

Wi-Fi Based Intelligent Image Sensing on Concerto MCU

SHEBI AHAMMED S , BINU C PILLAI

Dept of Communication Engineering
Amaljyothi College of Engineering
Kanjirapally, India

shabips@gmail.com,binucpillai@amaljyothi.ac.in

RAJESH R

Senior Engineer
SEG, CDAC
Trivandrum, India
rrajesh@cdac.in

Abstract— Wi-Fi is an industry standard way to connect computers over a network without wire, in other words, a wireless network. It is widely used in consumer electronics, industrial control and network monitoring. In this paper, we will introduce a real-time wireless image transmission system. In the present systems uses two separate processors for image acquisition and communication subsystems. Here we use a single Concerto chip instead of two processors. We transmit images with variable size. This system is based on SimpleLink CC3000 Wi-Fi Module and F28M35x (Concerto) MCU which controls the wireless transceiver. The Concerto is a multi-core system-on-chip microcontroller (MCU) with independent communication and real-time control subsystems. The data rate of the system is 11Mbps at least and 54Mbps at most. The Wi-Fi module uses infrastructure mode network of IEEE802.11g. Wi-Fi is connected through SPI (Serial Peripheral Interface Bus) to form the relatively energy saving wireless transmission. The common constrains that we are facing during the design and implementation of Wi-Fi networks are security issues, low data rates, distance problem, hidden node problems and error rate. These negative aspects can be simplified by using SimpleLink CC3000 Wi-Fi Module.

Keywords—Wireless-LAN;ConcertoMCU;Wi-Fi Module

I. INTRODUCTION

Wi-Fi has become a very widely used technology. In this technology, a wireless network can use an access point, or base station. In this type of network the access point acts like an active hub to provide wireless connectivity between the computers. Embedded technology is another mature technology in the field of consumer electronics and industrial control. Embedded wireless network technology is becoming a part of people's life. In this article, we introduce a imaging system which is combined with wireless technology, embedded technology and image processing technology. It uses the 802.11b/g works in the 2.4GHz band and supports the transmission data rates of 11 Mbps and 54Mbps. It can achieve a high-speed wireless data communication.

The present image tracking platforms are normally implemented with two processors; one for image acquisition and other for communication subsystem [1]. Above mentioned systems are having very high power loss, numerous interconnections, low BER and high cost. Due to these disadvantages we cannot afford the present platform and is replaced by Concerto platforms [6].

As Figure 1 shows, the data arrives at PC mainly through three parts. The most important part is the camera part. It gets the data from the image sensor and sends these data to the Wireless-LAN. The wireless router receives these data and sends them to PC. The communication between the router and PC machines can be wired or wireless. All the network communications are bi-directional. The computer controls the image's size by sending data packets to adapt to different environments. Some parameters of the image just like the gray value and bright, are also controlled by the PC.

Efforts have been made to improve data transfer speed of image. The improvement is greatly attributed to the rapid development of digital circuits and electronic technologies. Existing image transmission systems are mostly designed using digital signal processors (DSPs) but the implemented data rate can still fail to meet the technical definition.

This paper presents an Wi-Fi and Concerto MCU based system for performing a high-speed image data transmission that is competent to transfer all sorts of imaging data in real-time. The Concerto is a multi-core system-on-chip microcontroller (MCU) with independent communication and real-time control subsystems.

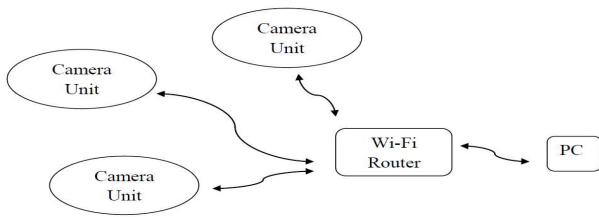


Figure 1. System Component Diagram

We briefly highlight some of the work done in the imaging system in the context of Wi-Fi network. Mingwei Li [1] discussed imaging system based on wireless embedded camera implemented on a WLAN networks. They presented some of the security issues. They used a sensor, a microprocessor and a RF chip and VNT6656 as Wi-Fi module. A processor with high performance in time control to drive the sensor just like FPGA. Another processor which have network protocol stack like ARM, it will be easy to join a network. But the sensor will work with an inaccurately clock and even hard to normally. They used SAM SUNG S3C2410 CPU core is a 16/32 bit ARM920T RISC as main processor and an FPGA chip is a Xilinx's Spartan-3E as a sensor processor. One SPI channel is used to send command to FPGA. All interfaces of ARM is driven by Linux driver program and controlled by application program in ARM.

To the best of our knowledge, the work being presented in this article is unique in providing significant, relevant, and practical information on the Wi-Fi based imaging system on concerto platform. The advantage of system we can replace both FPGA and ARM by a single concerto chip. The Concerto is a multi-core system-on-chip microcontroller (MCU) with independent communication and real-time control subsystems. The communications subsystem is based on the industry-standard 32-bit ARM Cortex-M3 CPU and control subsystem is based on TMS320C28x DSP[6]. Instead of VNT6656 we use CC3000 Wi-Fi module as a communication device that provides WPA2 based data protection and 802.11g supporting WLAN communication [7].

II. HARDWARE ELEMENTS

As is said above, our system has three parts. The wireless router and the PC is the existing platform. The camera is what we need to do the hardware design. We can use several options to design the camera, but it must have a sensor, a microprocessor and a RF chip at least. If we use a processor with high performance in time control to drive the sensor just like DSPs, it

will be difficult to control the RF chip to join a wireless network. If we use a processor which have network protocol stack like ARM, it will be easy to join a network. But the sensor will work with an inaccurately clock and even hard to normally. At last we used an ARM as a main processor and an DSP as a sensor processor. In this system we use the concerto MCU which is a dual core processor unit having both ARM Cortex M3 and TMS320C28x DSP, as Figure 2 shows.

A. Image Acquisition

The image acquisition part includes an image sensor and an TMS320C28x DSP in Concerto MCU. The image sensor we used is MT9M032 which is a 1/4-inch format CMOS active-pixel digital image sensor [5]. It is controlled by C28x to capture image data and send these data to PC. The TMS320C28x chip is the part of Concerto MCU. Its highest frequency is 150MHz. It can meet the requirements of realtime data acquisition and transmission. On one hand, this C28x receives command from SPI interface and control the image sensor to get different kinds of images. On the other hand, it sends the image data to ARM Cortex M3 through IPC interface. The IPC interface is widely used in communications between different CPU. In our system, IPC and SPI works in asynchronous communication mode and has 32-bit data lines and 32-bit address lines independently. C28x generates the read, write, address and other control signals for IPC interface. Then data given to CC3000 Wi-Fi module via SPI interface from CC3000 module data transmitted through AP in WLAN network to PC.

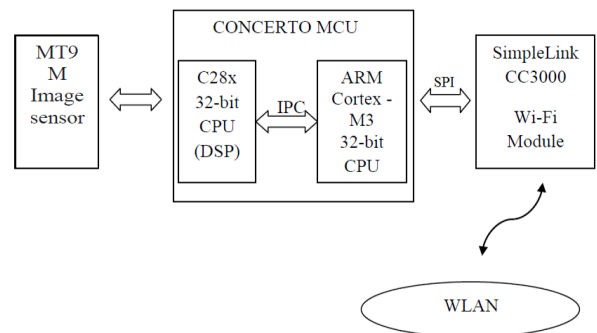


Figure 2. Block Diagram of the Wi-Fi based Imaging System

Regardless of the technology of image acquisition (CCD or CMOS), electronic image sensors must capture incoming light, convert it to electric signal, measure that signal, and output it to supporting electronics. Similarly, regardless of the technology of image acquisition, cinematographers can generally agree on a

short list of capabilities that a capture medium needs in order to provide great images for big-screen feature films: capabilities such as Sensitivity, Exposure Latitude, Resolving Power, Color Fidelity, Frame Rate, and one we might call "Personality."

B. Concerto MCU

The Concerto is a multi-core system-on-chip microcontroller (MCU) with independent communication and real-time control subsystems [6]. The communications subsystem is based on the industry-standard 32-bit ARM Cortex-M3 CPU and features a wide variety of communication peripherals, including Ethernet, USB OTG with PHY, CAN, UART, SSI, I2C, and an external interface. The real-time control subsystem is based on TI's industry-leading proprietary 32-bit C28x Floating-Point CPU and features the most flexible and high-precision control peripherals, including ePWMs with fault protection, and encoders and captures.

In addition, the C28-CPU has been enhanced with the addition of the Viterbi, Complex Math, CRC Unit (VCU) instruction accelerator that implements efficient Viterbi, Complex Arithmetic, 16-bit FFTs and CRC algorithms. A high-speed analog subsystem and supplementary RAM memory is shared, along with on-chip voltage regulation and redundant clocking circuitry. Safety considerations also include Error Correction Code (ECC), Parity, and Code Secure Memory, as well as documentation to assist with system-level industrial safety certification.

The ARM Cortex-M3 architecture is used widely throughout the industry for host communications and has a large ecosystem of tools and software. Cortex-M3 is also well-established as a proven platform for developing advanced human-machine interfaces (HMI) and graphical user interfaces (GUI). Similarly, the C28x core is the industry's leading platform for control applications of all types. The C28x architecture has been optimized to efficiently and reliably manage complex control algorithms with its integrated floating-point processor and streamlined memory architecture that exceeds the capabilities of more general-purpose cores. It also offers best-in-class control peripherals to provide the highest efficiency, accuracy, and performance. In our camera, we use SPI to connect with CC3000 Wi-Fi Module to send data to the Wireless-LAN or receive command from the PC. One IPC channel is used to send command to TMS320C28x DSP. All interfaces of ARM Cortex M3 is driven by Code composer studio V5 C-Program and controlled by application program in ARM Cortex M3.

C. SimpleLink CC3000 Wi-Fi Module

We use a SimpleLink CC3000 Wi-Fi Module in our system to connect with the Wireless-LAN [7]. The SimpleLink CC3000 module is a self-contained wireless network processor that simplifies the implementation of Internet connectivity. SimpleLink CC3000 Wi-Fi solution minimizes the software requirements of the host microcontroller (MCU) and is thus the ideal solution for embedded applications using any low-cost and low power MCU. The SimpleLink CC3000 module reduces development time, lowers manufacturing costs, saves board space, eases certification, and minimizes the RF expertise required.

The main features of SimpleLink CC3000 Wi-Fi Module are as follows:

- 802.11b/g integrated radio, modem, and MAC supporting WLAN communication as a BSS station with CCK and OFDM rates from 1 to 54 Mbps in the 2.4-GHz ISM band.
- Auto-calibrated radio with a single-ended 50-Ω interface enables easy connection to the antenna without requiring expertise in radio circuit design.
- Advanced connection manager with seven user-configurable profiles stored in an NVMEM allows automatic fast connection to an access point without user or host intervention.
- Supports all Wi-Fi security modes for personal networks: WEP, WPA, and WPA2 with on-chip security accelerators.
- Smart Configure WLAN provisioning tools allow customers to connect a headless device to a WLAN network using a smart phone, tablet, or PC.
- Integrated IPv4 TCP/IP stack with BSD socket APIs enables simple internet connectivity with any microcontroller, microprocessor, or ASIC.
- Supports four simultaneous TCP or UDP sockets
- Built-in network protocols: ARP, ICMP, DHCP client, and DNS client enable easy connection to the local network and to the Internet.
- Interfaces over 4-wire serial peripheral interface (SPI) with any microcontroller, or processor at clock speed up to 16 MHz
- Low footprint driver provided for Concerto MCUs and easily ported to any processor or ASIC
- Simple APIs enable easy integration with any single-threaded or multi-threaded application.

In the camera, CC3000 Wi-Fi Module is driven by ARM Cortex M3 embedded system. It is used to accomplish the communication between the ARM Cortex M3 and wireless router.

III. SOFTWARE ARCHITECTURE

The images we get from the sensor are gray images and we transmit them to the PC computer through wireless LAN. The size of the image is variable with the minimum size is 320*240 pixels and the maximum is 800*600 pixels. We use one byte to represent a pixel in the image [1]. The signals of the image sensor are analog signals. We put these signals to analog inputs of TMS320C28x one by one and then we will get an 8 bit data for each pixel. The TMS320C28x transmit these data to the ARM Cortex M3 by IPC interface. As the width of the local bus interface is 32 bit, the TMS320C28x can write 2 pixels to the ARM Cortex M3's RAM one time. After one whole image is written to the RAM, the TMS320C28x will send a signal to the ARM Cortex M3 by IPC bus. The ARM Cortex M3 can send this image CC3000 Wi-Fi Module via SPI interface then to the PC through wireless LAN.

The software of data transmission can be divided into two parts: one is setting up wireless network; the other is transmitting data in this network. The first part is done by the driver program of wireless module and the Windows operation system. The driver and the operation system package the image data into the 802.11 MAC frame. IEEE 802.11 has three kinds of MAC frames: data frame, management frame and control frame. The data frame is the most frequently used. The main part of frame is data field whose maximum length is 2304 bytes and minimum length is 0. It is responsible for transmitting upper layer data between workstations. Different from the wired network, the physical channel of wireless network is special that all wireless devices communicate with each other in the same medium. As the frequency is same, different Access Points (AP) need to distinguish their node by authentication and they are encrypted to keep other nodes away. In our system, the router and the Wi-Fi module both use the WPA2 64 bit encryption.

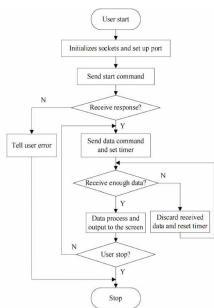


Figure 3. Flow chart of the communication in PC

The Code Composer Studio running on the ARM Cortex M3 is V5. Network hardware is registered in the CCS V5 in the form of "net device" structure. Network device driver connect the core functions and hardware by filling in the members of "net device" structure. Each device must have one structure like this. All the structure of network equipment will be added to a list beginning with "dev base". In OS, the entire network can be managed by the same commands. We can find a new network named "eth 1" after loading the driver program in OS. We use the "ifconfig" command to configure the state of network card such as the IP address and gateway. The IP of the camera must be unchanged. The related contents of wireless network, such as the BSSID of access point, password, communications channel, the transmission data rate are configured by using "iwconfig". After these operations we can join in the Wireless-LAN and communicate with the wireless router. The application program is written by C language. This part lets the data to be transmitted in the network. ARM Cortex M3 embedded system and PC communicate with each other using the UDP protocol. Before the communications start, PC and ARM Cortex M3 should have completed their initialization.

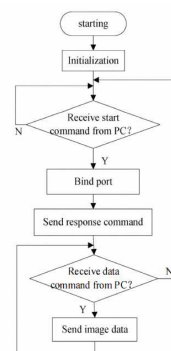


Figure 4. Flow chart of the communication in ARM Cortex M3

User controls the host PC to start the communication process by sending a start command to the ARM Cortex M3 with a fixed IP. After receiving the response signal, the image transfer process begins. If the PC doesn't receive any response from the network, we would think that there is no camera in the network with this IP or the network is something wrong and can't be used for transmitting data. The communication will stop. Of course, we can start again later. The first packet we let the program on PC send to the ARM Cortex M3 in Concerto MCU is the parameters of the image getting from the user panel, such as the size, brightness and so on. Then PC sends a command asking ARM Cortex M3 for data and set the timer on. A fixed number of data will be sent to the PC. We will discuss the question in next section that how much data is sent to PC one time. If the

command from PC to ARM Cortex M3 is lost in the network or the image data from ARM Cortex M3 does not arrive at the PC, the timer of the PC will time out and generate an interrupt message. The PC would discard the received data of this time and send a command ask ARM Cortex M3 for data once again. This process will be done 10 times before the communication stops itself. The flowchart of communication in PC is shown as Figure 4.

On the other hand, the ARM Cortex M3 creates a thread to wait for the start command of PC after the initialization. If it gets a command from an IP in network, a response will be sent to this IP. Once the ARM Cortex M3 connects with one PC, it can not receive start command again from other computers before it comes back to the waiting start status. Then, ARM Cortex M3 gets the parameters and sends some of the parameters to the TMS320C28x. Every time the ARM Cortex M3 receives a data command, it will send a fixed number of data to the PC immediately. The ARM Cortex M3 is always in a waiting status from beginning to the end except sending data. If it does not receive a command in the predetermined time, it will cognize the communication is finished and comes back to the status of waiting for start command. At the same time, PC can't receive data before restarting. The flowchart of communication in ARM Cortex M3 is shown as Figure 5.

IV. CONCLUSION

The data rate can be chosen in IEEE 802.11g as 1 Mbps to 54 Mbps. This work is done after enabling the wireless network. Usually, we should choose the fastest data rate to make the transfer faster. Actually, we have to consider the lost rate in the network and other factors. After the experiment, we decide to use 24Mbps artificially. It makes our system running in the best status.

The camera communicates with the PC through the Wireless-LAN. Many factors, such as distance and noise, will affect the data rate or the stability of the system. Sometimes we have to reduce the data rate to make the communication more stable. We test the system indoors to get the relation among the data rate, stability and buffer size. The data we put into the send buffer every time has a large effect on the stability of the system. If the data is so much that we can't put them all in the send buffer one time, the network will not work normally and the transmission will stop. If we use smaller packet, the times we operate buffer must be more. If we use larger packet must divide them into small UDP packets when sending in the network. These are all time-consuming. Otherwise, if we use large packet we also need to prolong the timeout value in the PC because the sending time is longer. The resend time will much longer too. In our system, we put 7680 bytes data into the buffer one time. It can make the communication be most stable, especially in low data rate.

Our system gathered image data by TMS320C28x, and sends them to the PC through the ARM Cortex M3-CCSV5 embedded system. Effective data rate is about 11Mbps. If the size of image is 100 thousand pixels and the image is uncompressed, we can send 3-4 frames per second. If we compress the image in ARM Cortex M3 embedded system with the JPEG standard [8], we can send 30 frames with the size of 640x480 per second at least. This can meet the requirements of monitor. It's worth noting that we obtain this result in doors. If the camera works outdoors, especially far from the router, it need a higher receive sensitivity. We can slow down the data rate to make our communication stable if necessary. We also need to know the coverage of the wireless router.

REFERENCES

- [1] Mingwei Li and Zhibo Li, "Design of Embedded Wireless Network Camera," 10.1109/ICAWST.2011.6163137, Awareness Science and Technology (iCAST), Sep 2011.
- [2] Sanjeev Dhawan, "Analogy of promising wireless technologies on different frequencies: Bluetooth, WiFi, and Wi MAX", Wireless Broadband and Ultra Wideband Communications, pp. 14,2007.
- [3] Xiufeng Feng, "Research of wireless network setting up based on 802.11", Information Science, pp. 91,133, April 2010
- [4] LI Hao, GAO Ze-hua and GAO Feng, "Research on IEEE 802. 11 wireless local area networks standards", Application research of computer, vol. 26, pp. 1616-1620, May 2009.
- [5] VNT6656 datasheet, <http://www.viatech.com.cn/>. Revision 1, October 17, 2006.
- [6] F28M35x datasheet, <http://www.ti.com/SPRS742D>-June 2011- Revised August 2012.
- [7] CC3000 Module datasheet, <http://www.ti.com/SWRS119>- Revision 1, December 2011
- [8] Hu Dong, "The basic method and international standards of encoding still images", Beijing University of Posts and Telecommunications Press, December 2003.

Shebi Ahammed S, is currently a post graduate student in Communication Engineering, Mahatma Gandhi University, India. He received the Bachelor of Technology Degree from Department of Electronics and Communication Engineering, Mahatma Gandhi University, India in 2008. He was Jn. Telecom Engineer from 2009-2010 in Saudi Telecom Co-operation, Saudi Arabia. His current research interests include wireless communication, digital signal processing and communication networks.

Binu C Pillai, is currently working as Assistant Professor in Amaljyothi College of Engineering, Kerala, India. He received the Bachelor of Technology degree from Department of Electronics and Communication Engineering, Kannur University, India in 2006 and Master of Technology degree from VLSI Engineering, NIT, Surathkal, India in 2009. His current research interests includes analog VLSI design, mixed circuit design and solid state electronics.

Rajesh R. is currently working as Senior Engineer, Strategic Electronics Group, C-DAC, Trivandrum. He received the Bachelor of Science degree from Department Electronics, Mahatma Gandhi University, India in 2003 and Master of Science degree from Department of Electronics, Bharathidasan University, India in 2005 and Master of Technology degree from Department of Embedded System Engineering, Vinayaka Mission University, India in 2007. His current research interest in embedded system, VLSI design and signal processing.

IJERT