Application Note

AN_414

FT90x UVC WebCam

Version 1.2

Issue Date: 2017-04-04

This application note describes using the camera interface on the FT90x to create a UVC compliant webcam USB device.

Bridgetek Pte Ltd (BRT Chip)

178 Paya Lebar Road, #07-03 Singapore 409030
Tel : +65 6547 4827 Fax : +65 6841 6071
Web Site: http://brtchip.com

Table of Contents

1 Introduction

The FT900/FT901 and FT905/FT906 devices include a Camera Interface and Serial Camera Control Bus (SCCB - implemented as an I^2C Master on the FT90x). This can be used with suitable USB code to implement a UVC webcam.

This application note demonstrates how data from the camera interface can be read and sent to the USB interface. A USB host can receive the video data and display the output as it would with a dedicated webcam.

1.1 Overview

The source of the video data is the camera interface. This receives data streamed from a camera module on an 8-bit parallel bus. This is buffered in the camera interface peripheral and an interrupt signals to the CPU that data is available. Data can then be read into memory by an application.

Because of the high throughput of data, special features of the FT90x core are used. The interrupt routine uses the "streaming" instruction to transfer data from the camera interface registers to memory as fast as possible.

Support is provided for two camera modules, the OV9657 and OV5640. Both modules can output an uncompressed YUY2 stream; the OV5640 can also output a compressed Motion JPEG (MJPEG) stream.

The UVC device has configurations for hi-speed and full-speed devices, however data will only be transferred if the device enumerates as a hi-speed device. This implementation, by default, will allow QVGA and VGA resolutions (320x240, 640x480 respectively) uncompressed at high-speed, and additionally at SVGA resolution (800x600) in MJPEG format. All are at 15 frames per second.

Data is transferred to the USB host using either bulk transfers or isochronous transfers. By default, a bulk endpoint is used to transfer data rather than the more common isochronous endpoint. This can be changed in the code. When using isochronous transfers only QVGA is supported.

1.2 Scope

The webcam application demonstrates the creation of a UVC compatible USB device which will stream video data to a USB host. It does not include audio. Either uncompressed data or MJPEG can be sent to the host. The format of both types of data stream is not altered by the application and is dependent on the camera module sending correctly formatted data.

This is a minimum implementation of the UVC class protocol. In hi-speed mode there is up-to 2 formats allowed and support for up-to 5 resolutions. The available formats and resolutions depend on the camera module used. There are no video streaming interfaces at full-speed. USB device endpoint types of bulk and isochronous are shown. Camera module type, formats, resolutions and endpoint type are configured as build options.

The UVC device is coded to support UVC revision 1.1 to allow for a wide range of USB hosts to work with the webcam.

1.2.1 Features

This application features methods to receive data from the camera interface as fast as possible and for efficiently sending that data to the USB host.

The FT90x features shown are:

- Providing a USB device with configuration descriptors for composite devices, containing interface association descriptors and supporting multiple interfaces and endpoints.
- Sending partial packets of data to the USB device using multiple calls to `USBD_transfer_ex()` function.
- Using the "streaming" instruction to access a peripheral register.

- Accessing the camera interface.
- Using 16-bit addressing on the I^2C master interface.

1.2.2 Enhancement

Enhancements to this application might be:

- To implement variable frame rates.
- Add audio channel.
- Improve feedback to the host on changes in gain.

This UVC webcam example application should be treated as an example. Full source code is provided allowing users to build and modify if required:

http://brtchip.com/ft90x/#FT90x UVC Webcam

2 Project Overview

The project files for this application note are divided into the following folders.

Folder	Description
Source	Application source code and camera module specific files.
Lib	Library files.
lib\tinyprintf	tinyprintf library.

Table 2.1 Project Files Overview

The application source code is contained within the "`Sources`" folder. The main() function and the high-level functions of the application are in the "`main.c`" file. In this folder there are camera module specific file for each supported camera module.

2.1 Main Program

The main program is responsible for setting up the hardware and handling data between the camera interface and the USB device.

The main() function performs hardware setup for the camera interface, I^2C master (for the SCCB), GPIO, timers, UART (for debug output) and power management.

Once the hardware is setup then the usbd_testing() function is called which receives USB requests, UVC device state changes and performs the streaming of data from the camera interface to the USB.

2.2 Camera Control

The camera module requires initialization and some control functions to allow it to be set to stream video data in the correct format. The functions for this are contained in separate files to allow changes to the supported camera modules.

At present the OmniVision OV9657 and OV5640 camera modules are supported and the required functions are included in the files "`ov965x_camera.c`" and "`ov5640_camera.c`" respectively.

These files implement an initialization function (mapped to `camera_init`), a setup function (`camera_set`), a start streaming function (`camera_start`) and a stop streaming function (`camera_stop`).

Preprocessor macros are used to map the required functions to module-specific functions in the module source files.

The setup function is responsible for configuring the camera module in the correct

2.3 Camera Interface

An interrupt handler is started when data is available from the camera interface. This handler streams a line of video data at a time to the main memory. An array capable of holding data from several lines of video is used to allow several lines of data to be buffered.

The interrupt handler will detect the end of a frame and wait for a frame synchronization signal from the camera module which is received on a GPIO line.

2.3.1 I²C Master Interface

The camera module is controlled by commands to read and write registers on an SCCB which is implemented as an I²C master bus. This is camera module specific.

The FT90x library driver for I²C only supports 8-bit addressing. The OV5640 module code includes a set of subroutines to allow 16-bit addressing on the I²C bus. These may be reused for other projects.

2.3.2 GPIO Handler

One GPIO input is configured to receive a frame synchronization signal from the camera module. This is used by the camera interface interrupt handler to signal the start of a new frame of video.

2.4 USB Interface

A handler has been implemented for UVC class requests. Only class requests used by the Windows USB video device driver are implemented.

Configuration descriptors are provided for hi-speed and full-speed operation. The UVC video configuration descriptors are included. These describe the capabilities and video formats used by the UVC interface.

There is an option to use bulk or isochronous data endpoints. A macro "USB_ENDPOINT_USE_ISOC" should be defined to enable isochronous otherwise bulk will be used. The data rate available from bulk data endpoints is higher and will therefore support higher resolutions. When isochronous data endpoints are enabled the SetInterface request is used to enable data transmission over the USB.

Class-specific configuration settings are defined for use by the GET_CUR, GET_MIN, GET_MAX, GET_DEF, GET_RES and SET_CUR requests.

2.5 Power Management Handler

When moving between suspend and resume states on USB, the power management handler is required to notify the USBD (USB device) library that the bus state has changed.

2.6 Other Features

The DFU-C facility may be enabled for this application. It can be enabled at startup or can provide a separate interface for updating.

To enable the DFU at startup define the `USB_INTERFACE_USE_STARTUPDFU` macro. This enables a call from the macro STARTUP_DFU() in the main() function. It will briefly enable the USB device on the FT90x and allow a DFU utility to update the application code. This can be removed entirely or configured to alter the number of milliseconds it will wait before closing the USB device and continuing with the application.

To have a DFU interface during normal operation define the `USB_INTERFACE_USE_DFU` macro.

3 Code Structure

The code is written to send data from the camera interface to the USB library efficiently.

Interrupt handlers receive data from the camera interface after synchronising with the vertical sync signal from the camera module (Figure 3.1).

The main function "`usbd_testing()`" polls flags from the camera interrupt handler and sends any data received from the camera interface to the USB device interface (Figure 3.2).

An efficient method of sending data to the USB interface is used. The number of times data is copied is minimised in both the interrupt handler and the main code. Although the FT90x has efficient and fast "`memcpy`" instructions implemented in silicon it is still faster to avoid copying data.

The initial stream of data from the camera interface is sent to a memory buffer which can hold several lines of video data. It is copied directly from this buffer into the USB transmit buffer when required. The `USBD_transfer_ex()` function allows a USB packet to be built up with one or more calls to the function being used to prepend the UVC Payload Header structure to the video data.

When the `USBD_transfer_ex()` function is called, setting the "`flags`" parameter will prevent the function from sending the data. It will wait for further calls to the function to allow the transmit buffer to be loaded. Once it is called with the flags parameter clear then it will send the data in the buffer, however, if the buffer fills to the endpoint maximum packet size then the data will be transmitted. The USB protocol, when using a bulk endpoint, will assume that this is part of a longer transfer. The "`offset`" parameter is used to ensure that data is sent on the endpoint maximum packet size boundary correctly.

Since this feature is used, the UVC payload header can be made separately from the camera interface buffer avoiding the need to copy data before transmission.

The camera interface data is buffered to allow the USB transmission to "catch up". There is a gap between frames and data remaining in the buffer may be sent in this time. If data from a new frame is received before the USB side has finished then the USB frame will restart with the new data.

When discussing "lines" of data in this application, an uncompressed line of data is defined as the data stream which includes one complete line (horizontal resolution * bytes per pixel); a compressed line is a configurable length of data of which multiple lines make up a compressed frame.

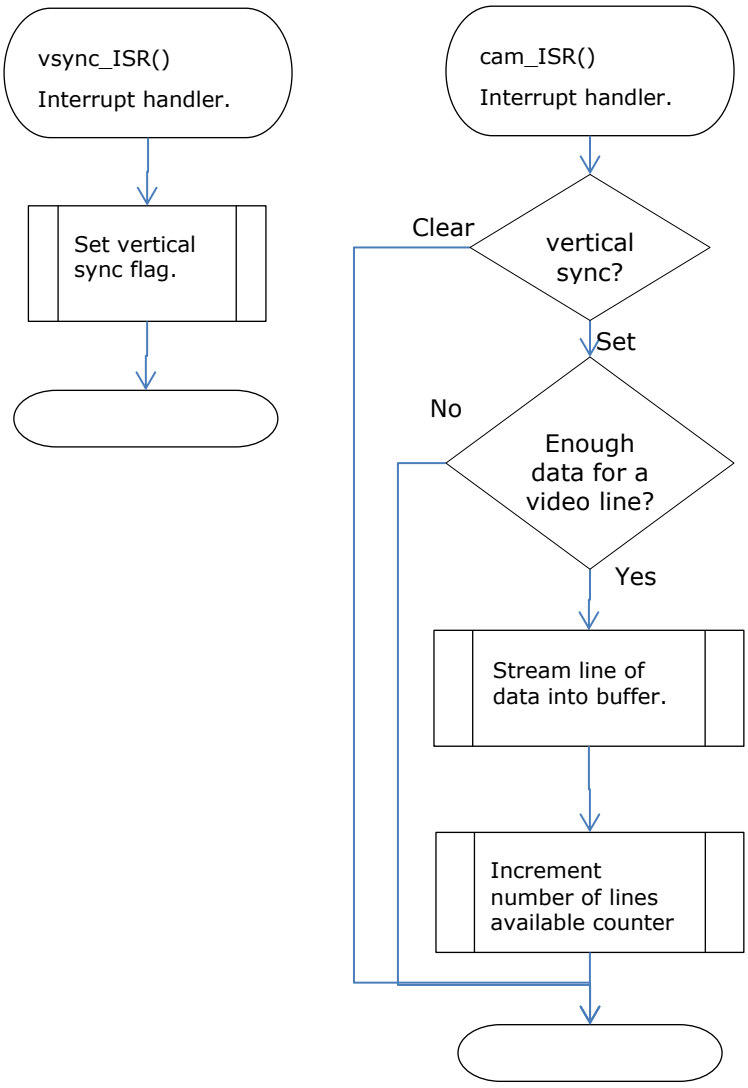

Figure 3.1 Camera Interface Flowchart

Frame detection is handled in the main code rather than the interrupt handler. The camera module is configured to send a fixed amount of data per frame (for both uncompressed and MJPEG frames).

For uncompressed data the frame detection code counts the number of lines received from the camera module and signals a frame end when the number of lines received reaches the expected lines in the frame.

For compressed MJPEG data the frame detection code parses the data stream for JPEG markers which indicate the start and end of each frame.

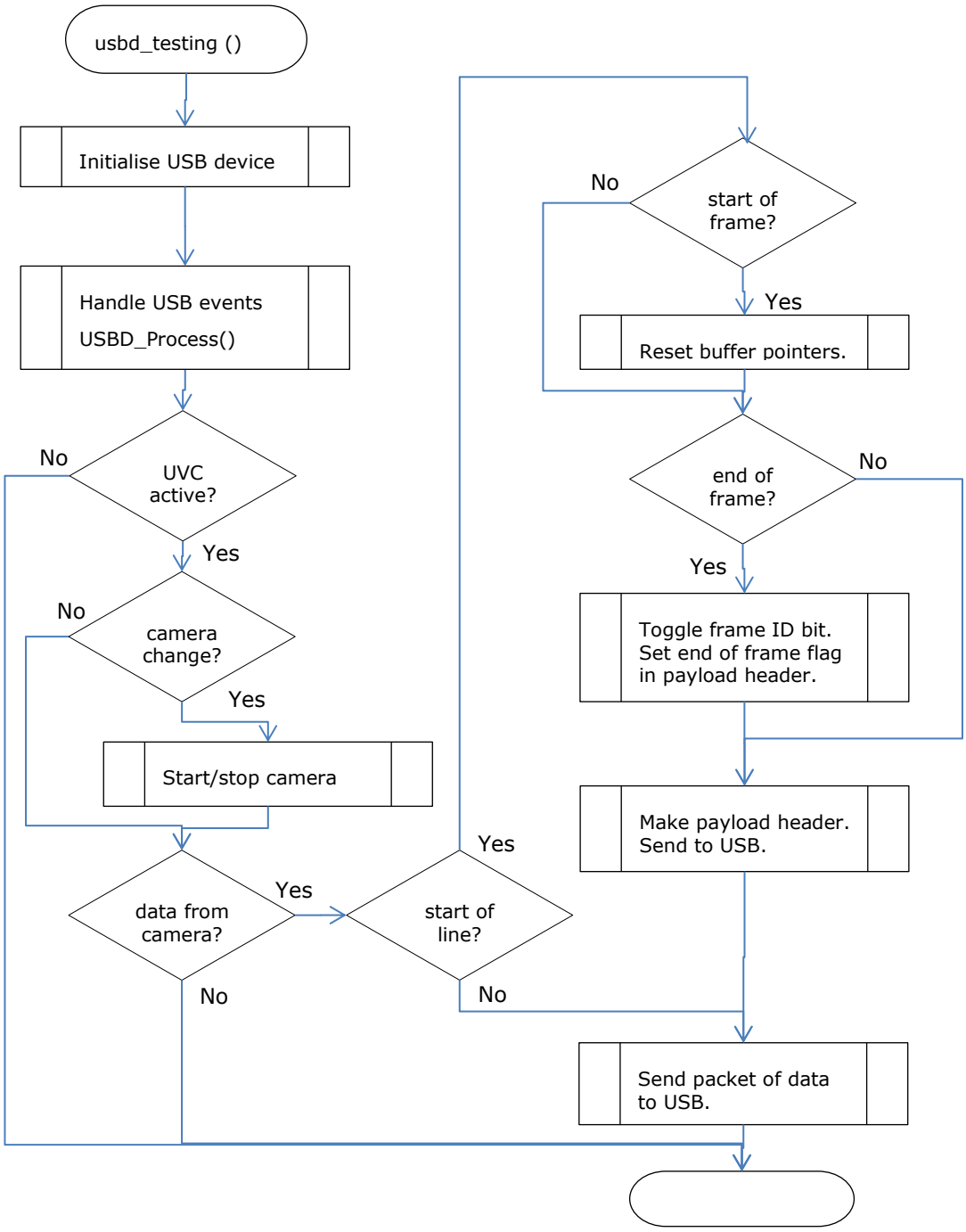

Figure 3.2 Main Control Flowchart

4 USB Configuration

4.1 High-speed Configuration Descriptor

The following configuration descriptor is taken from the FT90x device by "USBview" from Microsoft. It is based on the configuration where:

- The camera module is an OV5640,
- MJPEG streaming is enabled supporting QVGA (default), VGA and SVGA.
- Uncompressed streaming is enabled supporting QVGA (default) and VGA.
- A bulk endpoint is used for streaming.

```
       ------>Device Information<------
English product name: "FT900 UVC"

ConnectionStatus:
Current Config Value:              0x01  -> Device Bus Speed: High
Device Address:                    0x02
Open Pipes:                           2

          ===>Device Descriptor<===
bLength:                           0x12
bDescriptorType:                   0x01
bcdUSB:                          0x0200
bDeviceClass:                      0xEF  -> This is a Multi-interface Function Code Device
bDeviceSubClass:                   0x02  -> This is the Common Class Sub Class
bDeviceProtocol:                   0x01  -> This is the Interface Association Descriptor
protocol
bMaxPacketSize0:                   0x40 = (64) Bytes
idVendor:                        0x0403 = Future Technology Devices International Limited
idProduct:                       0x0FD8
bcdDevice:                       0x0101
iManufacturer:                     0x01
     English (United States)  "FTDI"
iProduct:                          0x02
     English (United States)  "FT900 UVC"
iSerialNumber:                     0x03
     English (United States)  "FT424242"
bNumConfigurations:                0x01

          ------>Open Pipes<------

          ===>Endpoint Descriptor<===
bLength:                           0x07
bDescriptorType:                   0x05
bEndpointAddress:                  0x81  -> Direction: IN - EndpointID: 1
bmAttributes:                      0x03  -> Interrupt Transfer Type
wMaxPacketSize:                  0x0040 = 1 transactions per microframe, 0x40 max bytes
bInterval:                         0x08

          ===>Endpoint Descriptor<===
bLength:                           0x07
bDescriptorType:                   0x05
bEndpointAddress:                  0x82  -> Direction: IN - EndpointID: 2
bmAttributes:                      0x02  -> Bulk Transfer Type
wMaxPacketSize:                  0x0200 = 0x200 max bytes
bInterval:                         0x00
```

```
       ------->Full Configuration Descriptor<------

          ===>Configuration Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x02
wTotalLength:                     0x0180  -> Validated
bNumInterfaces:                     0x03
bConfigurationValue:                0x01
iConfiguration:                     0x00
bmAttributes:                       0x80  -> Bus Powered
MaxPower:                           0xFA = 500 mA

          ===>IAD Descriptor<===
bLength:                            0x08
bDescriptorType:                    0x0B
bFirstInterface:                    0x00
bInterfaceCount:                    0x02
bFunctionClass:                     0x0E  -> Video Interface Class
bFunctionSubClass:                  0x03  -> Video Interface Collection
bFunctionProtocol:                  0x00  -> PC_PROTOCOL_UNDEFINED protocol
iFunction:                          0x02
     English (United States)  "FT900 UVC"

          ===>Interface Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x04
bInterfaceNumber:                   0x00
bAlternateSetting:                  0x00
bNumEndpoints:                      0x01
bInterfaceClass:                    0x0E  -> Video Interface Class
bInterfaceSubClass:                 0x01  -> Video Control Interface SubClass
bInterfaceProtocol:                 0x00
iInterface:                         0x02
     English (United States)  "FT900 UVC"

          ===>Class-Specific Video Control Interface Header Descriptor<===
bLength:                            0x0D
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x01
bcdUVC:                           0x0110
wTotalLength:                     0x0043  -> Validated
dwClockFreq:                 0x02DC6C00 = (48000000) Hz
bInCollection:                      0x01
baInterfaceNr[1]:                   0x01
USB Video Class device: spec version 1.1

          ===>Video Control Input Terminal Descriptor<===
bLength:                            0x11
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x02
bTerminalID:                        0x01
wTerminalType:                    0x0201 = (ITT_CAMERA)
bAssocTerminal:                     0x00
iTerminal:                          0x00
===>Camera Input Terminal Data
wObjectiveFocalLengthMin:         0x0000
wObjectiveFocalLengthMax:         0x0000
wOcularFocalLength:               0x0000
bControlSize:                       0x02
bmControls : 0x00 0x00
     D00 = 0   no -  Scanning Mode
     D01 = 0   no -  Auto-Exposure Mode
     D02 = 0   no -  Auto-Exposure Priority
```

```
     D03 = 0   no -  Exposure Time (Absolute)
     D04 = 0   no -  Exposure Time (Relative)
     D05 = 0   no -  Focus (Absolute)
     D06 = 0   no -  Focus (Relative)
     D07 = 0   no -  Iris (Absolute)
     D08 = 0   no -  Iris (Relative)
     D09 = 0   no -  Zoom (Absolute)
     D10 = 0   no -  Zoom (Relative)
     D11 = 0   no -  PanTilt (Absolute)
     D12 = 0   no -  PanTilt (Relative)
     D13 = 0   no -  Roll (Absolute)
     D14 = 0   no -  Roll (Relative)
     D15 = 0   no -  Reserved

          ===>Video Control Input Terminal Descriptor<===
bLength:                            0x08
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x02
bTerminalID:                        0x02
wTerminalType:                    0x0200 = (ITT_VENDOR_SPECIFIC)
bAssocTerminal:                     0x00
iTerminal:                          0x00

          ===>Video Control Output Terminal Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x03
bTerminalID:                        0x03
wTerminalType:                    0x0101 = (TT_STREAMING)
bAssocTerminal:                     0x00
bSourceID:                          0x05
iTerminal:                          0x00

          ===>Video Control Selector Unit Descriptor<===
bLength:                            0x08
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x04
bUnitID:                            0x04
bNrInPins:                          0x02
===>List of Connected Unit and Terminal ID's
baSourceID[1]:                      0x01
baSourceID[2]:                      0x02
iSelector:                          0x00

          ===>Video Control Processing Unit Descriptor<===
bLength:                            0x0C
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x05
bUnitID:                            0x05
bSourceID:                          0x04
wMaxMultiplier:                   0x0000
bControlSize:                       0x02
bmControls : 0x00 0x00
     D00 = 0   no -  Brightness
     D01 = 0   no -  Contrast
     D02 = 0   no -  Hue
     D03 = 0   no -  Saturation
     D04 = 0   no -  Sharpness
     D05 = 0   no -  Gamma
     D06 = 0   no -  White Balance Temperature
     D07 = 0   no -  White Balance Component
     D08 = 0   no -  Backlight Compensation
     D09 = 0   no -  Gain
```

```
     D10 = 0   no -  Power Line Frequency
     D11 = 0   no -  Hue, Auto
     D12 = 0   no -  White Balance Temperature, Auto
     D13 = 0   no -  White Balance Component, Auto
     D14 = 0   no -  Digital Multiplier
     D15 = 0   no -  Digital Multiplier Limit
iProcessing :                    0x00
bmVideoStandards :               0x00
     D00 = 0   no -  None
     D01 = 0   no -  NTSC  - 525/60
     D02 = 0   no -  PAL   - 625/50
     D03 = 0   no -  SECAM - 625/50
     D04 = 0   no -  NTSC  - 625/50
     D05 = 0   no -  PAL   - 525/60
     D06 = 0   no -  Reserved
     D07 = 0   no -  Reserved

          ===>Endpoint Descriptor<===
bLength:                         0x07
bDescriptorType:                 0x05
bEndpointAddress:                0x81  -> Direction: IN - EndpointID: 1
bmAttributes:                    0x03  -> Interrupt Transfer Type
wMaxPacketSize:                0x0040 = 1 transactions per microframe, 0x40 max bytes
bInterval:                       0x08

          ===>Class-specific VC Interrupt Endpoint Descriptor<===
bLength:                         0x05
bDescriptorType:                 0x25
bDescriptorSubtype:              0x03
wMaxTransferSize:              0x0040 = (64) Bytes

          ===>Interface Descriptor<===
bLength:                         0x09
bDescriptorType:                 0x04
bInterfaceNumber:                0x01
bAlternateSetting:               0x00
bNumEndpoints:                   0x01
bInterfaceClass:                 0x0E  -> Video Interface Class
bInterfaceSubClass:              0x02  -> Video Streaming Interface SubClass
bInterfaceProtocol:              0x00
iInterface:                      0x00

          ===>Video Class-Specific VS Video Input Header Descriptor<===
bLength:                         0x0F
bDescriptorType:                 0x24
bDescriptorSubtype:              0x01
bNumFormats:                     0x02  -> Uncompressed and MJPEG
wTotalLength:                  0x00F5  -> Validated
bEndpointAddress:                0x82  -> Direction: IN - EndpointID: 2
bmInfo:                          0x00  -> Dynamic Format Change not Supported
bTerminalLink:                   0x03
bStillCaptureMethod:             0x00  -> No Still Capture
bTriggerSupport:                 0x00  -> No Hardware Triggering Support
bTriggerUsage:                   0x00
bControlSize:                    0x01
Video Payload Format 1           0x00
     D00 = 0   no -  Key Frame Rate
     D01 = 0   no -  P Frame Rate
     D02 = 0   no -  Compression Quality
     D03 = 0   no -  Compression Window Size
     D04 = 0   no -  Generate Key Frame
     D05 = 0   no -  Update Frame Segment
     D06 = 0   no -  Reserved
```

```
     D07 = 0   no -  Reserved
Video Payload Format 2             0x00
     D00 = 0   no -  Key Frame Rate
     D01 = 0   no -  P Frame Rate
     D02 = 0   no -  Compression Quality
     D03 = 0   no -  Compression Window Size
     D04 = 0   no -  Generate Key Frame
     D05 = 0   no -  Update Frame Segment
     D06 = 0   no -  Reserved
     D07 = 0   no -  Reserved

          ===>Video Streaming Uncompressed Format Type Descriptor<===
bLength:                           0x1B
bDescriptorType:                   0x24
bDescriptorSubtype:                0x04
bFormatIndex:                      0x01
bNumFrameDescriptors:              0x02
guidFormat:                        {32595559-0000-0010-8000-00AA00389B71} = YUY2 Format
bBitsPerPixel:                     0x10
bDefaultFrameIndex:                0x01
bAspectRatioX:                     0x04
bAspectRatioY:                     0x03
bmInterlaceFlags:                  0x00
     D0    = 0x00 Interlaced stream or variable: No
     D1    = 0x00 Fields per frame: 2 fields
     D2    = 0x00 Field 1 first: No
     D3    = 0x00 Reserved
     D4..5 = 0x00 Field patterns  -> Field 1 only
     D6..7 = 0x00 Display Mode  -> Bob only
bCopyProtect:                      0x00  -> Duplication Unrestricted

          ===>Video Streaming Uncompressed Frame Type Descriptor<===
          --->This is the Default (optimum) Frame index
bLength:                           0x1E
bDescriptorType:                   0x24
bDescriptorSubtype:                0x05
bFrameIndex:                       0x01
bmCapabilities:                    0x00
wWidth:                          0x0140 = 320
wHeight:                         0x00F0 = 240
dwMinBitRate:                0x02328000
dwMaxBitRate:                0x02328000
dwMaxVideoFrameBufferSize:   0x00025800
dwDefaultFrameInterval:      0x000A2C2A = 66.666600 mSec (15.00 Hz)
bFrameIntervalType:                0x01
===>Additional Discrete Frame Type Data
dwFrameInterval[1]:          0x000A2C2A = 66.666600 mSec (15.00 Hz)

          ===>Video Streaming Uncompressed Frame Type Descriptor<===
bLength:                           0x1E
bDescriptorType:                   0x24
bDescriptorSubtype:                0x05
bFrameIndex:                       0x02
bmCapabilities:                    0x00
wWidth:                          0x0280 = 640
wHeight:                         0x01E0 = 480
dwMinBitRate:                0x04650000
dwMaxBitRate:                0x04650000
dwMaxVideoFrameBufferSize:   0x00096000
dwDefaultFrameInterval:      0x000A2C2A = 66.666600 mSec (15.00 Hz)
bFrameIntervalType:                0x01
===>Additional Discrete Frame Type Data
dwFrameInterval[1]:          0x000A2C2A = 66.666600 mSec (15.00 Hz)
```

```
          ===>Color Matching Descriptor<===
bLength:                            0x06
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x0D
bColorPrimaries:                    0x01
bTransferCharacteristics:           0x01
bMatrixCoefficients:                0x04

          ===>Video Streaming MJPEG Format Type Descriptor<===
bLength:                            0x0B
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x06
bFormatIndex:                       0x02
bNumFrameDescriptors:               0x04
bmFlags:                            0x01  -> Sample Size is Fixed
bDefaultFrameIndex:                 0x01
bAspectRatioX:                      0x04
bAspectRatioY:                      0x03
*!*ERROR:  bAspectRatioX and bAspectRatioY must be 0 if stream non-Interlaced
bmInterlaceFlags:                   0x00
     D00   = 0  non-Interlaced stream or variable
     D01   = 0  2 fields per frame
     D02   = 0  Field 1 not first
     D03   = 0  Reserved
     D4..5 = 0  Field patterns  -> Field 1 only
     D6..7 = 0  Display Mode  -> Bob only
bCopyProtect:                       0x00  -> Duplication Unrestricted

          ===>Video Streaming MJPEG Frame Type Descriptor<===
          --->This is the Default (optimum) Frame index
bLength:                            0x1E
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x07
bFrameIndex:                        0x01
bmCapabilities:                     0x00
wWidth:                           0x0140 = 320
wHeight:                          0x00F0 = 240
dwMinBitRate:                 0x00119400
dwMaxBitRate:                 0x00119400
dwMaxVideoFrameBufferSize:    0x000012C0
dwDefaultFrameInterval:       0x000A2C2A = 66.666600 mSec (15.00 Hz)
bFrameIntervalType:                 0x01
===>Additional Discrete Frame TypeData
dwFrameInterval[1]:           0x000A2C2A = 66.666600 mSec (15.00 Hz)

          ===>Video Streaming MJPEG Frame Type Descriptor<===
bLength:                            0x1E
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x07
bFrameIndex:                        0x02
bmCapabilities:                     0x00
wWidth:                           0x0280 = 640
wHeight:                          0x01E0 = 480
dwMinBitRate:                 0x001C2000
dwMaxBitRate:                 0x001C2000
dwMaxVideoFrameBufferSize:    0x00003C00
dwDefaultFrameInterval:       0x000A2C2A = 66.666600 mSec (15.00 Hz)
bFrameIntervalType:                 0x01
===>Additional Discrete Frame TypeData
dwFrameInterval[1]:           0x000A2C2A = 66.666600 mSec (15.00 Hz)

          ===>Video Streaming MJPEG Frame Type Descriptor<===
```

```
bLength:                            0x1E
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x07
bFrameIndex:                        0x03
bmCapabilities:                     0x00
wWidth:                           0x0320 = 800
wHeight:                          0x0258 = 600
dwMinBitRate:                 0x00232800
dwMaxBitRate:                 0x00232800
dwMaxVideoFrameBufferSize:    0x00004B00
dwDefaultFrameInterval:       0x000A2C2A = 66.666600 mSec (15.00 Hz)
bFrameIntervalType:                 0x01
===>Additional Discrete Frame TypeData
dwFrameInterval[1]:           0x000A2C2A = 66.666600 mSec (15.00 Hz)

          ===>Video Streaming MJPEG Frame Type Descriptor<===
bLength:                            0x1E
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x07
bFrameIndex:                        0x04
bmCapabilities:                     0x00
wWidth:                           0x0400 = 1024
wHeight:                          0x0300 = 768
dwMinBitRate:                 0x00106800
dwMaxBitRate:                 0x00106800
dwMaxVideoFrameBufferSize:    0x00004B00
dwDefaultFrameInterval:       0x00145855 = 133.333300 mSec (7.50 Hz)
bFrameIntervalType:                 0x01
===>Additional Discrete Frame TypeData
dwFrameInterval[1]:           0x00145855 = 133.333300 mSec (7.50 Hz)

          ===>Color Matching Descriptor<===
bLength:                            0x06
bDescriptorType:                    0x24
bDescriptorSubtype:                 0x0D
bColorPrimaries:                    0x01
bTransferCharacteristics:           0x01
bMatrixCoefficients:                0x04

          ===>Endpoint Descriptor<===
bLength:                            0x07
bDescriptorType:                    0x05
bEndpointAddress:                   0x82  -> Direction: IN - EndpointID: 2
bmAttributes:                       0x02  -> Bulk Transfer Type
wMaxPacketSize:                   0x0200 = 0x200 max bytes
bInterval:                          0x00

          ===>Interface Descriptor<===
bLength:                            0x09
bDescriptorType:                    0x04
bInterfaceNumber:                   0x02
bAlternateSetting:                  0x00
bNumEndpoints:                      0x00
bInterfaceClass:                    0xFE  -> This is an Application Specific USB Device
Interface Class
  -> This is a Device Firmware Application Specific USB Device Interface Class
bInterfaceSubClass:                 0x01
bInterfaceProtocol:                 0x01
iInterface:                         0x05
     English (United States)  "DFU Interface"

          ===>HID Descriptor<===
bLength:                            0x09
```

```
bDescriptorType:                0x21
bcdHID:                       0x000B
bCountryCode:                   0x20
bNumDescriptors:                0x00
```

4.2 Full-speed Configuration

The full-speed Configuration Descriptor is identical to the high-speed descriptor except the Video Streaming interface is removed.

5 Testing

This section describes how to test the webcam on Windows. Two options are available:

- An MM900EV2A module with a front facing OV9657 camera module.

Figure 5.1 MM900EV2A Module

- An MM900EV2A module with a CleO camera module (OV5640).
  This has a modification to the LDOs at U6 and U7 to support voltages of 1.2V and 2.8V for the OV5640 camera module. Exact modifications required are:
  U16: FT531GA (2.8V) replaces FT531EA (2.5V).
  U17: TPS76912DBVR (1.2V) replaces TPS79118DBVR (1.8V). Also remove C109.

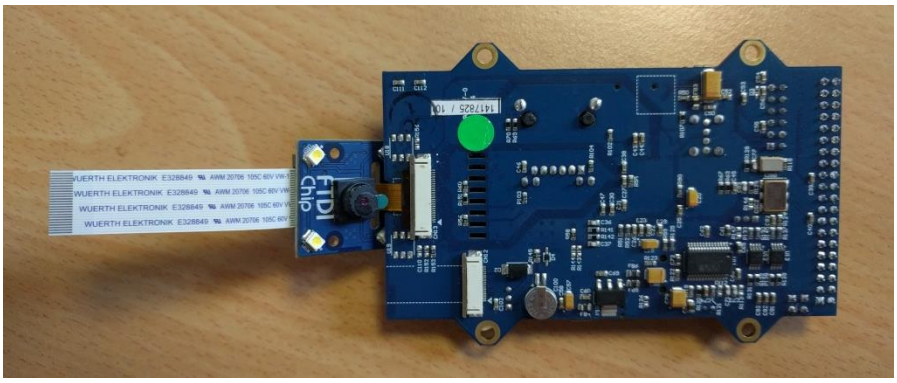

Figure 5.2 MM900EV2A Module with Cleo Camera Module

Each module was connected to and powered by a PC running Windows 7 or Windows 10.

To program the example code onto the FT900 device on the MM900EV2A board, refer to AN_325 FT9xx_Toolchain_Installation_Guide. The binary file name is "AN_414_FT90x_UVC_WebCam.bin" and it is found in the "Debug" directory.

5.1 VLC Media Player

The VLC media player uses Microsoft DirectShow to obtain video from the webcam. To connect to the FT90x webcam select "Open Capture Device…" from the Media menu. The menu item is shown in Figure 5.3.

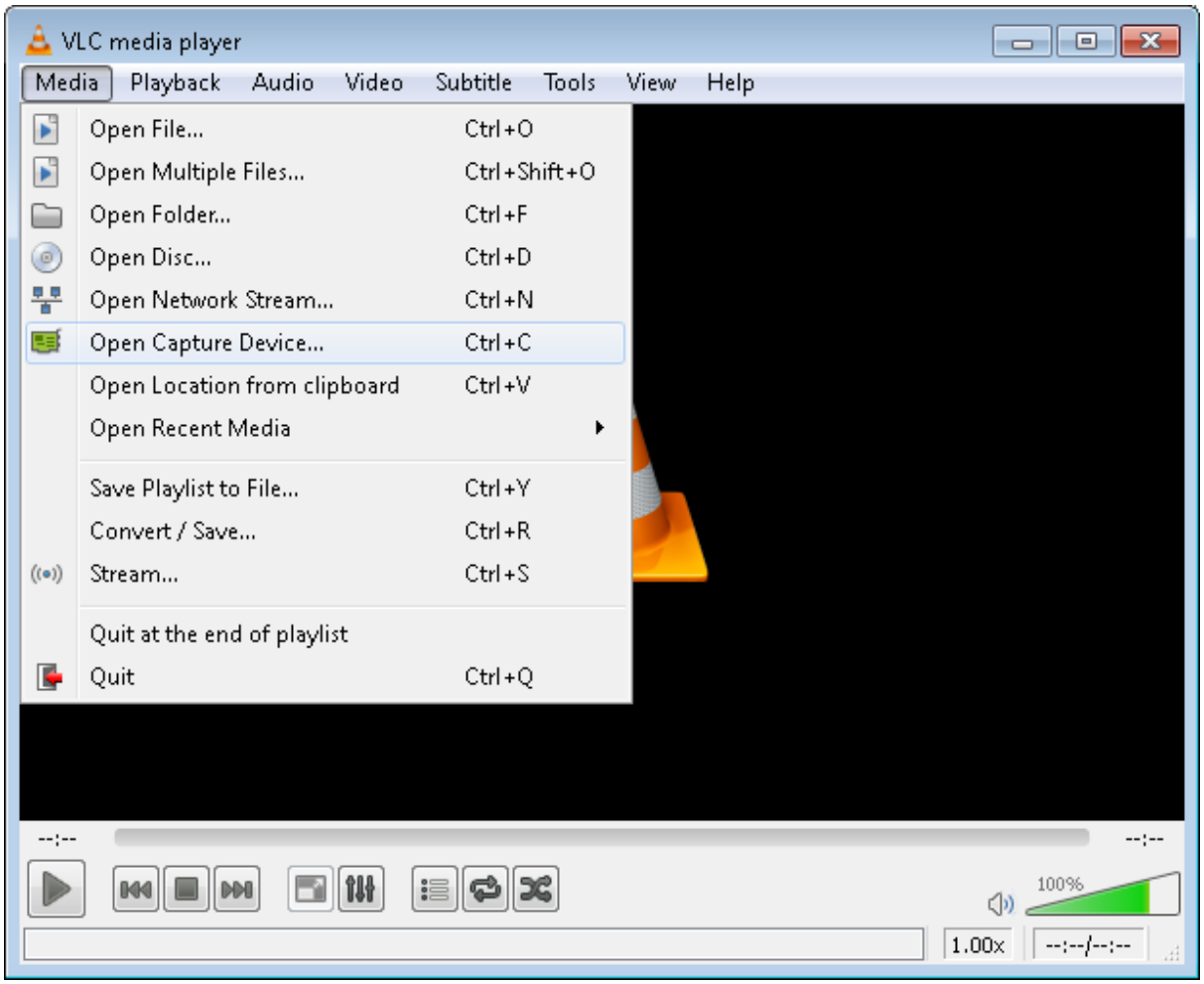

Figure 5.3 VLC Media Menu

The Open Capture Device dialog has a drop-down box to select the device and an option for "Video Size" as shown in **Figure** 5.4. The device name of "FT900 UVC" is used by this application note.

In hi-speed mode any of the three supported resolutions can be used for the video size. If it is left blank then the highest resolution is used.

Valid options are "320x240" for QVGA, "640x480" for VGA and "800x600" for SVGA.

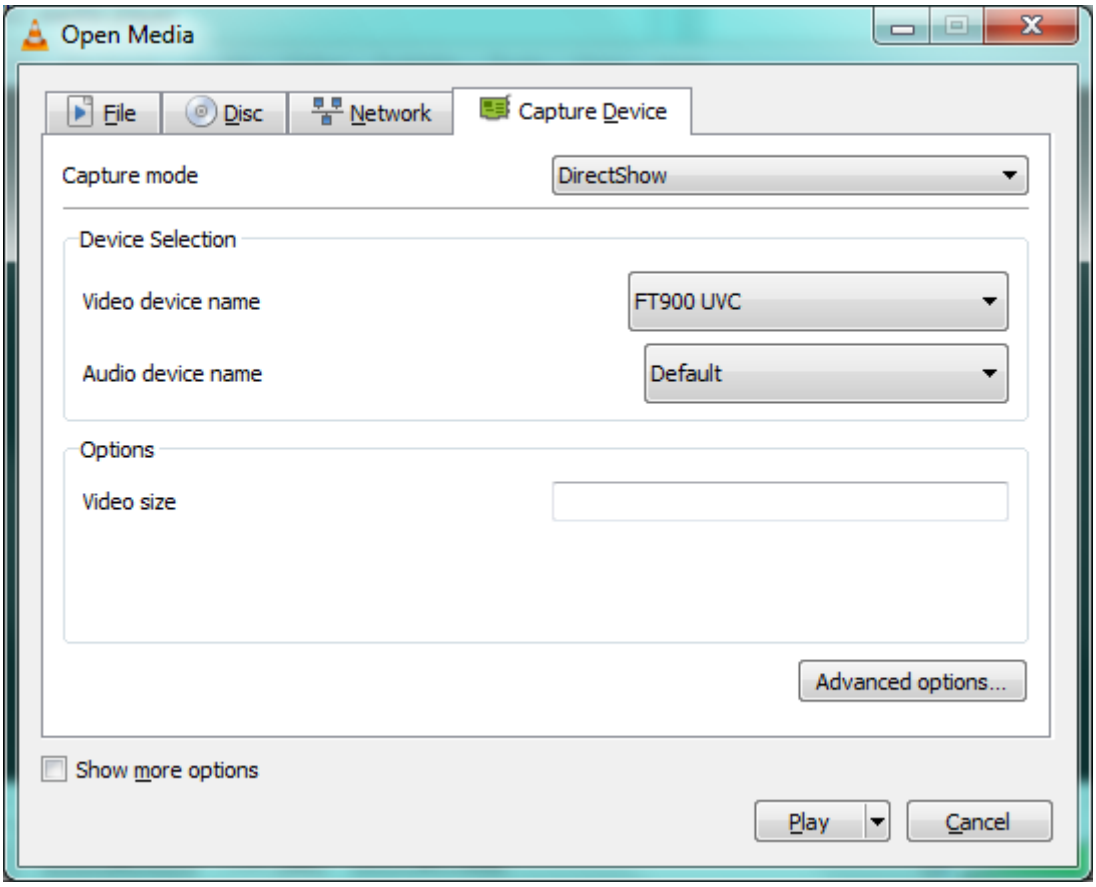

Figure 5.4 VLC Open Capture Device Dialog

5.2 Skype

When the device is selected in Skype it will try to open at VGA resolution (640x480).

To test this select Tools --> Options --> Video settings.

5.3 Windows 10 Camera App

In Windows 10 there is a camera app that will support video from USB webcams.

6 Performance

The following throughput results were obtained. These values are subject to verification and change.

USB hi-speed at VGA resolution: 7MB/sec.

7 Importing into the FT9xx Toolchain

The AN_414 Firmware found at the following link can be easily imported into the FT9xx Toolchain:

http://brtchip.com/ft90x/#FT90x UVC Webcam

Once installed, select File --> Import --> General --> Existing Projects into Eclipse, and point to the downloaded and extracted project directory.

The project will appear in Eclipse Project Explorer as shown in Figure 7.1.

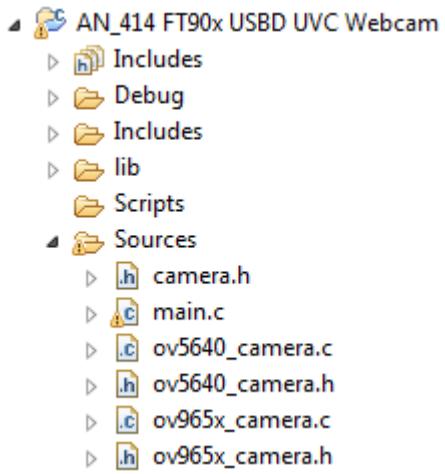

Figure 7.1 UVC WebCam Import

7.1 Changing the Application Software

The application software provided can be altered and changed if required. The FT9xx Toolchain is a free tool to enable code development and debug for the FT90x series and is based on plug-ins for the free popular IDE using the GCC compiler.

With each software change, the project should be rebuilt and reprogrammed into the FT90x IC. Please refer to AN_325 FT9xx Toolchain Installation Guide for further information.

8 Contact Information

Head Quarters – Singapore

Bridgetek Pte Ltd
178 Paya Lebar Road, #07-03
Singapore 409030
Tel: +65 6547 4827
Fax: +65 6841 6071

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office – Taipei, Taiwan

Bridgetek Pte Ltd, Taiwan Branch
2 Floor, No. 516, Sec. 1, Nei Hu Road, Nei Hu District
Taipei 114
Taiwan , R.O.C.
Tel: +886 (2) 8797 5691
Fax: +886 (2) 8751 9737

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Branch Office - Glasgow, United Kingdom

Bridgetek Pte. Ltd.
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) sales.emea@brtichip.com
E-mail (Support) support.emea@brtchip.com

Branch Office – Vietnam

Bridgetek VietNam Company Limited
Lutaco Tower Building, 5th Floor, 173A Nguyen Van Troi,
Ward 11, Phu Nhuan District,
Ho Chi Minh City, Vietnam
Tel : 08 38453222
Fax : 08 38455222

E-mail (Sales) sales.apac@brtchip.com
E-mail (Support) support.apac@brtchip.com

Web Site

http://brtchip.com/

Distributor and Sales Representatives

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

Appendix A – References

Document References

FT90x Datasheet: http://brtchip.com/m-ft9/

UMFT900EV2A Module Datasheet

tinyprintf http://www.sparetimelabs.com/tinyprintf/tinyprintf.php

AN_365 FT9xx API Programmers Manual

AN_325 FT9xx Toolchain Installation Guide

TN_158 What is The Camera Parallel Interface

DFU AN_380 FT900 Bootloader DFU Usage

TN_157 Ethernet Explained

USB and UVC specifications Revision 1.1 http://www.usb.org/developers/docs/devclass_docs/

Universal Serial Bus Device Class Definition for Video Devices

Universal Serial Bus Device Class Definition for Video Devices: Uncompressed Payload

FT9xx Toolchain Download

AN_414 Firmware:

http://brtchip.com/ft90x/#FT90x UVC Webcam

FT900 Software Examples: http://brtchip.com/ft90x/

Acronyms and Abbreviations

Terms	Description
DFU	Device Firmware Update
USB	Universal Serial Bus
UVC	USB Video Class
SCCB	Serial Camera Control Bus
JPEG/JPG	Joint Photographic Experts Group
VGA	Video Graphics Array (640 x 480 resolution)
QVGA	Quarter VGA (320 x 240 resolution)
SVGA	Super VGA (800 x 600 resolution)
UART	Universal Asynchronous Receiver and Transmitter (Serial Port)

Appendix B – List of Tables & Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: AN_414 FT90x UVC WebCam
Document Reference No.: BRT_000103
Clearance No.: BRT#061
Product Page: http://brtchip.com/m-ft9/
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2016-07-07
1.1	Updated Release Dual branding to reflect the migration of the product to the Bridgetek name – logo changed, copyright changed, contact information changed	2017-02-17
1.2	Updated AN_414 firmware download links to be generic since they were broken in version 1.1.	2017-04-04