Application Note

AN_432

FT90x Ethernet Video Bridge

Version 1.2

Issue Date: 2017-06-19

This application note describes an Ethernet Video Bridge implemented on the FT900. It uses lwIP for transferring data between a webcam on a server device and a USB UVC client on a client device. The RTSP is used to transfer the data on the network.

Table of Contents

Copyright © Bridgetek Pte Ltd

1 Introduction

The FT900/FT901 and FT905/FT906 devices include an Ethernet interface and camera interface. This can be used with a suitable networking stack to allow the device to be networked. This application note shows how the Ethernet interface can be used to transfer data received from the camera interface from one FT90x module to another. The received data being sent to a host device via a USB UVC interface.

1.1 Overview

The camera interface on one FT90x module can be used to stream video to another FT90x module. This effectively makes a simple closed-circuit video system. If the client device is connected *via* USB to a host PC then it will show up as a standard UVC webcam.

The camera interface supports the Omnivision OV5640 module and OV9655. The best results are obtained using the OV5640 camera module's built-in MJPEG compression.

The network interface supports DHCP to obtain configuration information or it can be given a fixed configuration. The network configuration is stored on external EEPROM, so that they are not lost after power off.

Third-party open source code is used to implement a TCP/IP stack in this application note:

- TCP/IP - lwIP (LightWeight IP) library which has been ported to the FT90x.
- Printf – tinyprintf.

Links to resources for these libraries are in Appendix A – References.

There are 2 FT90x modules required for this application note. The connections between them and to external devices are as shown in Figure 1.1.
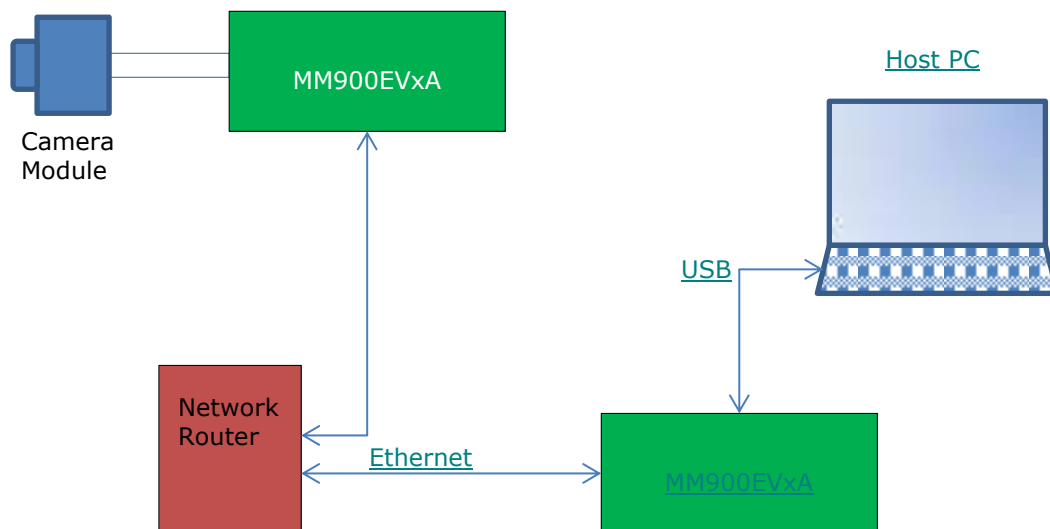
Figure 1.1 Application Note Diagram

1.2 Scope

Both firmware applications for the client and server assume a fixed camera module, resolution, and frame rate. There is no mechanism for the client to change the data streamed from the camera module by the server.

The external I^{2}C EEPROM used in this example is the 24AA02E48T-I/OT EEPROM which is a 2K serial EEPROM device. This memory is present on the MM900EVxA , excluding the MM900EV-LITE.

1.2.1 Features

The application note highlights the use of lwIP to provide a TCP/IP stack and to implement an arbitrary client/server protocol. The "arch" folder of lwIP in the source code for the application note contains the architecture-specific parts of lwIP required for the FT90x.

1.2.2 Enhancement

Enhancements to this application might include:

- Allow the client to change the properties of the video streamed by the server.
- Add in UPNP functionality to allow the client to discover the server.
- Improve RTP and RTSP support to allow third-party access to the server without the client being required. In general this is close to being supported; the video encoding support on the majority of clients does not cover the uncompressed or compressed output of the camera module.
- Allow only permitted IP addresses to connect to the device.
- Reduce the overall size of the application code.

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

http://brtchip.com/wp-content/uploads/FT90x/AN_432_FT90x_Ethernet_Video_Bridge.zip

See section 5 Importing into the FT9xx Toolchain for more information.

2 Project Overview

The project files for the client and server applications are divided into the following folders.

Folder	Description
Source	Application source code and abstraction files.
httpdfs	File system for web server.
lib	Library files.
lib\tinyprintf	tinyprintf library.
lib\lwip	lwIP library.

Table 2.1 Project Files Overview

The application source code is contained within the "Sources" folder.

In the server code, the main() function and the high-level functions of the application are in the "server.c" file; in the client code they are in the "client.c" file.

The network interface for both applications is within "net.c". The server has abstraction code for the OV9655 module in the "camera_ov965x.c" file and for the OV5640 in the "camera_ov5640.c". The abstraction file and the camera support are set with the macro CAMERA_TYPE in both applications. Only certain resolutions are supported for each camera module: PAYLOAD_TYPE and FRAME_TYPE set compressed or uncompressed modes and the camera resolution respectively.

2.1 Server Main Program

The server main program is responsible for receiving the video data stream from the camera module, performing any formatting required for network transmission, then sending the data to the Ethernet interface.

2.2 Client Main Program

The client will implement a USB UVC class device and when requested by the host will transfer data received from the Ethernet interface to the USB device interface.

2.3 Network Code

The networking layer is provided by the lwIP library which has been ported to support the FT90x. The lwIP code is available separately from Bridgetek with the FT90x architecture extensions.

2.3.1 Network Abstraction

The network interface is abstracted from the main project in the file "net.c". All lwIP features which refer to an interface (netif) are passed through this layer.

The lwIP network interface is initialized and configured using data supplied from the main application. The function which processes incoming packets for the network interface within lwIP is called from the net_tick() function.

2.3.2 lwIP Library

The library is available with a BSD-style license. Version 2.0.0 of the code is used. No changes have been made to the code from the lwIP project page.

The FT90x architecture code is responsible for obtaining packets from the Ethernet interface and transmitting packets to the Ethernet interface.

The application makes extensive use of the callback methods in lwIP to permit sending and receiving of data.

2.3.3 Network Transmission Protocols

The Real-Time Streaming Protocol (RTSP) is used to initiate data streaming from the client to the server using RFC 2326. This protocol closely resembles the HTTP and the server code for RTSP is based on the lwIP HTTPD application code. When streaming the UDP-based Real-time Transport Protocol (RTP) is used to transfer data from the server to the client using the formats from RFC 3550.

Only one client may connect to the server at any one time.

2.4 Other Features

The DFU-C facility can be enabled for this application. This is called 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.

3 Code Structure

Data sent and received from the network are handled by the lwIP library.

The FT90x API is used to receive data from the camera interface and to send data *via* the USB device interface to the host. Various other calls are used to supporting interrupts and a UART monitor/debugging interface.

More information on these FT90x API functions can be found in AN_365 FT9xx API Programmers Manual.

3.1 Connection

The client initiates the connection with the server using RTSP messages. These are sent *via* TCP after a connection is made to the server. The server code responds to these messages with acknowledgements. There is a state machine in the main code of the client and the server to keep track of the status of the connection.

3.2 Streaming

After a connection has been made then streaming is started from the server to the host *via* UDP. This consists of UDP packets sent from the server to the client containing video data and occasional feedback packets from the client to the server. The later act as keep-alive packets and are only used by the server to ensure that the client is accepting data. The UDP is connectionless so a feedback mechanism is used to ensure the client is accepting data from the server.

3.2.1 Server Code

The server code for obtaining the data stream from the camera interface is identical to the method used in AN_414 FT90x UVC WebCam. An interrupt service routine stores the data stream in a series of buffers, the main code polls the buffers for available data and then formats it before transmitting it to the RTP layer. A protocol handler in the RTP layer adds headers and transmits the data over the Ethernet interface.

A timer is used to stop transmitting to the client if an RTP feedback packet has not been received within a set period.

3.2.2 Client Code

The client code for sending the data stream to the host via USB is identical to the method used in AN_414 FT90x UVC WebCam. A USB UVC device is instantiated which tells the host PC that it is a webcam that transmits video data. The main code receives packets from the server module and will buffer and assemble them into video frames to send to the host PC.

The client periodically sends an empty RTP feedback message to signal that it is still requiring the video stream from the server.

4 Configuration

This section describes the default configuration used by the application and how to change the configuration.

The client and the server code must be configured with an IP address. DHCP can be enabled, allowing the application to request an IP address when it is first started. However, the client must know the IP address of the server to initiate the connection.

The MAC address is obtained from the onboard 24AA02E48T-I/OT EEPROM during initialization (see 'net_init' in net.c). This memory is present on the MM900EVxA , excluding the MM900EV-LITE. The pre-programmed permanently protected MAC address resides at the last 6 bytes of memory in the EEPROM.

Note: The code should be changed if you are using hardware that does not have this external EEPROM.

4.1 Network

The network addresses are fixed on both the client and the server for simplicity.

Two macros "FIXED_IP_ADDRESS" and "FIXED_IP_ADDRESS_ALT" are used to select the IP address. The code which sets this is in the video_server() function on the server and video_client() function on the client. To change the settings, alter the IP Addresses set by the code.

```
#if defined(FIXED_IP_ADDRESS)
        IP4_ADDR(&ip_addr, 192,168,1,2);
        IP4_ADDR(&gw_addr, 192,168,1,1);
        IP4_ADDR(&mask_addr, 255,255,255,0);
#elif defined(FIXED_IP_ADDRESS_ALT)
        IP4_ADDR(&ip_addr, 172,16,3,251);
        IP4_ADDR(&gw_addr, 172,16,2,5);
        IP4_ADDR(&mask_addr, 255,255,252,0);
#else
        IP4_ADDR(&ip_addr, 0,0,0,0);
#endif
```

The IP Address of the server is specified later in the same function on the client. This must match the configured IP address of the server.

```
#if defined(FIXED_IP_ADDRESS)
        IP4_ADDR(&ip, 192, 168, 1, 3);
#elif defined(FIXED_IP_ADDRESS_ALT)
        IP4_ADDR(&ip, 172,16,3,250);
#else
        IP4_ADDR(&ip, 0,0,0,0);
#endif
```

If DHCP was enabled then the client would need to be informed of the server's address either using DNS or uPNP.

4.2 Video

The camera interface may be set to use either an OV5640 module or an OV9655 module. These have different capabilities and some options are not supported. The OV9655 is available on the. MM900EVxA modules.

The OV5640 can do compressed (MJPEG) video output which greatly decreases the amount of data to be streamed for a given resolution and frame rate. Both modules support uncompressed data.

The size of uncompressed data somewhat limits the resolutions which can reliably be transmitted though and QVGA (320x240) resolution is recommended for uncompressed video.

The quality and reliability of the transferred data relies on the underlying network between the client and server modules. **MJPEG data transmission is strongly recommended.**

The target camera module is selected with the macro CAMERA_TYPE and can be set to the OV9655 module:

#define CAMERA_TYPE CAMERA_OV965X

Or the OV5640 module:

#define CAMERA_TYPE CAMERA_OV5640

The video resolutions can be changed with the macros PAYLOAD_TYPE and FRAME_TYPE. For example, uncompressed QVGA:

```
#define PAYLOAD_TYPE CAMERA_FORMAT_UNCOMPRESSED
#define FRAME_TYPE CAMERA_MODE_QVGA
```

Or compressed VGA:

```
#define PAYLOAD_TYPE CAMERA_FORMAT_MJPEG
#define FRAME_TYPE CAMERA_MODE_VGA
```

There are guards to set the default for the OV9655 to uncompressed QVGA and the OV5640 to compressed VGA.

4.3 UART

The following settings are used by default for the UART which displays debug messages:

- 19200 baud
- 8 bits data
- 1 stop bit
- no parity

4.4 Hardware Setup

Figure 4.1 shows the general hardware setup required to run the Ethernet Video Bridge.

If the OV5640 camera is required, this would need to be sourced and connected externally to the MM900EVxA. The CleO Camera uses this camera type. Note that the LDO's on the MM900EVxA do not support OV5640 so some hardware modification would be required for this module type.
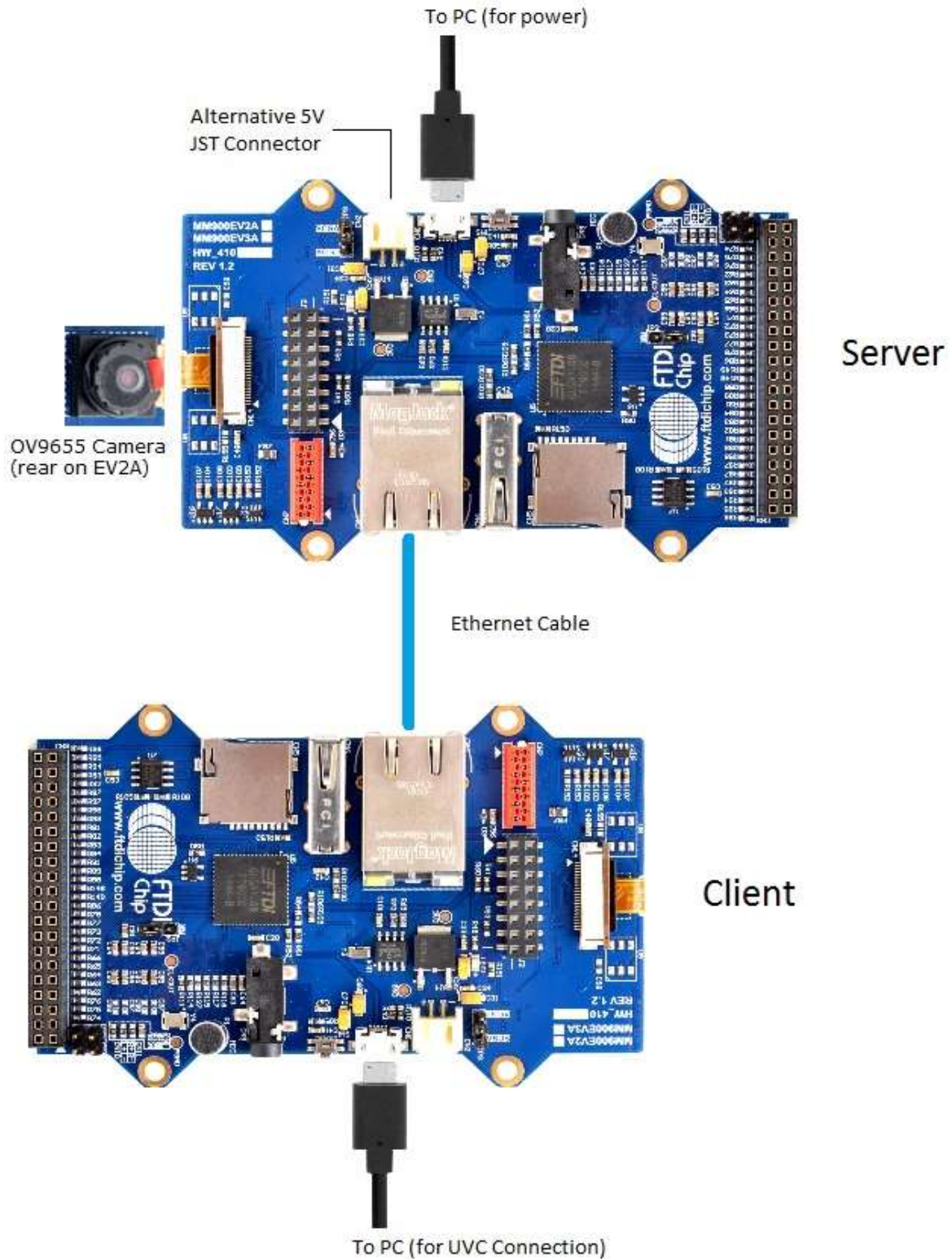
Figure 4.1 Hardware Setup

5 Importing into the FT9xx Toolchain

The AN_432 Firmware found at the following link can be easily imported into the FT9xx Toolchain:

AN_415_FT90x_Ethernet_to_GPIO_Bridge.zip

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 5.1.
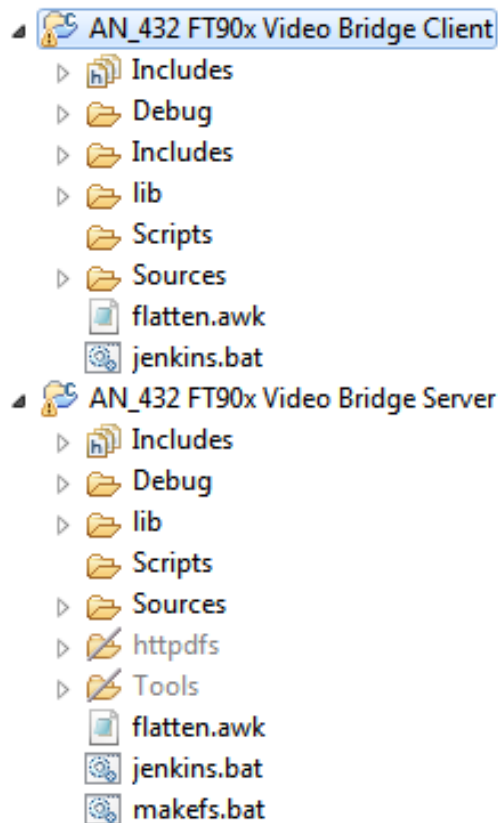
Figure 5.1 Eclipse Project Structure

5.1 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.

6 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@brtchip.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](#)

lwIP http://savannah.nongnu.org/projects/lwip/

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

[AN_325 FT9xx Toolchain Installation Guide](#)

[AN_365 FT9xx API Programmers Manual](#)

[AN_414 FT90x UVC WebCam](#)

DFU [AN_380 FT900 Bootloader DFU Usage](#)

[TN_157 Ethernet Explained](#)

[EasySYNC USB Oscilloscope](#)

[FT9xx Toolchain](#)

[AN_415_FT90x_Ethernet_to_GPIO_Bridge.zip](#)

[24AA02E48T-I/OT EEPROM](#)

[MM900EVxA](#)

Acronyms and Abbreviations

Terms	Description
CGI	Common Gateway Interface
DFU	Device Firmware Update
EEPROM	Electrically Erasable Programmable Read Only Memory
GPIO	General Purpose Input Output
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
I2C	Inter-Integrated Circuit
MAC	Media Access Controller
SSI	Server Side Include
USB	Universal Serial Bus
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_432 FT90x Ethernet Video Bridge

Document Reference No.:   BRT_000073

Clearance No.:   BRT#054

Product Page: http://brtchip.com/m-ft9/

Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2017-01-23
1.1	Updated release under BRT Chip	2017-04-04
1.2	Minor edits to reflect code enhancements	2017-06-19