Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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 1.2 Kbuad to 12 Mbaud (RS422, RS485, RS232) at TTL levels
- Configurable GPIOs can be easily controlled by software applications under HID class via the USB bus
- 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 385µA (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; 4 mA(min) and 16 mA(max)
- Integrated power-on-reset circuit.
- USB Battery Charger Detection.
- UHCI / OHCI / EHCI / XHCI host controller compatible.
- Extended operating temperature range; -40 to 85°C.
- Available in compact Pb-free 28 Pin WQFN or TSSOP packages (RoHS compliant).

1 Typical Applications

- HID class Device controller
- USB to HID-over-I2C Bridge
- USB to I^2C 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 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 is:

- 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

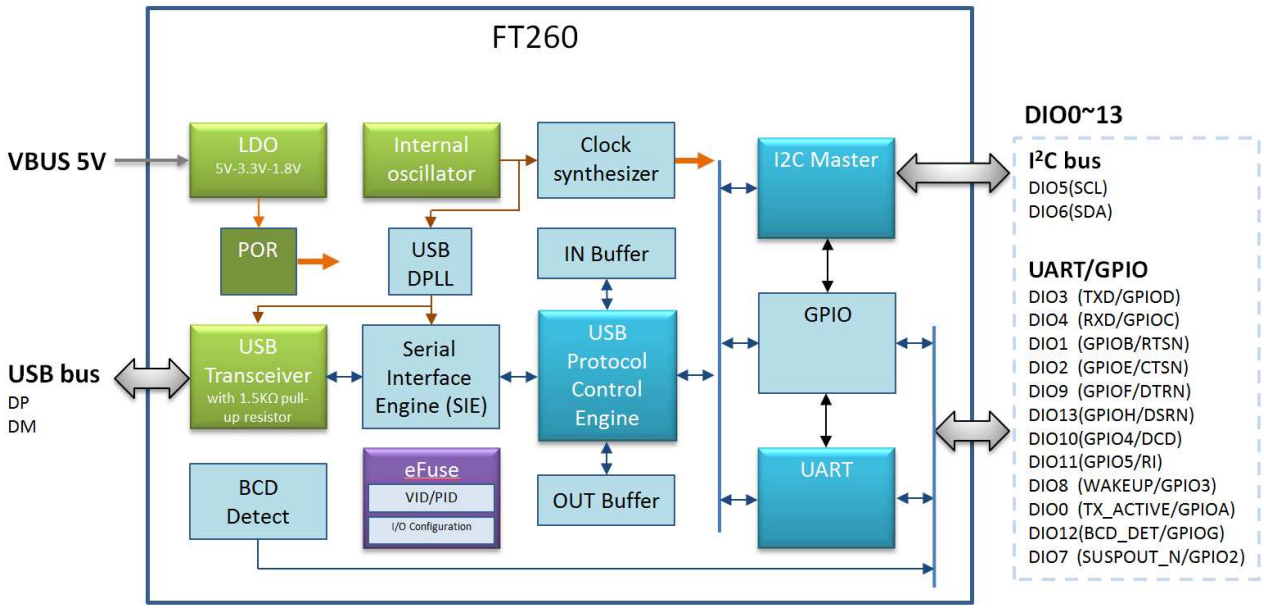

Figure 2.1 FT260 Block Diagram

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

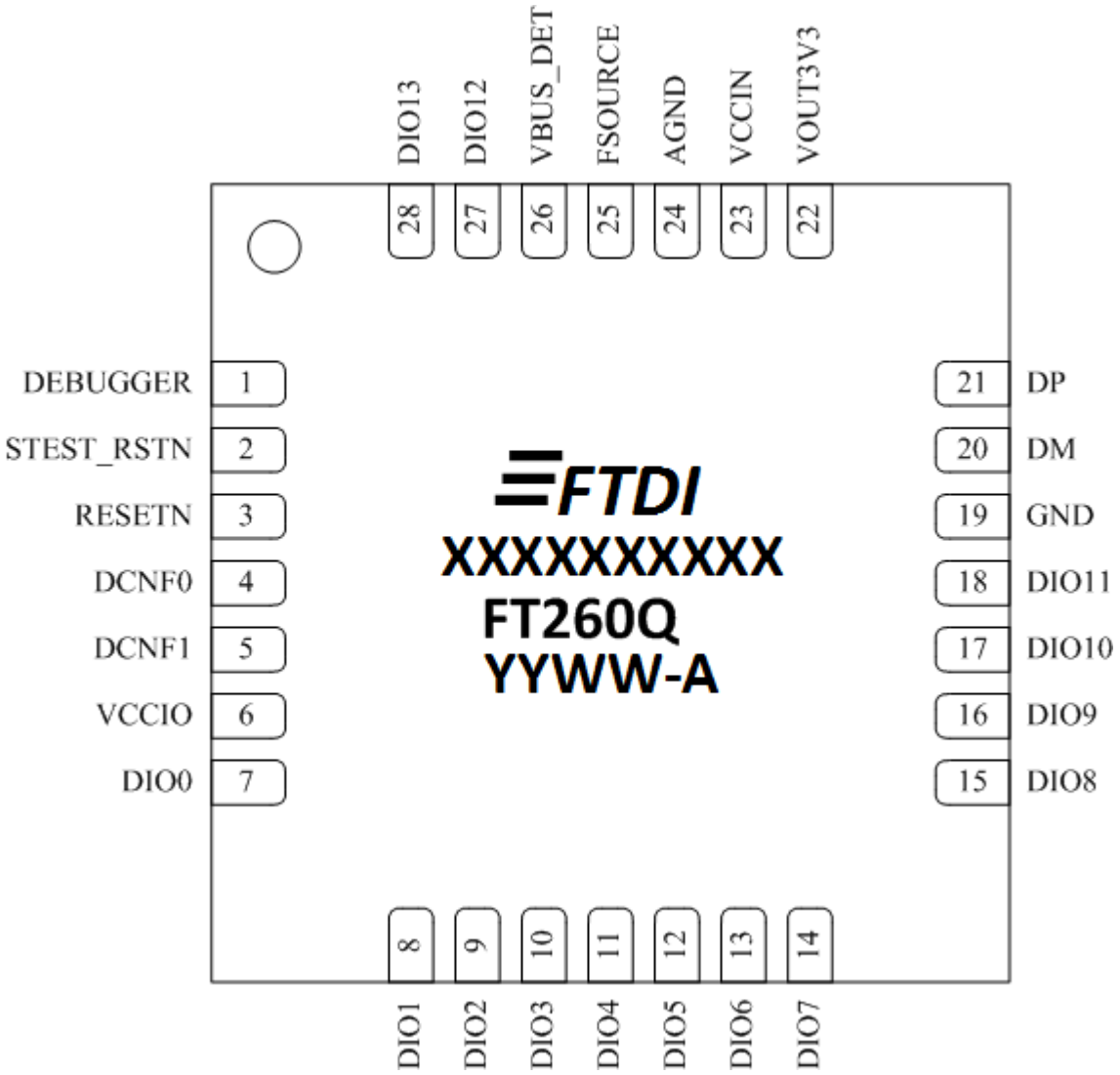

Figure 3.1 Pin Configuration WQFN-28 (top-down view)

3.2 TSSOP-28 Package Pin Out

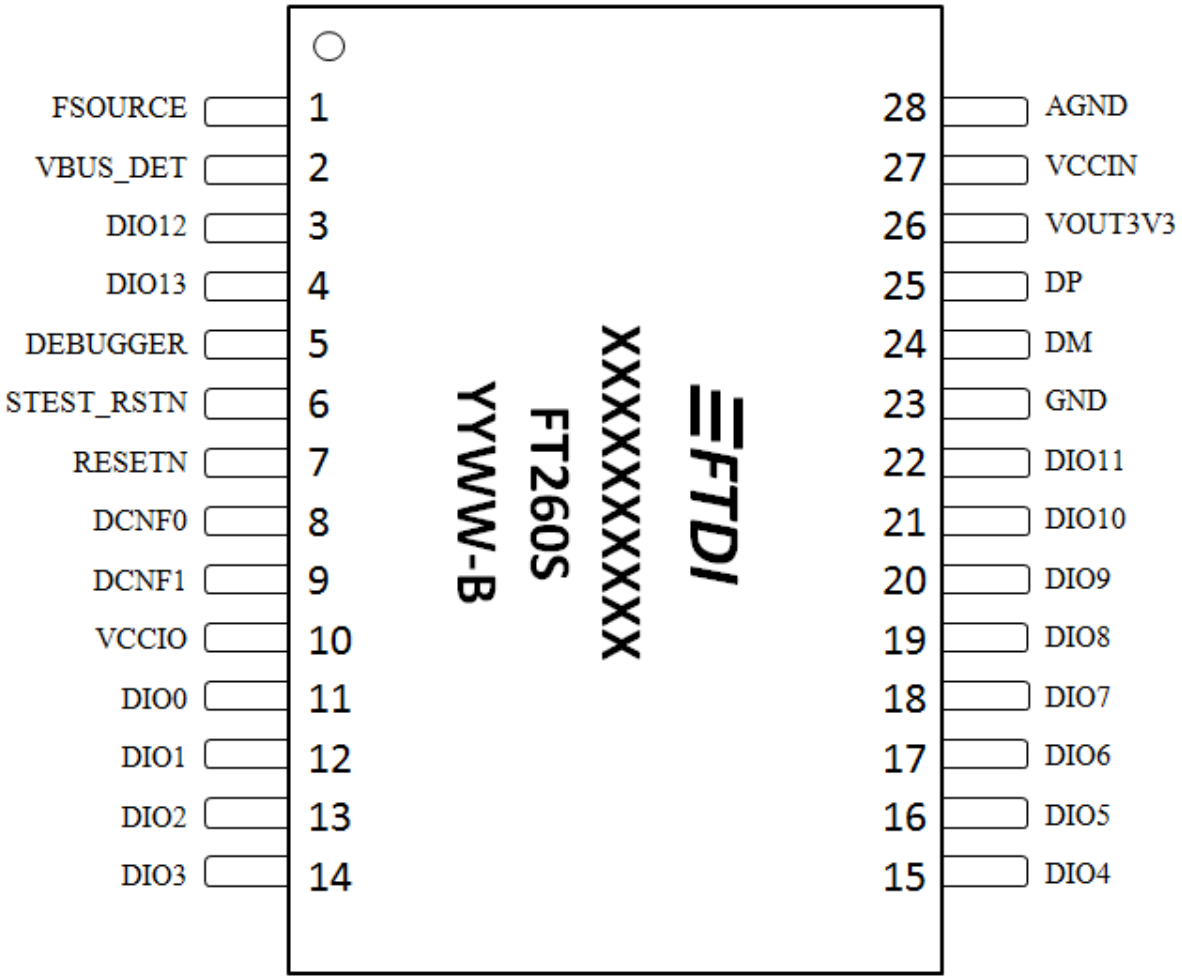

Figure 3.2 Pin Configuration TSSOP-28 (top-down view)

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)	**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**	**I/O**	DIO5, Digital Input/Output Pin 5.

FT260Q Pin No.	FT260S Pin No.	Pin Name (Function)	Type	Description
		(SCL / GPIO0)	I/O I/O	SCL, Serial clock for I^2C bus with open 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)	**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 / 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. 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. GPIO1, General Purpose I/O. GPIO1 is another optional function and can be enabled via a USB command.
16	20	**DIO9** (GPIOF / DTRN)	**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 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

FT260Q Pin No.	FT260S Pin No.	Pin Name (Function)	Type	Description
				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, then VOUT3V3 and VCCIO must also be driven with this 3.3V power source. For details refer to Section 7.3.

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 have to communicate with the FT260 via the HID reports. Please refer to Application Notes for detail formats. In order 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 1 ms, 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.

An 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 I^2C slave function compliant to HID-over-I^2C 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

(http://www.ftdichip.com/Support/Utilities.htm#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 is capable of operating 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 USB 2.0 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 1.2 Kbaud to 12 Mbaud.

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 1.2 Kbaud to 12 Mbaud 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 (http://www.ftdichip.com/Support/Utilities.htm#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 100 Kbit/s, fast mode (FM) with a bit rate up to 400 Kbit/s, Fast mode plus (FM+) with a bit rate up to 1 Mbit/s, and High Speed mode (HS) with the bit rate up to 3.4 Mbit/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 100 KHz 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 60 KHz 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

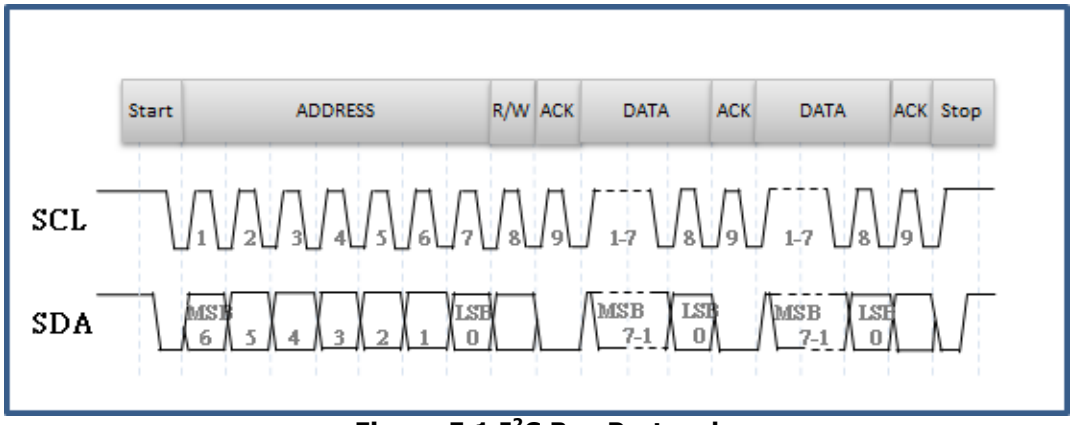

Figure 5.1 I²C Bus Protocol

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 message, 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 is able to 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

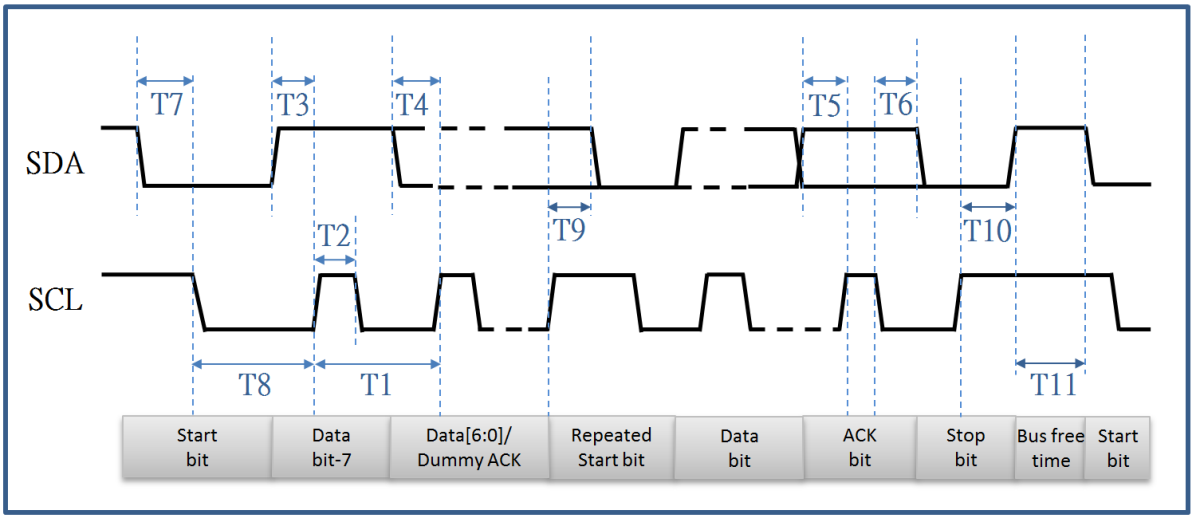

Figure 5.2 I²C Bus Timing

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 I^2C 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 I^2C Timing for VCCIO=3.3V

*Note that 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 1.2 Kbaud to 12 Mbaud 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 below. 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%
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 I^2C 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 are sent.
  - **Even** parity. This means that the parity bit is set to either '1' or '0' so that an even number of 1's are 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 receive waveforms are illustrated in the below figures.

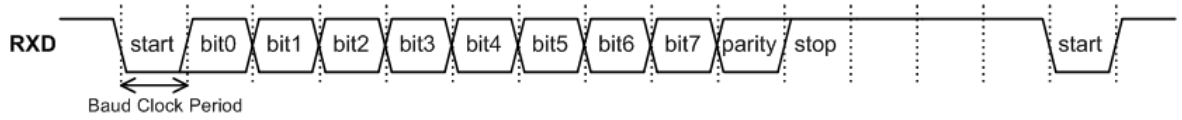

Figure 5.3 UART RX Waveform consist of 8 data bits, 1 optional parity bit and 1 stop bit

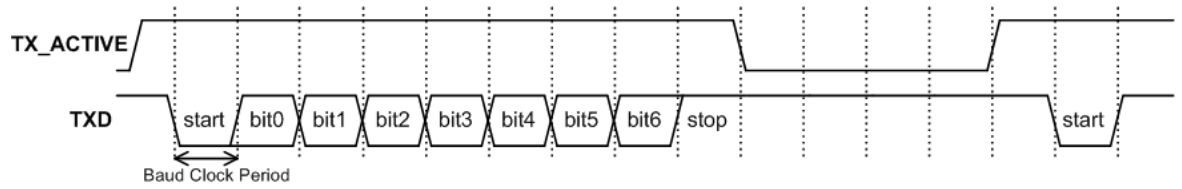

Figure 5.4 UART TX Waveform consist of 7 data bits, no parity bit and 1 stop bit

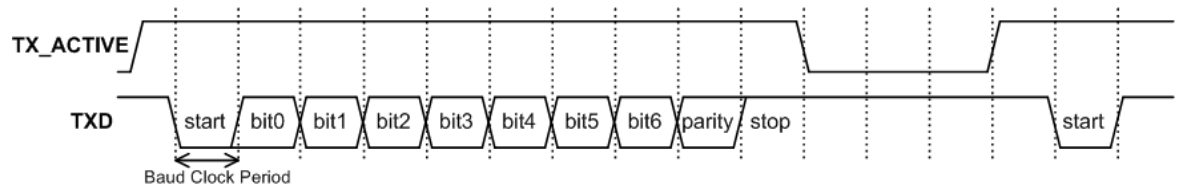

Figure 5.5 UART TX Waveform consist of 7 data bits, 1 optional parity bit and 1 stop bit

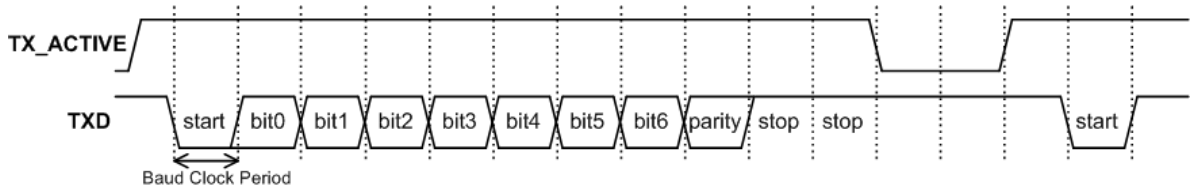

Figure 5.6 UART TX Waveform consisting 7 data bits, 1 optional parity bit and 2 stop bits

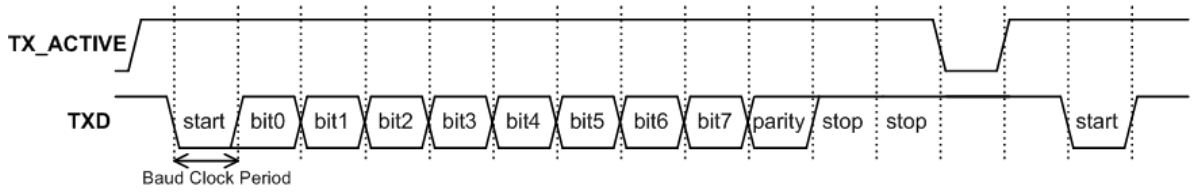

Figure 5.7 UART TX Waveform consisting 8 data bits, 1 optional parity bit and 2 stop bits

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.