Application Note

AN_437

FT602_I2C_User Guide

Version 1.0

Issue Date: 2017-12-08

This application note describes how to use the I^2C master for sideband configuration from a user supplied host application.

Future Technology Devices International Limited (FTDI)
Unit 1,2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): support1@ftdichip.com Web: http://www.ftdichip.com

Table of Contents

1 Introduction

The FT602 device uses the on-board I2C master interface exchange configuration information with the FIFO master. The FIFO master is required to implement a fixed set of I2C slave registers to receive information from the device during initialisation and also from the UVC host driver. The I2C interface may also be used for general purpose access to additional I2C slaves on the bus. This document shows how this may be achieved.

1.1 Overview

The FT602 provides an I^2C bus which operates as master with a default transmission speed of 1Mb/s. This speed is configurable and may be chosen from amongst 100Kb/s, 400Kb/s and 1Mb/s through the configuration programmer.

When using video capture applications, when any UVC control value is changed, a UVC control message is sent to the UVC device (FT602). The device, in turn, relays the message to the FPGA using the I^2C Interface. The I^2C slave address to be used for this communication is set via the configuration programmer. The I^2C slave (e.g. FPGA or FIFO Master) shall implement the registers described in the next sections. Apart from these registers, I^2C slave may implement custom registers which are controlled or configured through a user application.

2 Package Overview

FT602 software release package has the following components.

Component	Description
FT602WinUSBInstallation.exe	An installer to install WinUSB drivers for FT602 devices
FT602WinUSBInstallationGuide.docx	Documentation of the install process.
Configuration Programmer	Application to make configuration changes to FT602 device
I2C Demo Application	Sample source code and application which demonstrates I2C access

Table 1 - Package Components

3 Auxiliary Interface

The FT602 UVC device supports up to 4 video channels and one auxiliary interface. All the I^2C related commands from the user application are communicated over the auxiliary interface. Auxiliary interface is optional and may be disabled when it is not required in the product application. When the interface is enabled, it is enumerated as highlighted in the below picture.

Section 4, describes the process for enabling / disabling the auxiliary interface.

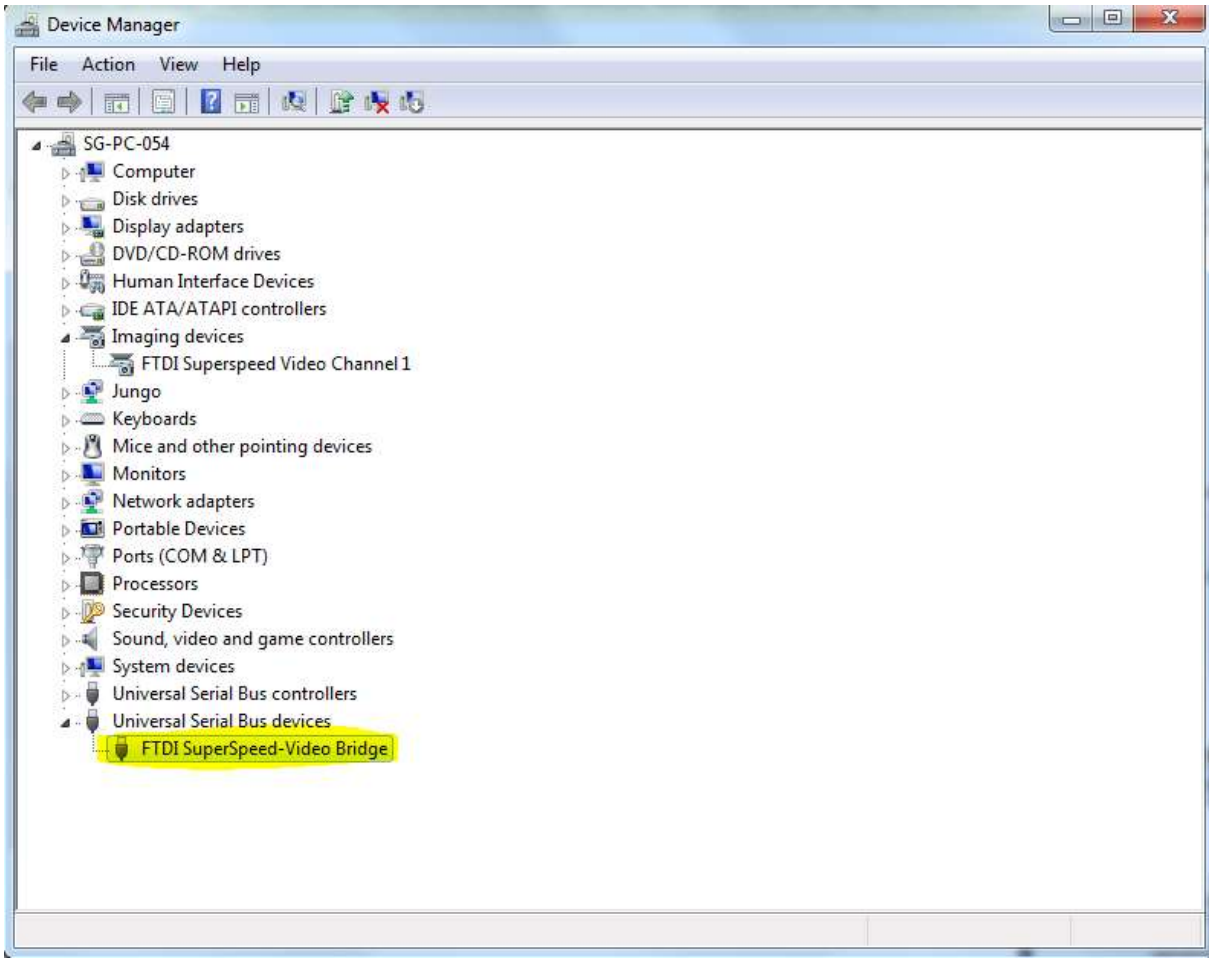

Figure 1 Auxiliary Interface Highlighted

The auxiliary interface uses Microsoft WinUSB drivers. In most cases, Windows is able to successfully install WinUSB drivers automatically from Windows Update. However, on certain versions WinUSB drivers for this interface may not load or install successfully. In such cases, manual installation of WinUSB drivers is required.

For manual installation follow the below steps:

1. Double click FT602WinUSBInstallation.exe.
2. Click on the 'extract' button to unpack the installer.

Figure 2 WinUSB Setup Procedure

3. Click Next.

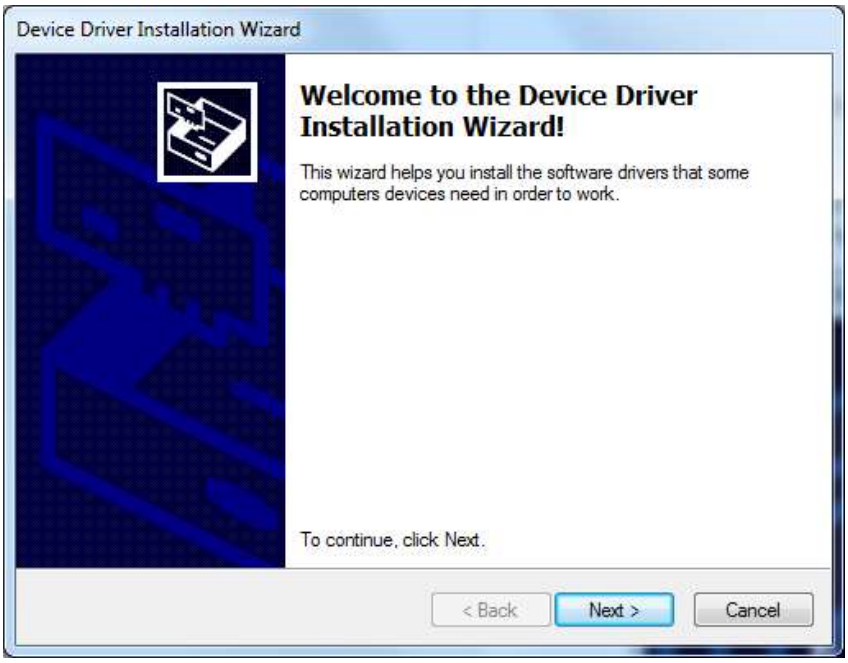

Figure 3 WinUSB Setup Procedure

4. Select 'I accept this agreement' and click Next.

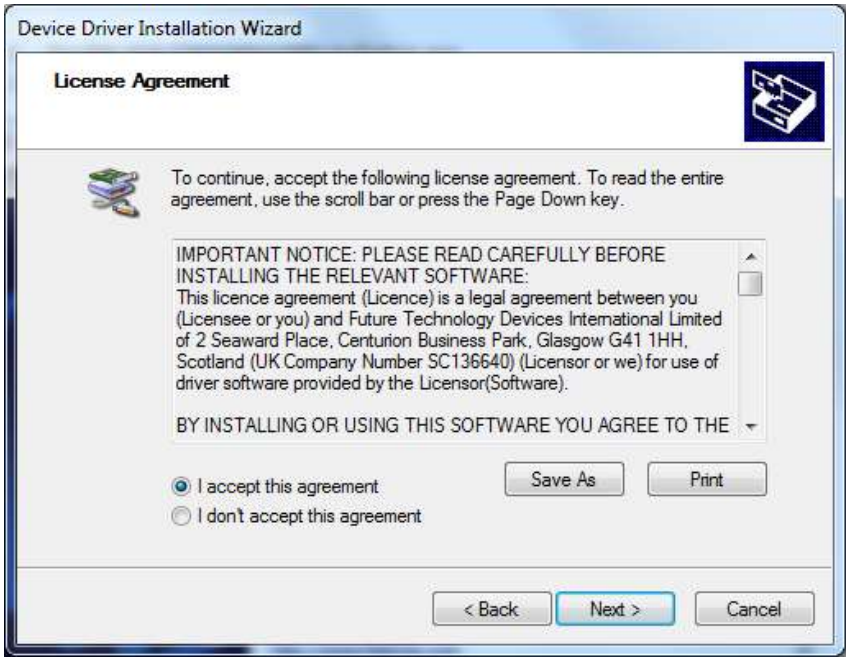

Figure 4 WinUSB Setup Procedure

5. Click on Finish button. Installation is completed.

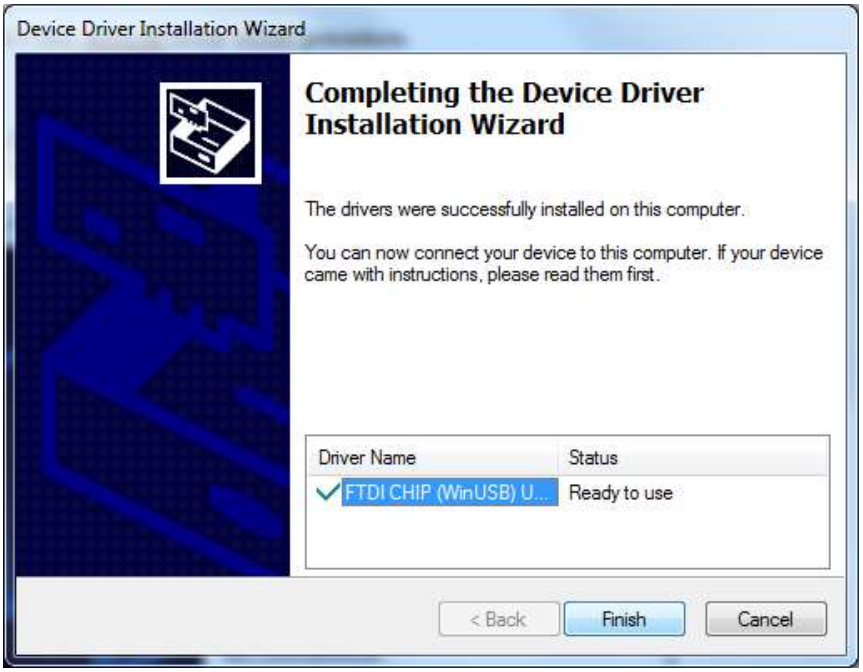

Figure 5 WinUSB Setup Procedure

4 Enable/Disable Auxiliary Interface

To enable/disable the auxiliary interface, launch the FT602 Configuration Programmer. Check the option "Auxiliary Interface" and then click "Write Configuration" to enable the interface. Uncheck the option and write configuration to disable the interface.

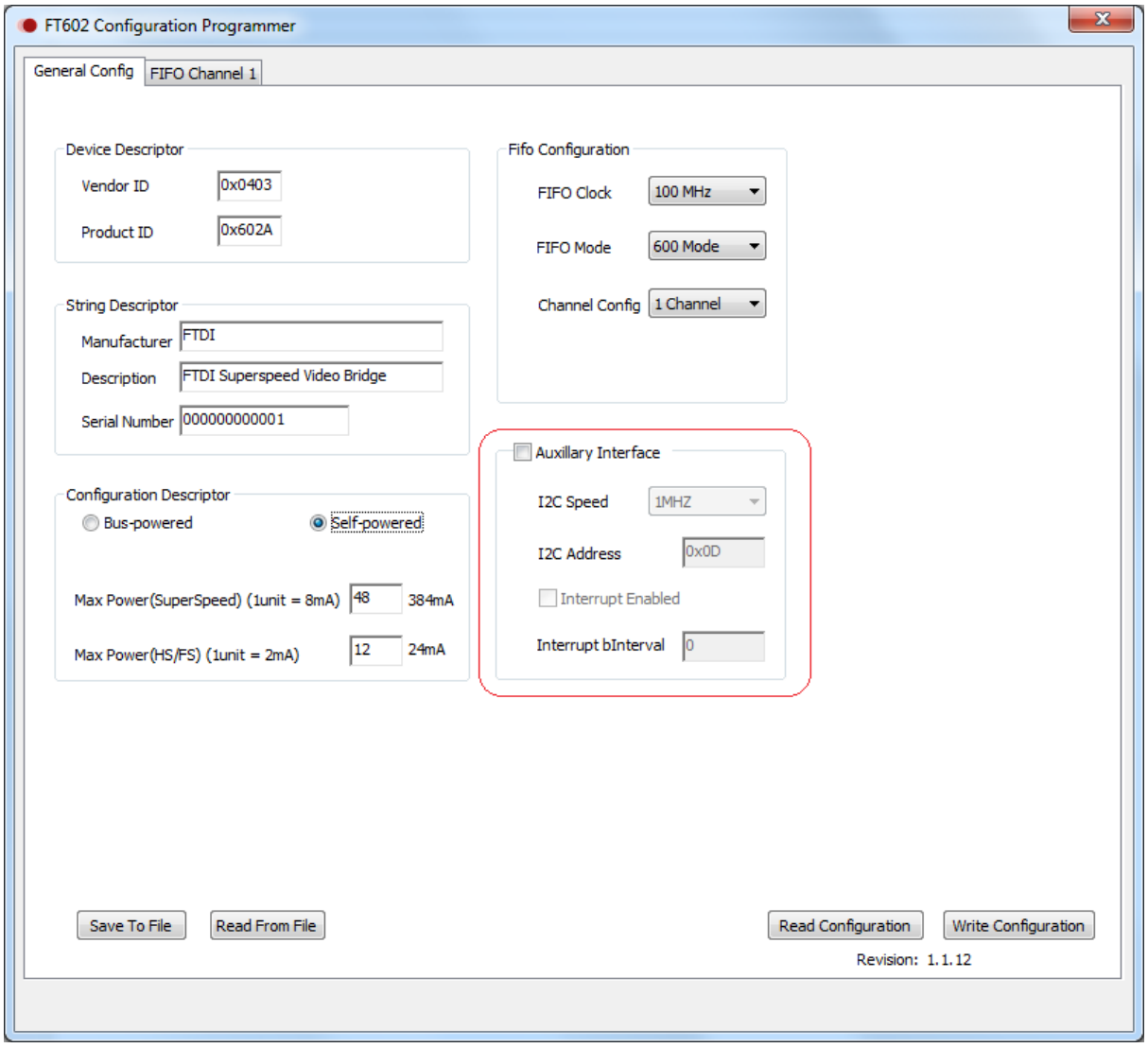

Figure 6 Enable / Disable Auxiliary Interface

5 I²C Register Description

1. The I²C registers have different widths and must be written or read in full.
2. In the following, some registers use a unit of pixel clock. The pixel clock depends on the frame resolution and the selected frame rates. Error! Reference source not found. provides the frame resolutions and pixel clock frequency
3. 16 bytes of software defined registers are provided (e.g. scratch) per channel for storing UVC control and command information for storage and retrieval. Customer designs may act on the values written into these locations or provide a response to the USB host via these locations. The IRQ signal may be used to signal the host of change in conditions in the FPGA.

➢ **Convention**

RO: Read Only

RW: Read, Write

Device Address: 0x0D (seven bits address)

Address	Name	Size (Byte)	Type	Description	Default
0x00	Configuration	1	RW	bit[7:4]: reserved bit[3]: I²C interrupt, 0 = disable, 1 = enable bit[2:1]: 2'b00 = unknown speed 2'b01 = Super Speed 2'b10 = High Speed 2'b11 = Full Speed B[0]: FIFO Mode, 0 = FT245, 1 = FT600	0x03
0x01	FT602 Power Status	1	RW	Bit[7:2]: Reserved Bit[1:0]: FT602 Power Status 00: Active 01: Suspend 1x: Power off	0x00
0x10	Ch0_Control	2	RW	Bit[15:8]:Frame Rate 00-01: 1 Frame per second 02-0xFF: 2-255 frames per second Bit[7]: Pattern Select 0: Moving Pattern	0x3C00

Address	Name	Size (Byte)	Type	Description	Default
				1: Fixed Pattern Bit[6:5]: Four kind of patterns 2'b11: 3 vertical bars 2'b10: 5 vertical bars 2'b01: 7 vertical bars 2'b00: 8 vertical bars Bit[4:0]: Reserved	
0x11	Ch0_LSG_LEG	2	RW	Bit[15:8]: Line Start Gap (units of pixel clock) Bit[7:0]: Line End Gap (units of pixel clock)	0x0804
0x12	Ch0_DSG_DEG	2	RW	Bit[15:8]: Data Start Gap (units of pixel clock) Bit[7:0]: Data End Gap (units of pixel clock)	0x0804
0x13	Ch0_H_Blank	2	RW	Number of clock cycle of H Blank (units of pixel clock)	0x0010
0x14	Ch1_Control	2	RW	Bit[15:8]:Frame Rate 00-01: 1 frame per second 02-0xFF: 2-255 frames per second Bit[7]: Pattern Select 0: Moving Pattern 1: Fixed Pattern Bit[6:5]: Four kind of patterns 2'b11: 3 vertical bars 2'b10: 5 vertical bars 2'b01: 7 vertical bars 2'b00: 8 vertical bars Bit[4:0]: Reserved	0x3C20
0x15	Ch1_LSG_LEG	2	RW	Bit[15:8]: Line Start Gap (units of pixel clock) Bit[7:0]: Line End Gap (units of pixel clock)	0x0804
0x16	Ch1_DSG_DEG	2	RW	Bit[15:8]: Data Start Gap (units of pixel clock) Bit[7:0]: Data End Gap (units of pixel clock)	0x0804

Address	Name	Size (Byte)	Type	Description	Default
0x17	Ch1_H_Blank	2	RW	Number of clock cycle of H Blank (units of pixel clock)	0x0010
0x18	Ch2_Control	2	RW	Bit[15:8]:Frame Rate 00-01: 1 frame per second 02-0xFF: 2-255 frames per second Bit[7]: Pattern Select 0: Moving Pattern 1: Fixed Pattern Bit[6:5]: Four kind of patterns 2'b11: 3 vertical bars 2'b10: 5 vertical bars 2'b01: 7 vertical bars 2'b00: 8 vertical bars Bit[4:0]: Reserved	0x3C40
0x19	Ch2_LSG_LEG	2	RW	Bit[15:8]: Line Start Gap (units of pixel clock) Bit[7:0]: Line End Gap (units of pixel clock)	0x0804
0x1A	Ch2_DSG_DEG	2	RW	Bit[15:8]: Data Start Gap (units of pixel clock) Bit[7:0]: Data End Gap (units of pixel clock)	0x0804
0x1B	Ch2_H_Blank	2	RW	Number of clock cycle of H Blank (units of pixel clock)	0x0010
0x1C	Ch3_Control	2	RW	Bit[15:8]:Frame Rate 00-01: 1 frame per second 02-0xFF: 2-255 frames per second Bit[7]: Pattern Select 0: Moving Pattern 1: Fixed Pattern Bit[6:5]: Four kind of patterns 2'b11: 3 vertical bars 2'b10: 5 vertical bars 2'b01: 7 vertical bars 2'b00: 8 vertical bars Bit[4:0]: Reserved	0x3C50

Address	Name	Size (Byte)	Type	Description	Default
0x1D	Ch3_LSG_LEG	2	RW	Bit[15:8]: Line Start Gap (units of pixel clock) Bit[7:0]: Line End Gap (units of pixel clock)	0x0804
0x1E	Ch3_DSG_DEG	2	RW	Bit[15:8]: Data Start Gap (units of pixel clock) Bit[7:0]: Data End Gap (units of pixel clock)	0x0804
0x1F	Ch3_H_Blank	2	RW	Number of clock cycle of H Blank (units of pixel clock)	0x0010
0x20	Ch0_Low_Mark	2	RW	Water Mark of channel 0	0x0000
0x21	Ch0_High_Mark	2	RW	Buffer size of channel 0	0x0FFF
0x22	Ch1_Low_Mark	2	RW	Water Mark of channel 1	0x0000
0x23	Ch1_High_Mark	2	RW	Buffer size of channel 1	0x0FFF
0x24	Ch2_Low_Mark	2	RW	Water Mark of channel 2	0x0000
0x25	Ch2_High_Mark	2	RW	Buffer size of channel 2	0x0FFF
0x26	Ch3_Low_Mark	2	RW	Water Mark of channel 3	0x0000
0x27	Ch3_High_Mark	2	RW	Buffer size of channel 3	0x0FFF
0x28	Ch0_Frame_Drop	4	RW	Number of discarded frames of channel 0	0x0000_0000
0x29	Ch1_Frame_Drop	4	RW	Number of discarded frames of channel 1	0x0000_0000
0x2A	Ch2_Frame_Drop	4	RW	Number of discarded frames of channel 2	0x0000_0000
0x2B	Ch3_Frame_Drop	4	RW	Number of discarded frames of channel 3	0x0000_0000
0x40-0x5F	UVC Control Channel 0	16	RW	Software Define	X
0x60	Ch0_Start_Stream	9	RW	Big Endian Byte 0-1: Video Frame Width Byte 2-3: Video Frame Height Byte 4-7: Clock Frequency (specified in Hz) Byte 8: Resolution index (defined in	Byte 0-1: 0x0280 (640) Byte 2-3: 0x01E0 (480) Bytes 4-7: 0x019BFCC0

Address	Name	Size (Byte)	Type	Description	Default
				USB descriptor). 0 : VGA 1 : HD 2 : FHD Others : unused	(27MHz) Byte sequence (0-8): {0x80, 0x02, 0xE0, 0x01, 0x01, 0x9B, 0xFC, 0xC0, 0x00}
0x70-0x8F	UVC Control Channel 1	16	RW	Software Define	X
0x90	Ch1_Start_Stream	9	RW	Big Endian Byte 0-1: Video Frame Width Byte 2-3: Video Frame Height Byte 4-7: Clock Frequency (specified in Hz) Byte 8: Resolution index (defined in USB descriptor). 0 : VGA 1 : HD 2 : FHD Others : unused	Byte 0-1: 0x0280 (640) Byte 2-3: 0x01E0 (480) Bytes 4-7: 0x019BFCC0 (27MHz) Byte sequence (0-8): {0x80, 0x02, 0xE0, 0x01, 0x01, 0x9B, 0xFC, 0xC0, 0x00}
0xA0-0xBF	UVC Control Channel 2	16	RW	Software Define	X
0xC0	Ch2_Start_Stream	9	RW	Big Endian Byte 0-1: Video Frame Width Byte 2-3: Video Frame Height Byte 4-7: Clock Frequency (specified in Hz) Byte 8: Resolution index (defined in USB descriptor). 0 : VGA 1 : HD 2 : FHD Others : unused	Byte 0-1: 0x0280 (640) Byte 2-3: 0x01E0 (480) Bytes 4-7: 0x019BFCC0 (27MHz) Byte sequence (0-8): {0x80, 0x02, 0xE0, 0x01, 0x01, 0x9B, 0xFC, 0xC0, 0x00}

Address	Name	Size (Byte)	Type	Description	Default
0xD0- 0xEF	UVC Control Channel 3	16	RW	Software Define	X
0xF0	Ch3_Start_Stream	9	RW	Big Endian Byte 0-1: Video Frame Width Byte 2-3: Video Frame Height Byte 4-7: Clock Frequency (specified in Hz) Byte 8: Resolution index (defined in USB descriptor). 0 : VGA 1 : HD 2 : FHD Others : unused	Byte 0-1: 0x0280 (640) Byte 2-3: 0x01E0 (480) Bytes 4-7: 0x019BFCC0 (27MHz) Byte sequence (0-8): {0x80, 0x02, 0xE0, 0x01, 0x01, 0x9B, 0xFC, 0xC0, 0x00}

Table 2 - I²C Registers

UVC Control Register Address				Register Name	Description
Channel 1	Channel 2	Channel 3	Channel 4		
40	70	A0	D0	REG_CT_AE_MODE	Auto Exposure Mode
41	71	A1	D1	REG_CT_AE_PRIORITY	Auto Exposure Priority
42	72	A2	D2	REG_CT_EXPOSURE_TIME_ABSOLUTE	Exposure Time Absolute
43	73	A3	D3	REG_CT_EXPOSURE_TIME_RELATIVE	Exposure Time Relative
44	74	A4	D4	REG_CT_FOCUS_ABSOLUTE	Focus Absolute
45	75	A5	D5	REG_CT_FOCUS_RELATIVE	Focus Relative
46	76	A6	D6	REG_CT_FOCUS_AUTO	Focus, Auto.
47	77	A7	D7	REG_CT_IRIS_ABSOLUTE	IRIS Absolute
48	78	A8	D8	REG_CT_IRIS_RELATIVE	IRIS Relative.
49	79	A9	D9	REG_CT_ZOOM_ABSOLUTE	Zoom Absolute
4A	7A	AA	DA	REG_CT_ZOOM_RELATIVE	Zoom Relative
4B	7B	AB	DB	REG_CT_PANTILT_ABSOLUTE	Pan-tilt Absolute
4C	7C	AC	DC	REG_CT_PANTILT_RELATIVE	Pan-tilt Relative

UVC Control Register Address				Register Name	Description
Channel 1	**Channel 2**	**Channel 3**	**Channel 4**		
4D	7D	AD	DD	REG_CT_ROLL_ABSOLUTE	Roll Absolute
4E	7E	AE	DE	REG_CT_ROLL_RELATIVE	Roll Relative
4F	7F	AF	DF	REG_PU_BACKLIGHT_COMPENSATION	Backlight Compensation
50	80	B0	E0	REG_PU_BRIGHTNESS	Brightness
51	81	B1	E1	REG_PU_CONTRAST	Contrast
52	82	B2	E2	REG_PU_GAIN	Gain
53	83	B3	E3	REG_PU_POWER_LINE_FREQUENCY	Power line frequency
54	84	B4	E4	REG_PU_HUE	Hue
55	85	B5	E5	REG_PU_SATURATION	Saturation
56	86	B6	E6	REG_PU_SHARPNESS	Sharpness
57	87	B7	E7	REG_PU_GAMMA	Gamma
58	88	B8	E8	REG_PU_WHITE_BALANCE_TEMPERATURE	White Balance Temp.
59	89	B9	E9	REG_PU_WHITE_BALANCE_TEMPERATURE_AUTO	White Balance Auto.
5A	8A	BA	EA	REG_PU_WHITE_BALANCE_COMPONENT	White Balance Component.
5B	8B	BB	EB	REG_PU_WHITE_BALANCE_COMPONENT_AUTO	White Balance Component Auto.
5C	8C	BC	EC	REG_PU_DIGITAL_MULTIPLIER	Digital Multiplier
5D	8D	BD	ED	REG_PU_DIGITAL_MULTIPLIER_LIMIT	Digital Multiplier Limit
5E	8E	BE	EE	REG_PU_HUE_AUTO	Hue Auto.
5F	8F	BF	EF	REG_PU_ANALOG_VIDEO_STANDARD	Analog Video Standard.

Table 3 - UVC Registers

Index	USB MODE	Resolution	Pixel Clock
1	High Speed / Full Speed	QVGA	27 MHz
2	Super Speed	VGA	27 MHz
3	Super Speed	HD	74.25 MHz
4	Super Speed	Full HD	148.5 MHz

Table 4 – Pixel Clock

6 I²C Access

A user application can use the FTDI API FT_I2CAccess to communicate with an I²C slave. The demo application and sample reference code can be found here.

6.1 Reading a Register from FIFO Master/FPGA

Below piece of code is reading a register 0x13 which is number of clock cycle of H Blank (units of pixel clock).

```
            i2c.addr = 0x0D;            /* I2C device address as set in the config */
            i2c.len = 2;                /* Length of Data*/
            i2c.read_access = TRUE;     /* Read */
            i2c.reg = 0x13;            /* Offset to read */
            i2c.reg_is_16_bit = 0;      /* Set to 1 if the register address is 16bit
Otherwise 0 */

            FT_I2CAccess(ftHandle, &i2c, (uint8_t*)&read_value);
            /* read_value returns the value read from the I2C Slave */
```

6.2 Writing to a FIFO Master/FPGA Register

Below piece of code is for updating the register 0x13 to a new value (0x12) for the number of clock cycle of H Blank.

```
            i2c.addr = 0x0D;            /* I2C device address as set in the config */
            i2c.len = 2;                /* Length of Data */
            i2c.read_access = FALSE;    /* WRITE */
            i2c.reg = 0x13;            /* Offset to Write */
            i2c.reg_is_16_bit = 0;      /* Set to 1 if the register address is 16bit
otherwise 0*/
            uint16_t Value = 0x12;
            FT_I2CAccess (ftHandle, &i2c, (uint8_t*) &Value);
```

6.3 Accessing Another I2C Device

Accessing another device on the I2C bus using FT_I2CAccess API is more or less same as accessing a FIFO Master / FPGA register. The only change will be to input the right I2C device address. Assume there is an EEPROM attached to the bus and the address is 0x23. Below sample shows how to read a single byte value from offset 0x10 of the EEPROM.

```
            i2c.addr = 0x23;           /* I2C EEPROM address */
            i2c.len = 1;               /* Length */
            i2c.read_access = TRUE;    /* Read */
            i2c.reg = 0x10;           /* Offset to read */
            i2c.reg_is_16_bit = 0;     /* Set to 1 if the register address is 16bit
otherwise 0*/

            FT_I2CAccess(ftHandle, &i2c, (uint8_t*)&read_value);
```

6.4 Interrupt Notification

A user application can also receive notifications based on certain events. I^2C slave can utilize IRQ signal to notify the host. As seen in the figure 6, from the configuration programmer, after enabling the auxiliary interface, check the interrupt enable option.

A user application needs to register a callback to receive interrupt notifications. The FTDI API FT_SetNotificationCallback registers the callback for interrupt notification.

Below piece of code shows how to register for a notification callback.

```
            /* create an event */
            hNotification = CreateEvent(NULL, TRUE, FALSE, NULL);
            if (!hNotification)
            {
                  goto exit;
            }
            /* setup a callback function to receive notifications */
            FT_SetNotificationCallback(ftHandle, NotificationCallback, NULL);

            /* wait for the event to signal
             * Program exits only when this event is signalled
             * NotificationCallback happens asynchronously */
            DWORD dwRet = WaitForSingleObject(hNotification, INFINITE);
```

Notification callback gets called when the FPGA generates an interrupt on the IRQ line. What action to be taken in the notification callback is based on FPGA design and the FPGA designers can implement custom protocols.

Below piece of code shows a sample callback function which reads a FPGA register to decide on what action to be taken.

```
Static VOID NotificationCallback(PVOID pvCallbackContext,
              E_FT_NOTIFICATION_CALLBACK_TYPE eCallbackType,
              PVOID pvCallbackInfo)
{
       /* eCallbackType : unused argument and can be ignored */
       /* pvCallbaclInfo : context passed in FT_ SetNotificationCallback */

       struct i2c_access i2c;
       DWORD dwAction = 0;
        i2c.addr = 0x0D;           /* I2C device address as set in the config*/
       i2c.len = 4;                /* Length */
       i2c.read_access = TRUE;     /* Read */
       i2c.reg = IRQ_INFO_REG;     /* Offset to read */
       i2c.reg_is_16_bit = 0;      /* Set to 1 if the register address is 16bit
otherwise 0 */
       FT_I2CAccess(ftHandle, &i2c, (uint8_t*)& dwAction);

       If(dwAction & FRAME_LOST)
       {
              DWORD dwFrameLostCount;
              i2c.reg = FRAME_ERR_COUNT_REG;
              FT_I2CAccess(ftHandle, &i2c, (uint8_t*)& dwFrameLostCount);
        }
}
```

7 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
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 8791 3570
Fax: +886 (0) 2 8791 3576

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

FT602 IC Datasheet

UMFT602x Datasheet

Altera Cyclone V

Acronyms and Abbreviations

Terms	Description
API	Application Programming Interface
I^2C	Inter-Integrated Circuit
FPGA	Field Programmable Gate Array
USB	Universal Serial Bus
UVC	USB Video Class

Appendix B – List of Tables & Figures

List of Figures

List of Tables

Appendix C – Revision History

Document Title : AN_437 FT602_I2C_User Guide
Document Reference No.: FT_001416
Clearance No.: FTDI#537
Product Page: http://www.ftdichip.com/FTProducts.htm
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2017-12-08