USB self-powered configuration like Figure 7-2. The difference here is that the self-power source is 3.3V. If using 3.3V as the power source in, remember to connect it to VOUT3V3 to supply an operating voltage for USB signalling in the FT260.

7.4 Bus Powered Configuration with +1.8V/+2.5V I/O Voltage

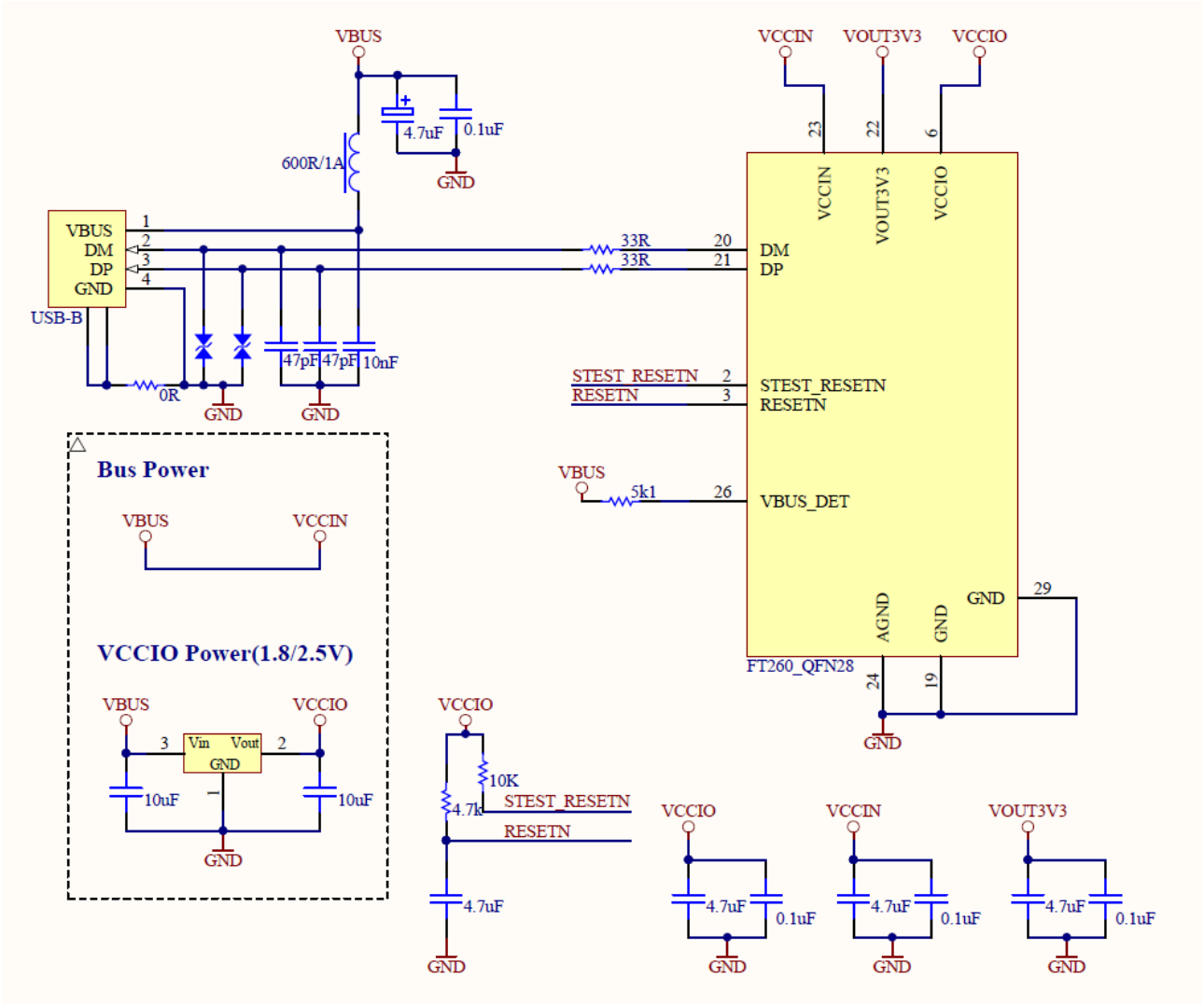


Figure 7-4 Bus Powered Configuration with +1.8V/+2.5V I/O voltage Level

Figure 7-4 illustrates the FT260 in a typical USB bus-powered configuration like Figure 7-1. The difference here is that the I/O pin voltage source is 2.5V or 1.8V, not 3.3V. An external regulator can source the power from VBUS and regulate out the required I/O voltage level. Then, VCCIO can connect to the output of the regulator to achieve I/O voltage level operating at +1.8V or +2.5V. VCCIN should be connected to VBUS to supply the power source for FT260 operation.

7.5 Self Powered Configuration with +1.8V/+2.5V I/O Voltage

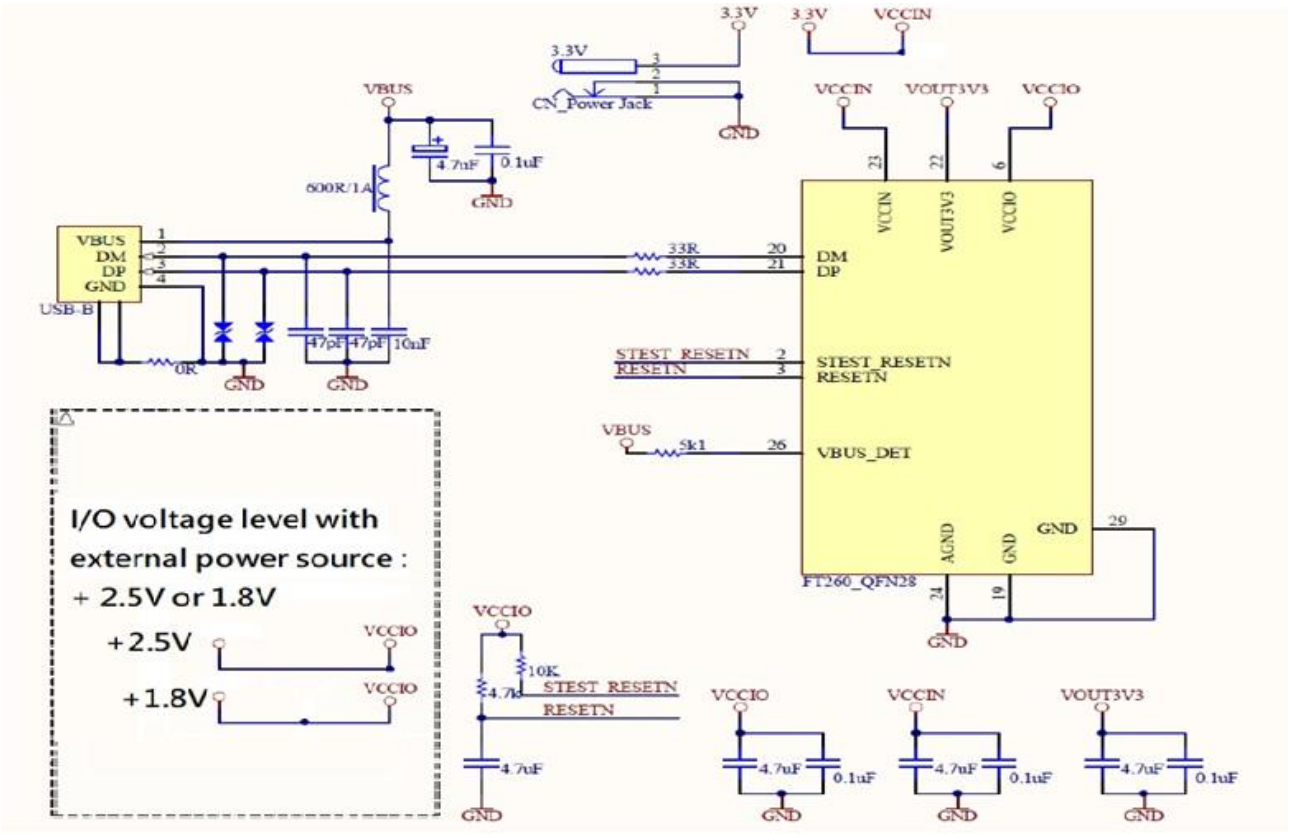


Figure 7-5 Self-Powered Configuration with +1.8V/+2.5V I/O voltage Level

Figure 7-5 illustrates the FT260 in a typical USB self-powered configuration. A USB self-powered device gets its power from its own power supply, 3.3V, and does not draw current from the USB bus. An external power source can provide +1.8V/+2.5V I/O voltage to VCCIN.

7.6 Configuration for System Pins

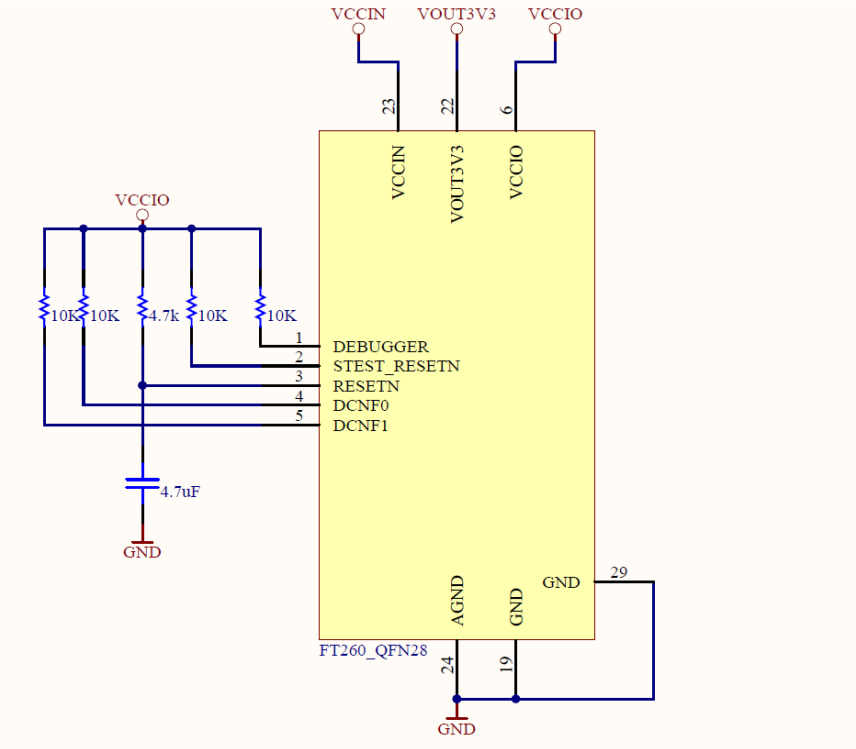


Figure 7-6 Recommended FT260 Configuration of System Pins

The pins, DCNF0 and DCNF1, will determine one of 3 configurations for the FT260 as defined in Section 5.1. These 2 pins have internal pull-down resistors; these 2 pins can be left floating for logic-0. If logic-1 is applied for DCNF0 and/or DCNF1, a 10kΩ resistor should be connected to VCCIO as shown in Figure 7-5.

The pin RESETN is the external reset source for the FT260. There is also a power-on-reset (POR) design in the FT260. If there is no requirement for an external reset, RESETN can be left floating or weakly tied to logic-high. If an external reset is required in the design, the related circuit in Figure 7-6 can be used for reference.

The DEBUGGER pin is reserved for debugging purposes and should be tied to VCCIO, the I/O power domain for the FT260. The pin, STEST_RESETN, is also a reserved pin and should be tied to logic-high.

Note that the GND pin located at pin-29 in Figure 7-5 is the paddle in the bottom side of the WQFN28 package. It should be tied together with the GND for FT260.

7.7 Power for Programming eFUSE

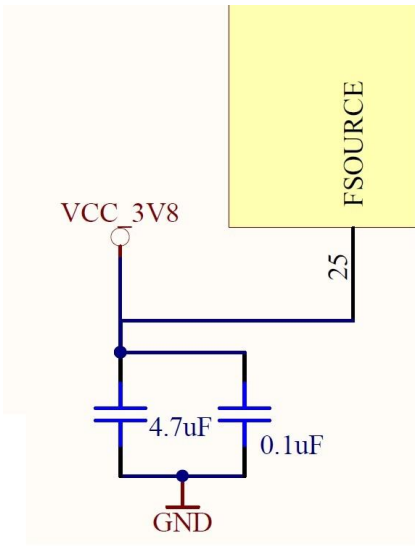


Figure 7-7 FSOURCE for eFUSE

When the FT260 is in normal operation without programming the eFUSE, the FSOURCE pin can be left floating. If the programming mechanism is required in the system, a power source with 3.8V should be applied. Figure 7-7 shows the related components for FSOURCE.

The programmer module, UMFTPD3A, which is developed by FTDI, can supply the power source for FSOURCE. With the programming utility FT_PROG, it can control the programming procedure and timing to the embedded eFUSE in the FT260. Users can easily set the vendor specifying parameters which are defined in eFUSE for customizing the FT260.

8 Application Examples

The following diagrams show the possible applications of the FT260. The illustrations have omitted the electrical design for the power domain plan. For power details refer to Section 7.

8.1 USB HID-over-I2C

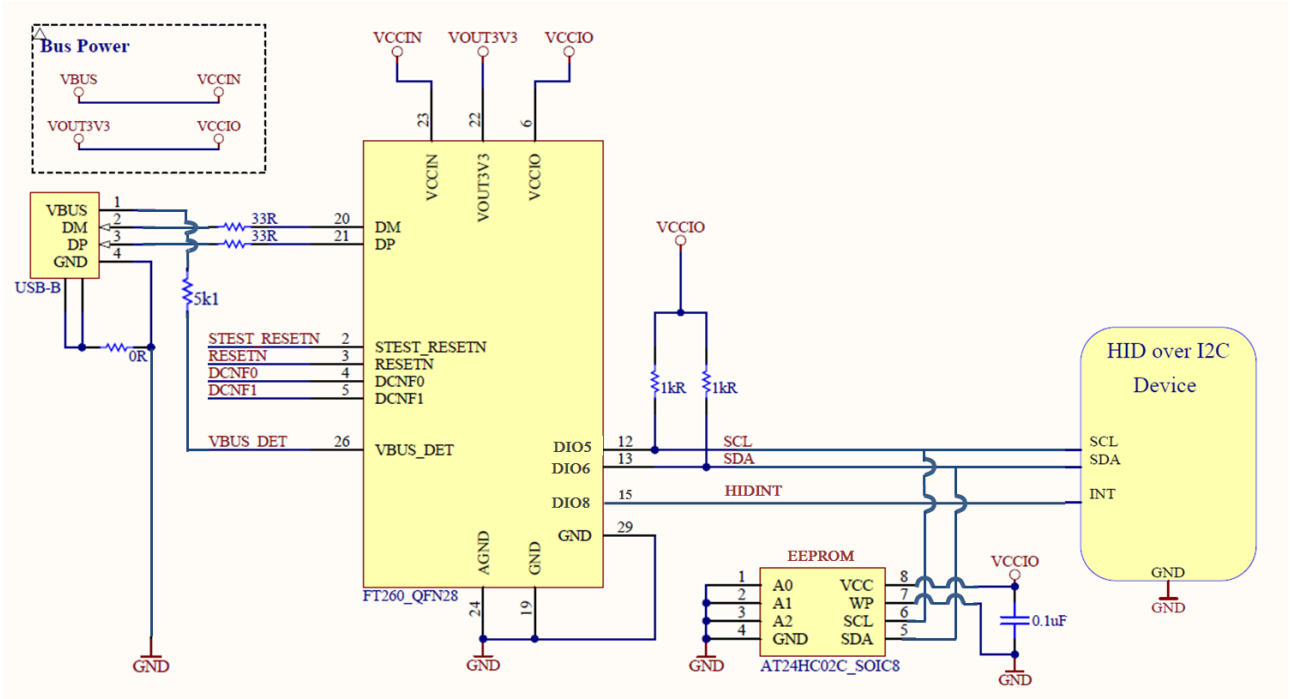


Figure 8-1 Application Example 1: HID over I²C

In Figure 8-1, a HID-over-I²C device can easily connect to USB by integrating the FT260 into the system. With an on-board EEPROM for customization, the FT260 can connect to both EEPROM and a HID class device with I²C slave interface simultaneously. DIO8 can be set as INTRIN, an interrupt input source from a HID class device for the requirement of the HID-over-I²C specification. With 1kΩ pull up resistors on SCL and SDA, the I²C bus can run at Standard mode.

Note: HID-over-I²C supported Report Descriptor length is up to 512 bytes.

8.2 USB to RS232 Converter

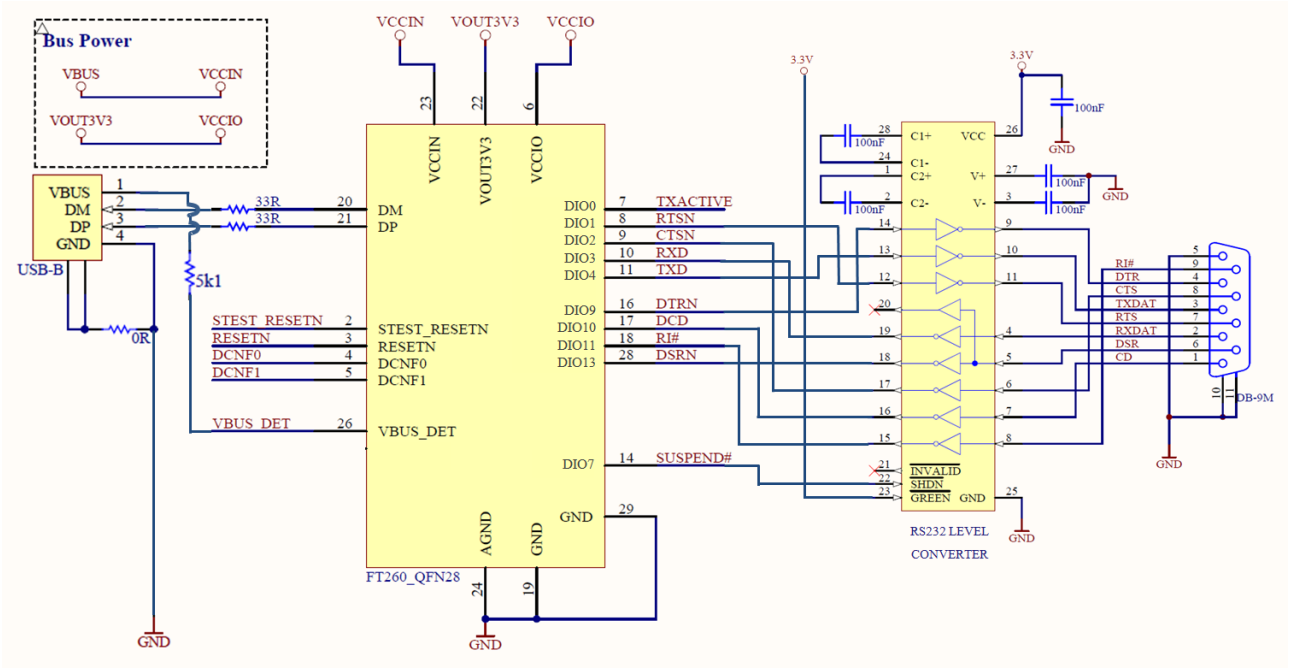


Figure 8-2 Application Example 2: USB to RS232 Converter

An example of using the FT260 as a USB to RS232 converter is illustrated in Figure 8-2. In this application, a TTL to RS232 Level Converter IC is used on the serial UART interface of the FT260 to convert the TTL levels of the FT260 to RS232 levels. This level shift can be done using the popular "213" series of TTL to RS232 level converters. These "213" devices typically have 4 transmitters and 5 receivers in a 28-LD SSOP package and feature an in-built voltage level converter to convert the +5V (nominal) VCC to the +/- 9 volts required by RS232.

The Shut Down control ($\overline{\text{SHDN}}$) signal of the converter device is the suspend control. The pin DIO7 of the FT260 can be set to function as SUSPOUT_N with active-low output and can be the control source to the converter IC. Note that the power source for the converter IC in Figure 8-2 is not supplied from the VOUT3V3 of the FT260 since the current consumption of this kind of converter is high. The supply current of VOUT3V3 is limited. Details can be referred to Table 6.1.

8.3 USB to RS485 Converter

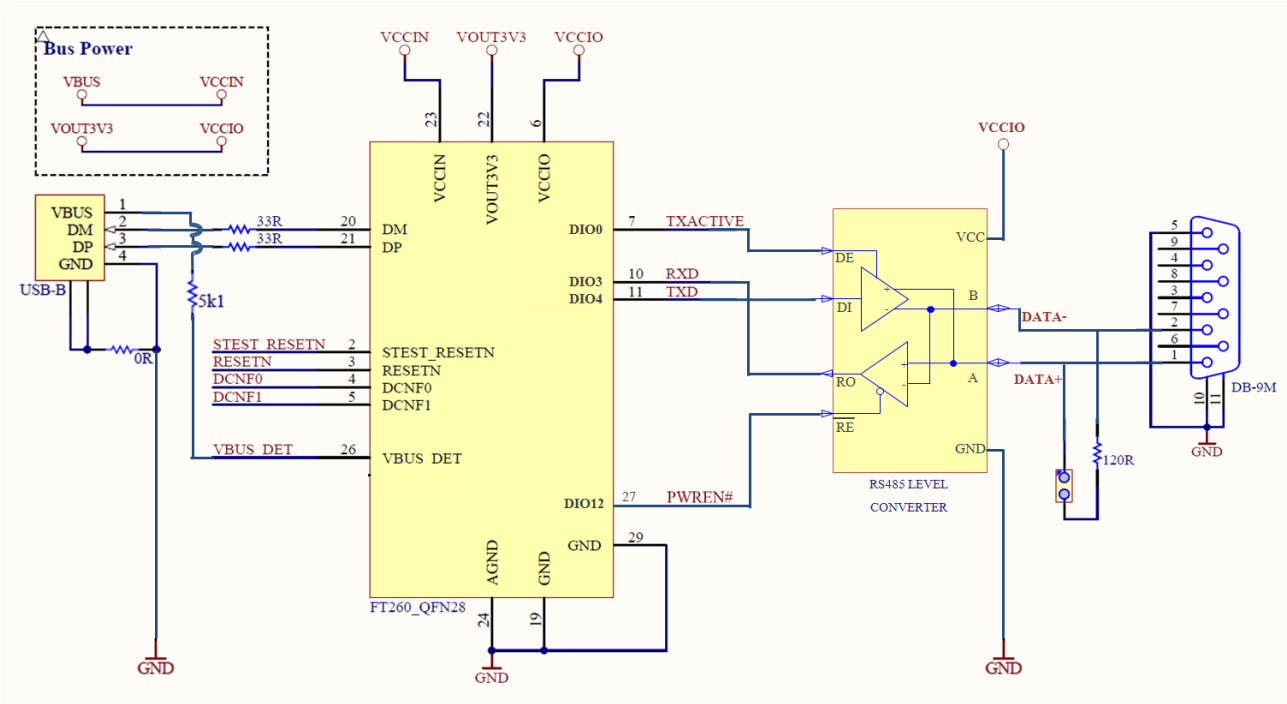


Figure 8-3 Application Example 3: USB to RS485 Converter

An example of using the FT260 as a USB to RS485 converter is shown in Figure 8-3. In this application, a TTL to RS485 level converter IC is used on the serial UART interface of the FT260 to convert the TTL levels of the FT260 to RS485 levels.

The converter device requires separate enable signals on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. Setting DIO0 as TX_ACTIVE is provided for the same purpose and wired to the transmitter enable (DE) of converter device. The pin DIO12 of FT260 can be configured as PWREN_N and wired to the receiver enable ($\overline{RE}$) of the converter device. With these configurations of the pins, the FT260 can be used as the USB to RS485 converter.

RS485 is a multi-drop network; so many devices can communicate with each other over a two-wire cable interface. The RS485 cable requires to be terminated at each end of the cable. A link (which provides the 120Ω termination) allows the cable to be terminated if the converter device is physically positioned at either end of the cable.

In this example, the data transmitted by the FT260 is also present on the receive path of the converter device. This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT260, it is possible to do this entirely in hardware by modifying the example shown in Figure 8.3 by logically OR-ing the FT260 TX_ACTIVE and the receiver output (RO) of converter device then connecting the output of the OR gate to the RXD of FT260.

Note that the TX_ACTIVE is activated 1 bit ahead the start bit. TX_ACTIVE is de-activated at the same time as the stop bit. This is not configurable.

8.4 USB to RS422 Converter

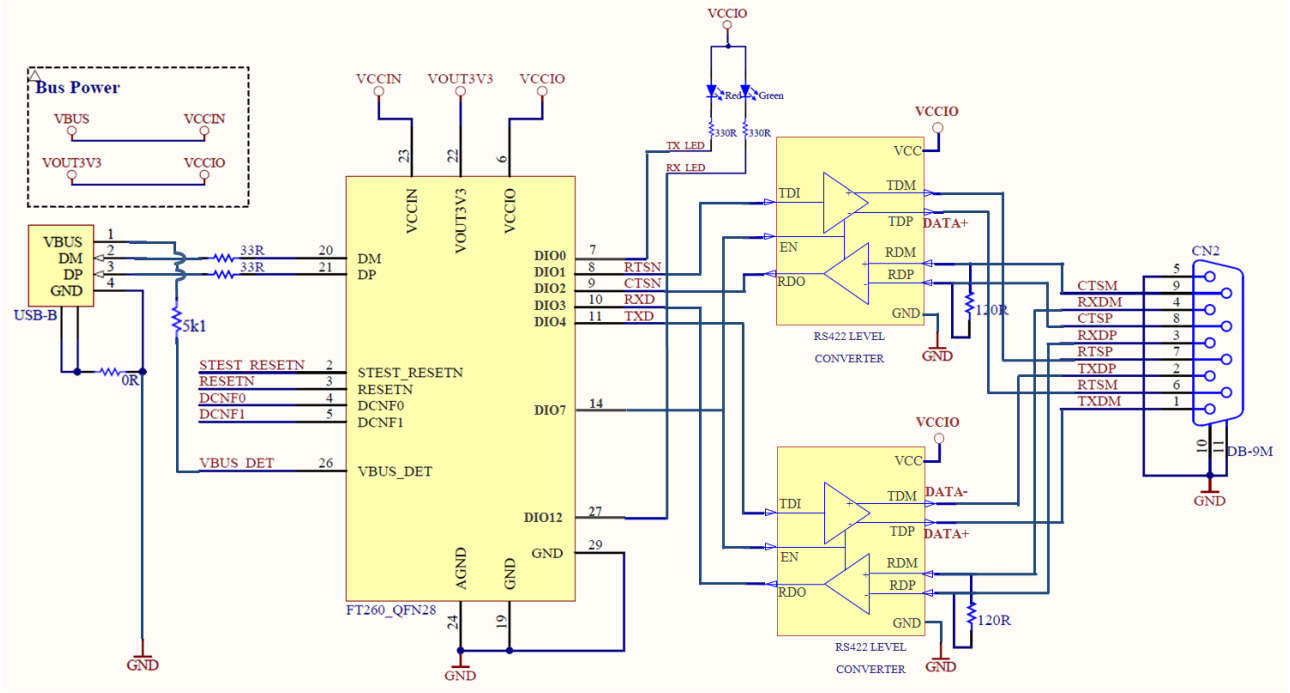


Figure 8-4 Application Example 4: USB to RS422 Converter

An example of using the FT260 as a USB to RS422 converter is shown in Figure 8-4. In this application, two TTL to RS422 Level Converter ICs are used on the serial UART interface of the FT260 to convert the TTL levels of the FT260 to RS422 levels.

There are many suitable level converter devices available. In Figure 8-4, the converter devices have an enable control (EN) to activate the data lines. Setting DIO7 as SUSPOUT_N allows the FT260 to disable the level converters when the system is in suspend. SUSPENDOUT_N is an active-low signal. The DIO0 and DIO12 of the FT260 can be configured as TX_LED and RX_LED. The FT260 can toggle LEDs with these signals when the UART interface is transmitting and receiving data.

9 User Configuration

The FT260 provides two storage paths for customization. One is eFUSE which is embedded in the FT260; another is the executable path to an external EEPROM. Parameters are defined for the customization and are categorized into several groups. The groups are USB-related, GPIO function selection, pin feature settings, UART settings and HID-over-I2C.

Embedded eFUSE provides a cost-effective customization. Without external devices, the FT260 can have customised VID/PID for USB, driving strength for digital pins, GPIOA, GPIOG, GPIO2 configurations and HID-over-I2C basic settings.

For customization demand, an EEPROM is required in the application. With this external EEPROM, more parameters are defined for customization. In this storage area, parameters in eFUSE are all included. USB string descriptor, pins status when USB suspending, and detailed pin configurations are included in EERPOM. Details can refer to Section 9.2.2.

There is already a default value for each parameter in the design of the FT260. When the parameters in eFUSE are programmed and enabled, the parameters in eFUSE will be loaded when the FT260 is powered up or reset.

Both eFUSE and EEPROM for the FT260 can be programmed over USB. This method is the same as for the MTP on other FTDI devices such as the FT-X series. Please note that to program eFUSE, the FT260 requires an additional programming voltage (3.8V) on its FSOURCE pin. The programming board, UMFTPD3A, supplies an easy connection bridge between the FT260 and a USB host for supplying the power source, for timing control of eFUSE, and for communicating with the programming utility FT_Prog. Further details may be found in the UMFTPD3A datasheet.

The FT_Prog utility is provided free-of-charge from the FTDI website, and can be found at the link below - https://ftdichip.com/utilities/#ft_prog.

9.1 Programming the embedded eFUSE over USB

The eFUSE in the FT260 can be programmed over USB. This method is the same as for the MTP on other FTDI devices such as the FT-X series. Note that to program eFUSE, the FT260 requires an additional programming voltage (3.8V) on FSOURCE (pin-25 @ WQFN28; pin-1 @ TSSOP28). The programming board, UMFTPD3A, can supply an easy connection bridge between the FT260 and a USB host for supplying the power source to FSOURCE and for communicating with the programming utility FT_Prog.

9.1.1 Default Values

The parameters defined in eFUSE are shown in Table 9.1. Default values are defined in the hardware design of the FT260. If the parameters defined in eFUSE are enabled, the settings in eFUSE will replace the default value.

Parameter	Default Value	Notes
USB Vendor ID (VID)	0403h	USB Vendor ID. Defined in the USB device descriptor. The format is 16-bit hex coded, and default is set as FTDI VID.
USB Product ID (PID)	6030h	USB Product ID. Defined in the USB device descriptor The format is 16-bit hex coded, and default is set as 6030h for FT260.
Power Source	Bus Powered	Define whether the power source is from the USB bus or a local source.
Max Bus Power Current	100mA	The max power that will be drawn from VBUS when using bus power. Range from 0~500mA. If the power source is defined as self-powered, it must be set as 0mA.
Remote Wake Up	Enable	Define if the FT260 supports remote wake up or not.
GPIO2 Function	SUSPOUT_N	The GPIO2 pin can be set as one of the alternative functions: - GPIO2 - PWREN# (low active), device ready indicator

Parameter	Default Value	Notes
		- TX_LED, UART TX transferring indicator - SUSPOUT_N, USB suspend low-active indicator.
GPIOA Function	TX_ACTIVE	The GPIOA pin can be set as one of the alternative functions: - TX_LED, UART TX transferring indicator - TX_ACTIVE
GPIOG Function	BCD_DET	The GPIOG pin can be set as one of the alternative functions: - GPIOG - PWREN# (low active), device ready indicator - RX_LED, UART RX receiving indicator - BCD_DET, Battery Charger Detection indicator
UART Drive Strength	4mA	Adjustable drive strength for UART related pins TXD/RXD, CTS/RTS, DTR/DSR, TX_ACTIVE. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
GPIO Drive Strength	4mA	Adjustable drive strength for GPIO related pins GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, and GPIO5. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
BCD_DET Function Disable?	No	Battery Charger Detection function can be disabled on BCD_DET pin.
BCD_DET Drive Strength	4mA	Adjustable drive strength for BCD_DET pin. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
BCD_DET Polarity	Active-high	Set the polarity on BCD_DET pin for indicating battery charge detected. Default is set as active-high.
Power Saving Mode	Enable	If power saving mode is enabled and the FT260 is idle for 5 seconds, it will switch the system clock to 30KHz for saving power.
HID over I²C address	0h	The I²C slave address of the target HID-over-I²C device. The address 0h means no HID-over-I²C device connected.
HID over I²C Descriptor Address	0h	The start address of the descriptor of the target HID-over-I²C device.
HID over I²C Interrupt	Rising Edge	Define the interrupt trigger type of the target HID-over-I²C device. It can be rising edge, falling edge, level-high, level-low.
HID over I²C Option	Supported	According to Microsoft HID over I²C Protocol Specification, the following requests are optional: - GET_IDLE/SET_IDLE supported or not? - GET_PROTOCOL/SET_PROTOCOL supported or not? SET_POWER supported or not?

Table 9.1 Parameters defined in internal eFUSE for FT260

9.2 Programming the external EEPROM over USB

The external EEPROM can be programmed over USB, which is supported by FT_PROG as most FTDI chips do.

The FT_Prog utility is provided free-of-charge from the FTDI website, and can be found at the link below. The user guide is also available at this link.
https://ftdichip.com/utilities/#ft_prog

Please note that a user needs to specify the data address type as one-byte or two-byte when programming the external EEPROM.

9.2.1 Supported EEPROM Spec

The FT260 supports an external EEPROM with I²C interface, slave address 0x50~0x57, and data size larger than 256 bytes. When the FT260 powers on, it will scan the I²C bus and try to find if an external EEPROM is present. If it is present, it will check the content and load the configuration data from the EEPROM into the FT260. Note that loading data from an external EEPROM is the last step of power-on, and it will overwrite the configuration data from eFUSE.

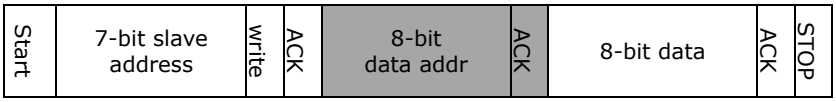
There are two different types of EEPROM. One has one-byte data address; the other one has two-byte data address. Usually, the first one has a data size smaller than 256bytes, and the second one has a data size more than 256bytes. It is not possible for the FT260 to automatically identify the EEPROM type, therefore, FT_Prog will require the user's input to specify the type of the external EEPROM.

In summary, the supported external EEPROM has:

- I²C interface with slave address 0x50~0x57
- 256 bytes at least
- One-byte data address or two-byte data address

The protocol of one-byte data address EEPROM

Write:



Read:

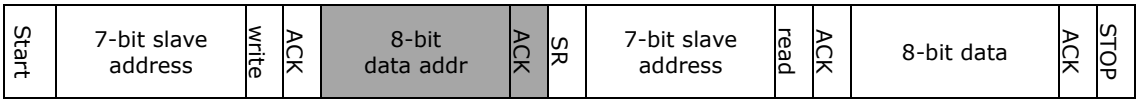


Figure 9-1 Protocol Format for EEPROM with One Byte Data Address

The protocol of two-byte data address EEPROM

Write:

Read:

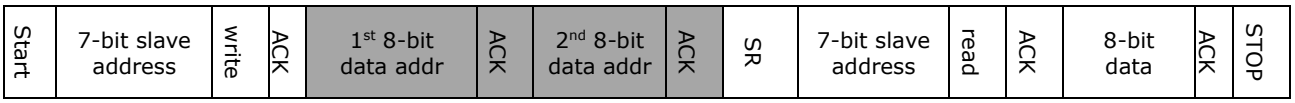


Figure 9-2 Protocol Format for EEPROM with Two Bytes Data Address

Table 9.2 shows the list of the external EEPROMs that are tested with the FT260.

Manufacturer	Part Number	Type
Atmel	AT24CS04-SSHM-TCT-ND 512 x 8	One-byte data address
Atmel	AT24C08D-PUM-ND 1K x 8	One-byte data address
Atmel	AT24CS16-SSHM-TCT-ND 2K x 8	One-byte data address
Atmel	AT24CM01 (1M bit)	Two-byte data address
On Semiconductor	CAT24C04WI-G-ND 512 x 8	One-byte data address
On Semiconductor	CAT24C08WI-GT3CT-ND 1k x 8	One-byte data address
On Semiconductor	CAT24C08WI-GT3CT-ND 2k x 8	One-byte data address
On Semiconductor	CAT24C512	Two-byte data address
Microchip	24AA08-I/SN-ND 1k x 8	One-byte data address
Microchip	24AA16-I/SN-ND 2k x 8	One-byte data address

Table 9.2 Tested EEPROM List for FT260

9.2.2 Default Values

The parameters defined in the EEPROM are shown in Table 9.3. Default values are defined in the FT260. If an EEPROM exists in the application, the priority of the EEPROM is higher than default values.

Parameter	Default Value	Notes
Device Type	FT260	Read-Only. Indicate the Chip is FT260.
USB Vendor ID (VID)	0403h	USB Vendor ID. Defined in the USB device descriptor. The format is 16-bit hex coded, and default is set as FTDI VID.
USB Product ID (PID)	6030h	USB Product ID. Defined in the USB device descriptor The format is 16-bit hex coded, and default is set as 6030h for FT260.
USB Version	0200h	Read-only. Returns the USB 2.0 device descriptor to the host. Note: FT260 is a Full-speed USB2.0 device.
Power Source	Bus Powered	Define whether the power source is from the USB bus or a local source.
Max Bus Power Current	100mA	The max power that will be drawn from VBUS when using bus power. Range from 0~500mA. If the power source is defined as self-powered, it must be set as 0mA.
Remote Wake Up	Enable	Define if the FT260 supports remote wake up or not.
Manufacturer Name	FTDI	Describing the manufacturer. A string descriptor defined in USB device descriptors
Product Description	FT260	Describing the product. A string descriptor defined in USB device descriptors
Serial Number Enabled?	No	Enable the string descriptor for serial number or not.
Serial Number	None	A unique serial number is generated and programmed into the EEPROM. Refer to the Utility FT_Prog for details.
Suspend Out Polarity	Active-low	Set the polarity on GPIO2 pin for indicating suspend out. Default is set as active-low.
RI as Wake-Up	Disable	UART RI can be the source to remote wakeup the USB host when this remote wake up is allowed. - Disable (default) - Enable
RI Wake-Up Config	Falling Edge	Specify the criteria for RI to trigger a remote wake-up. - Falling Edge(default), RI from Logic-High to Low

Parameter	Default Value	Notes
		- Rising Edge, RI from Logic-Low to High
GPIO2 Function	SUSPOUT_N	The GPIO2 pin can be set as one of the alternative functions: - GPIO2 - PWREN# (low active), device ready indicator - TX_LED, UART TX transferring indicator - SUSPOUT_N/SUSPOUT, USB suspend indicator
GPIOA Function	TX_ACTIVE	The GPIOA pin can be set as one of the alternative functions: - GPIOA - TX_LED, UART TX transferring indicator - TX_ACTIVE
GPIOG Function	BCD_DET	The GPIOG pin can be set as one of the alternative functions: - GPIOG - PWREN# (low active), device ready indicator - RX_LED, UART RX receiving indicator - BCD_DET, Battery Charger Detection indicator
UART Drive Strength	4mA	Adjustable drive strength for UART related pins TXD/RXD, CTS/RTS, DTR/DSR, TX_ACTIVE. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
UART Slew Rate Enable?	Disable	Set the slew rate control for UART related pins TXD/RXD, CTS/RTS, DTR/DSR, TX_ACTIVE. Default is disabled
GPIO Drive Strength	4mA	Adjustable drive strength for GPIO related pins GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, and GPIO5. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
GPIO Weak Pullup/Pulldown	Disable	Enable the weak pullup / pulldown resistor on the pins GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, and GPIO5. Default is disabled (without any pull).
GPIO Slew Rate Enable?	Disable	Set the slew rate control for GPIO related pins GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, and GPIO5. Default is disabled
BCD_DET Function Disable?	No	Battery Charger Detection function can be disabled on BCD_DET pin.
BCD_DET Drive Strength	4mA	Adjustable drive strength for BCD_DET pin. Drive strength can be set as 4mA, 8mA, 12mA and 16mA
BCD_DET Polarity	Active-high	Set the polarity on BCD_DET pin for indicating battery charge detected. Default is set as active-high.
Interrupt Trigger	Level-high	Define the interrupt trigger type when GPIO3 is set as INTR/WAKEUP function. The possible settings are rising edge, falling edge, level-high, level-low.
Trigger Level Setting	30ms	Interrupt level width select. When the interrupt is set to level trigger and it exceeds the specified level width, the interrupt signal will be generated. The level width can be set as 1ms, 5ms and 30ms.
Power Saving Mode	Enable	If power saving mode is enabled and the FT260 is idle for 5 seconds, it will switch the system clock to 30KHz for saving power.
Pin Status During Suspend	No change	During suspend, each digital I/O pin of the FT260 can be set as - No change, keep the original pin function - tristate - push low when suspend - push high when suspend
HID-over-I²C address	0h	The I²C slave address of the target HID-over-I²C device. The address 0h means no HID-over-I²C device connected.
HID-over-I²C Descriptor Address	0h	The start address of the descriptor of the target HID-over-I²C device.

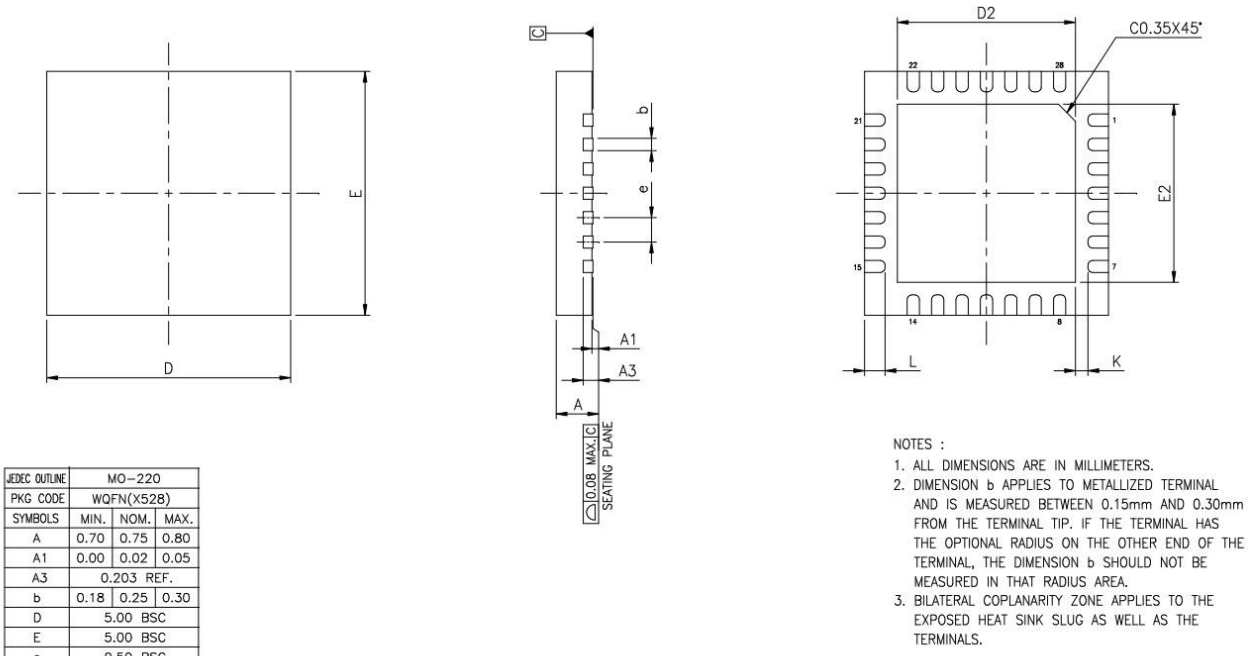
Parameter	Default Value	Notes
HID-over-I²C Interrupt	Rising Edge	Define the interrupt trigger type of the target HID-over-I²C device. It can be rising edge, falling edge, level-high, level-low.
HID-over-I²C Option	Not supported	According to Microsoft HID over I²C Protocol Specification, the following requests are optional: - GET_IDLE/ SET_IDLE supported or not - GET_PROTOCOL/ SET_PROTOCOL supported or not - SET_POWER supported or not
HID-over-I²C Subclass	No Subclass	The HID subclass description. It can be: - No Subclass - Boot Interface Subclass
HID-over-I²C Protocol	None	The HID protocol code description. It can be: - None - Keyboard - Mouse

Table 9.3 Parameters defined in external EEPROM for FT260

10 Package Parameters

The FT260 is available in WQFN-28 and TSSOP-28 package. The solder reflow profile for WQFN-28 and TSSOP-28 is described in Section 10.5.

10.1 WQFN-28 Package Mechanical Dimensions



JEDEC OUTLINE	MO-220		
PKG CODE	WQFN(X528)		
SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.18	0.25	0.30
D	5.00 BSC		
E	5.00 BSC		
e	0.50 BSC		
K	0.20	—	—

PAD SIZE	D2			E2			L			LEAD FINISH		JEDEC CODE
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
138X13* MIL	3.20	3.25	3.30	3.20	3.25	3.30	0.50	0.55	0.60	V	X	W(V)HHD-1

Figure 10-1 WQFN-28 Package Dimensions

The FT260Q is supplied in a RoHS2.0 compliant leadless WQFN-28 package. The package is lead (Pb) free and uses a 'green' compound. The package is fully compliant with European Union REACH 2006/121/EC.

This package is nominally 5.00mm x 5.00mm. The solder pads are on a 0.5mm pitch. The above mechanical drawing shows the WQFN-28 package. All dimensions are in millimetres.

The centre pad on the base of the FT260Q is internally connected to GND and the PCB should not have signal tracking on the top layer under this area. Connect to GND.

10.2 WQFN-28 Package Markings

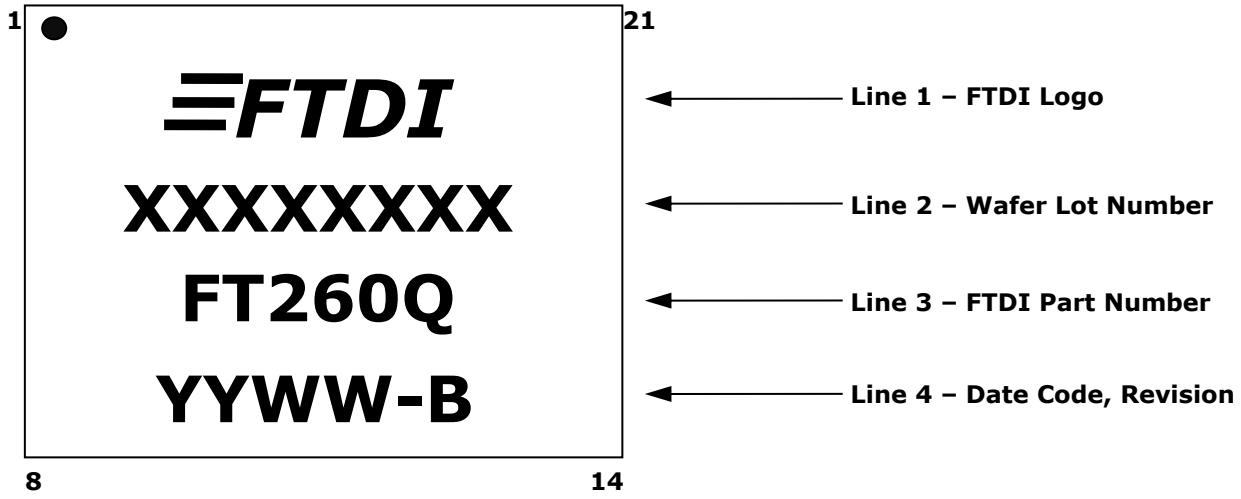
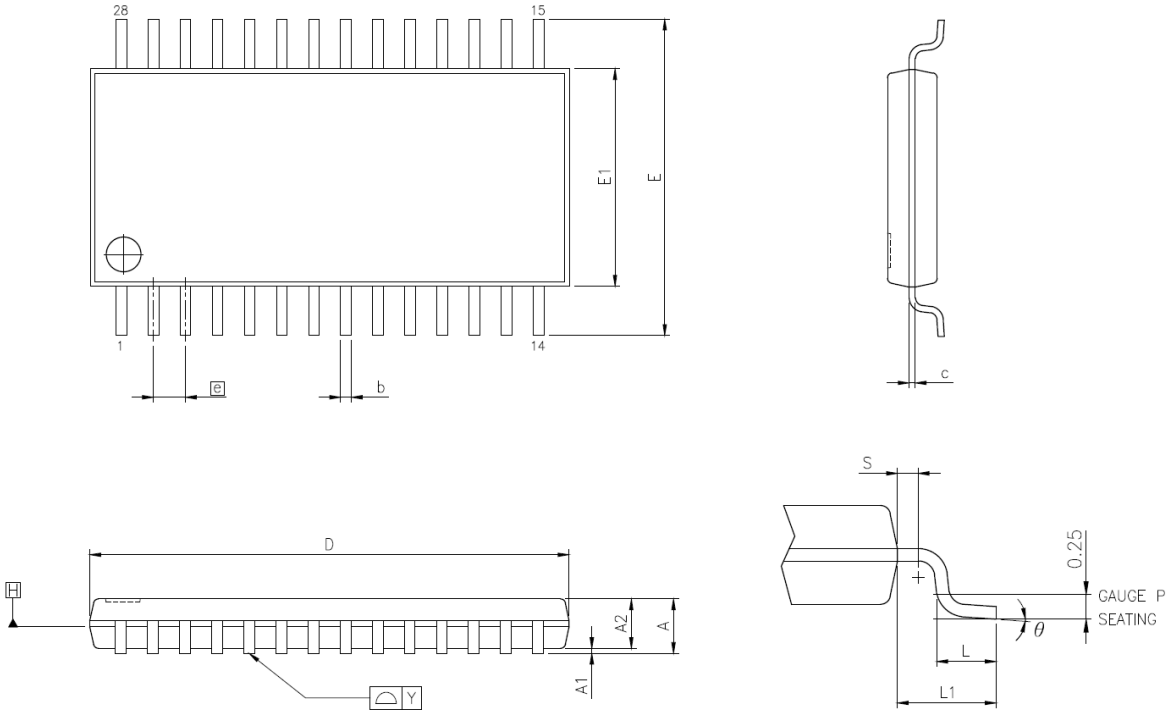


Figure 10-2 WQFN-28 Package Markings

The date code format is **YYWW** where WW = 2-digit week number, YY = 2-digit year number. This is followed by the revision number.

The code **XXXXXXXX** is the manufacturing LOT code.

10.3 TSSOP-28 Package Mechanical Dimensions



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	–	–	1.20
A1	0.00	–	0.15
A2	0.80	1.00	1.05
b	0.19	–	0.30
c	0.09	–	0.20
D	9.60	9.70	9.80
E1	4.30	4.40	4.50
E	6.40 BSC		
e	0.65 BSC		
L1	1.00 REF		
L	0.45	0.60	0.75
S	0.20	–	–
θ	0°	–	8°
Y	0.10		

Figure 10-3 TSSOP-28 Package Dimensions

The FT260S is supplied in a RoHS2.0 compliant leadless TSSOP-28 package. The package is lead (Pb) free and uses a 'green' compound. The package is fully compliant with European Union REACH 2006/121/EC.

This package is nominally 9.70mm x 4.40mm. The above mechanical drawing shows the TSSOP-28 package. All dimensions are in millimetres.

10.4 TSSOP-28 Package Markings

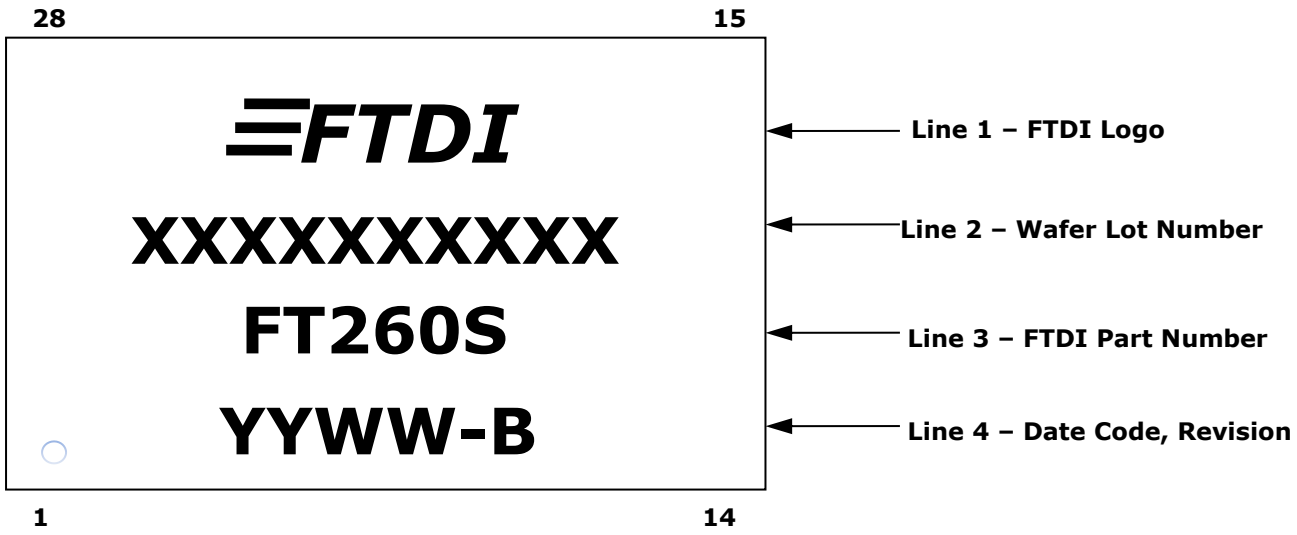


Figure 10-4 TSSOP-28 Package Markings

The date code format is **YYWW** where WW = 2-digit week number, YY = 2-digit year number. This is followed by the revision number.

The code **XXXXXXXX** is the manufacturing LOT code.

10.5 Solder Reflow Profile

The FT260 is supplied in a Pb free WQFN-28 and TSSOP-28 package. The recommended solder reflow profile is shown in Figure 10-3.

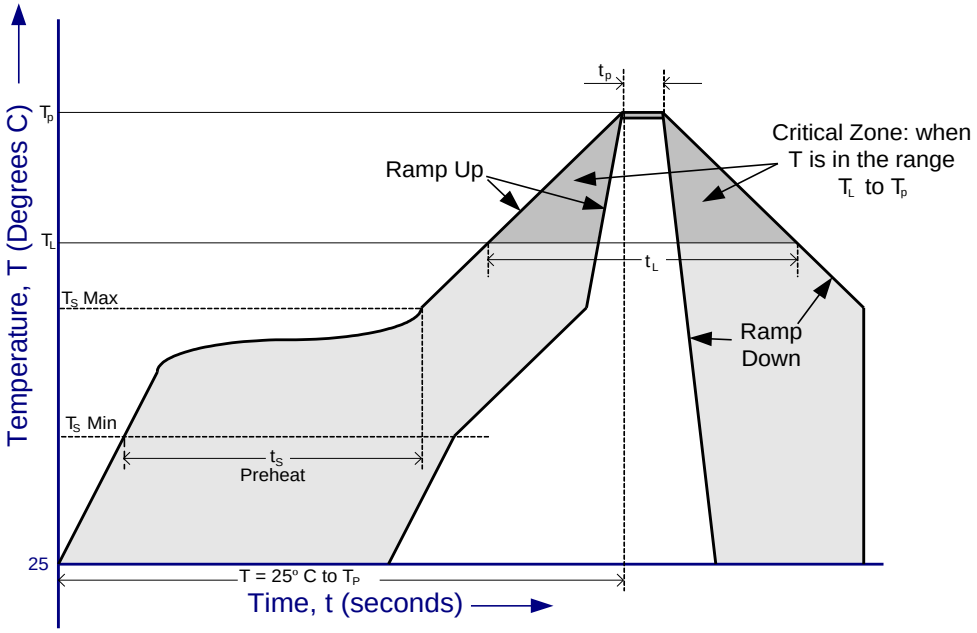


Figure 10-5 FT260 Solder Reflow Profile

The recommended values for the solder reflow profile are detailed in Table 10.1. Values are shown for both a completely Pb free solder process (i.e., the FT260 is used with Pb free solder), and for a non-Pb free solder process (i.e., the FT260 is used with non-Pb free solder).

Profile Feature	Pb Free Solder Process	Non-Pb Free Solder Process
Average Ramp Up Rate (T_s to T_p)	3°C / second Max.	3°C / Second Max.
Preheat - Temperature Min (T_s Min.) - Temperature Max (T_s Max.) - Time (t_s Min to t_s Max)	 150°C 200°C 60 to 120 seconds	 100°C 150°C 60 to 120 seconds
Time Maintained Above Critical Temperature T_L: - Temperature (T_L) - Time (t_L)	 217°C 60 to 150 seconds	 183°C 60 to 150 seconds
Peak Temperature (T_p)	260°C	240°C
Time within 5°C of actual Peak Temperature (t_p)	20 to 40 seconds	20 to 40 seconds
Ramp Down Rate	6°C / second Max.	6°C / second Max.
Time for T= 25°C to Peak Temperature, T_p	8 minutes Max.	6 minutes Max.

Table 10.1 Reflow Profile Parameter Values

11 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited (UK)
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-mail (Sales) us.sales@ftdichip.com
E-mail (Support) us.support@ftdichip.com
E-mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 (21) 62351596
Fax: +86 (21) 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

http://ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – References

Document References

Application Notes

AN_394 User Guide for FT260
AN_395 User Guide for LibFT260
AN_444 Using the FT260 with Linux
AN_124 User Guide for FTDI FT_Prog Utility
AN_184 FTDI Device Input Output Pin States
AN_175 Battery Charger Detection Over USB with FT-X Devices

Technical Notes

TN_100 USB Vendor ID/Product ID Guidelines
TN_111 What is UART

Datasheets

DS_UMFT260EV
UMFTPD3A Program Module Datasheet

FT_PROG Utility

https://ftdichip.com/utilities/#ft_prog

Related Document or Specification

http://i2c2p.twibright.com/spec/i2c.pdf
https://msdn.microsoft.com/en-us/library/windows/hardware/dn642101(v=vs.85).aspx
https://msdn.microsoft.com/en-US/library/jj131705(v=vs.85).aspx

Acronyms and Abbreviations

Terms	Description
API	Application Programming Interface
CTS	Clear To Send
DCD	Direct Carrier Detect
DLL	Dynamic Link Library
DSR	Data Set Ready
DTR	Data Terminal Ready
EEPROM	Electrically Erasable Programmable Read Only Memory
GPIO	General Purpose Input Output
HID	Human Interface Device
I2C	Inter-Integrated Circuit
LDO	Low Drop Out regulator
LED	Light-emitting diode
POR	Power-On-Reset
RTS	Request To Send
SIE	Serial Interface Engine
SSOP	Shrink Small Outline Package
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transmitter

Appendix B - List of Figures and Tables

List of Figures

List of Tables

Appendix C - Revision History

Document Title: FT260 HID-class USB to UART/I2C Bridge IC

Document Reference No.: FT_001272

Clearance No.: FTDI#484

Product Page: https://ftdichip.com/product-category/products/ic

Document Feedback: Send Feedback

Revision	Changes	Date
Version 1.0	Initial Release	01-02-2016
Version 1.1	Updated to include FT260S information	23-05-2016
Version 1.2	Updated DIO8 to GPIO3 on page 10 Added important notes under table 3.1	21-03-2022
Version 1.3	Added reference t Linux HID-FT260 driver for I2C functionality	01-08-2023
Version 1.4	Added TX_LED to DIO7 in Table 3.1. Added number of bytes to HID-over-I^2C Report Descriptor length in section 8.1. Added Note 4 to Table 3.1.	21-11-2023