INTERFACING C328 CMOS CAMERA WITH AN ATMega32L MICROCONTROLLER

Er. Shrijandra Nath Kayastha
Department of Electronics and Communication Engineering,
Khwopa Engineering College, Libali, Bhaktapur

Abstract
A small low power VGA camera module C328 can perform as a JPEG compressed still camera and can be attached to a microcontroller, PC, wireless or PDA host, etc. This paper introduces the C328 CMOS camera interface with a low power CMOS 8 bit ATmega32L microcontroller based on AVR enhanced RISC architecture. The ATmega32L executes powerful instruction in single clock cycle and achieves throughputs approaching 1MIPS per MHz that allows designing the system with optimum power consumption versus processing speed. The camera module C328 was serially interfaced with microcontroller, and the lowest resolution JPEG image (80x64) compression mode was made for the analysis of packet of image data in hex format. It was specifically designed for the cost effective embedded vision system that can be used in robot, security, monitoring etc.

Keywords: Camera, Embedded system, Microcontroller, Hex format

1. Introduction
As in the modern world, the vision system is main concern. People are concerned about the security of their homes and property through CCTV camera. The robotic systems are becoming smaller, lower power, and cheaper enabling their application in areas previously impossible, and this is also true of the vision system. Traditionally, these systems comprise a camera with complex hardware system to interface and require powerful computer and large memory for the image processing. Recent developments of low cost CMOS color camera modules and high speed microcontrollers make it possible to build a simpler and cost effective system. The well known camera C328 which consists of Omnivision image sensor OV7640 and OV528 JPEG image compressor chip. The compression chip includes a serial interface suitable for a direct connection to a microcontroller’s UART. By issuing the appropriate commands, a snap-shot as JPEG-compression byte streams of the lowest resolution 80x64 of image was taken within the limit of 2Kbyte of internal SRAM of ATMega32L.

For the simulation and debugging, the VMLAB was used since it provides a true virtual microcontroller design lab in which the hardware and software are co-simulation, making unnecessary in-circuit emulator.

VMLAB uses WinAVR as a compiler for compiling the code [6]. WinAVR is a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors hosted on the Windows platform. It includes the GNU GCC compiler for C and C++ [8]. A simple parallel port PonyProg programmer was used for programming the AVR that also provides the facility of setting the fuse bits of the microcontroller.

2. Camera Sensors Interfacing
The C328 uses Omni Vision OV7640/8 VGA Camera Chips with an 8-bit YCbCr interface and compressor chip. The OV528 Serial Bridge is a controller chip that can transfer image data from Camera Chips to external device [5]. The OV528
Shrijandra N. Kayastha
takes 8-bit YCbCr 422 progressive video data from
an OV7640CameraChip [2]. The camera interface
synchronizes with input video data and performs
down-sampling, clamping and windowing functions
with desired resolution, as well as color conversion
that are requested by the user through serial bus
host commands. The JPEG CODEC with variable
quality settings can achieve higher compression
ratio & better image quality for various image
resolutions which are shown in Fig 2(a-b). A serial
type program memory is built-in for C328, which
provides user-friendly commands to interface
external control units. This command was issued
by the host to change the size of data package which
was used to transmit JPEG image data from the
C328-7640 to the host [2]. This command was
issued before sending Snapshot Command or Get
Picture command to C328-7640. It was noted that
the size of the last package varies for different
image.

The C328 needs 3.3V to power up which was
designed using LM317 voltage regulator. Since, the
camera operating voltage is 3.3V, the microcontroller
ATMega32L was selected to operate in low voltage
of 3.3V, and this removed the use of voltage logic
level converter between microcontroller and camera.
The C328 camera module responds to serial
commands and each command of the C328 is equal
to 6 bytes. Exact value of external crystal was used
to synchronize properly between the camera and
microcontroller [1] and low byte of fuse bits
CKSEL3…0 are configured as (1111) through
PonyProg (setting CKSEL3…0 as unchecked) [7] in
order to use external crystal [4].

2.1 Package Size
The default size is 64 bytes and the maximum size
is 512 bytes [2].

<table>
<thead>
<tr>
<th>ID</th>
<th>Data size</th>
<th>Image Data</th>
<th>Verify code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2 bytes)</td>
<td>(2 bytes)</td>
<td>(Package size - 6) bytes</td>
<td>(2 bytes)</td>
</tr>
</tbody>
</table>

ID -> Package ID; starts from zero for an image
Data Size -> Size of image data in the package
Verify Code -> Error detection code, equals to the lower byte of sum of the whole package data except the verify code field. The higher byte of this code is always zero.

Number of packets = image size/ (data size-6)
Image size = HH LL of AA 0A 01 LL HH ZZ
Data size = HH LL of packet size (00 00, LL HH, Image Data, Verify Code)

The SYNC command AA 0D 00 00 00 00 (@14400bps) was sent until receiving ACK command AA 0E 0D XX 00 00 from C328-7640 (usually an ACK command is received after sending 40 times of SYNC command) [2]. Note: The value of XX can be 00 or 01 or 10 or 11. Similarly, the host should acknowledge the camera by sending the ACK command AA 0E 0D 00 00 00, so in this way the camera was synchronized with the host (microcontroller). Now, the microcontroller set up the configuration of camera, and the snap shot command was sent to get picture. Since the camera goes to sleep mode after receiving the command AA 09, the microcontroller should send SYNC command AA 0D to wake up C328 camera until ACK command is received from camera.

2.2 JPEG Snapshot Picture
By default package size is 64 bytes, so there was no need to set packet size.
The camera has to be initialized and setup in the configuration that the user defines. Here, the camera was set in the following configuration:

- Initial Cmd: Color Type = 07h = JPEG, JPEG Resolution = 01h = 80x64
- Get Picture Cmd: Picture Type = 01h = JPEG
- Snapshot Cmd: Snapshot Type = 00h = Compressed
- Package Size Cmd: Leave as it is for Default = 64 bytes
- Data Cmd: Data Type = 01h = Snapshot Picture

The software Hex Editor Free Serial Monitor was used to view the data from camera that was sent or received from the Serial Port [9]. The sample data of above picture from camera for 80x64 resolutions was as follows:

3. Packet information
After initializing the camera JPEG snap shot picture command AA 04 01 00 00 00 was sent and the camera acknowledged by the command AA 0E 04

Fig.2. Snap shot picture taken from C328 CMOS camera

Fig.3. Snap shot of HEX data format (Left) and ASCII format (Right) of Microcontroller.jpg (80x64) in Hex Editor Software, showing starting JPEG format FF D8.
04 00 00 along with the packet size information by AA 0A 01 C4 02 00.

Here the size of image data is store in the format HH LL of AA 0A 01 LL HH ZZ, i.e. image size is 02 C4 means 708 bytes of image data are sent from the camera to calculate the number of packets. The default packet size 64 bytes was used and the number of packets= image size/ (data size-6) = ceil[708/(64-6)]=12 [3]. Now, the camera was ready to send the number of packets of image data and to receive the first packet data the command AA 0E 00 00 00 00 was sent, where 00 means initial packet and the camera will send image data:

First packet:
00 00 3A 00 FF D8 FF E0 00 11 4A 46 49 46 00 01
02 03 04 05 06 07 08 09 0A FF DB 00 43 00 10 0C
0C 0E 0C 0A 10 0E 0E 0E 12 12 10 14 18 28 1A
18 16 18 32 24 26 1E 28 3A 34 3E 3C 3A 34 10 00

Here, 00 means first packet with the packet containing 3A or 58 bytes of image data and 3A is 1 byte i.e. 8 bits and AVR microcontroller can transfer or receive 1 byte (e.g. AA) at a time. Similarly, to receive the second packet of image data, the command AA 0E 00 00 01 00 was sent.

Second Packet:
01 00 3A 00 38 38 40 48 5C 4E 40 44 58 46 38 38
50 6E 52 58 60 62 68 68 68 3E 4E 72 7A 70 64 78
5C 66 68 64 FF DB 00 43 01 12 12 16 16 16 30
1A 1A 30 64 42 38 42 64 64 64 64 64 64 64 64 29 00

Third Packet:
02 00 3A 00 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 01 00 00 F6 00

Fourth Packet:
03 00 3A 00 00 00 00 00 00 00 00 00 00 00 01 02 03 04 05
06 07 08 09 0A 0B FF C4 00 B5 10 00 02 01 03 03
02 04 03 05 05 04 04 00 00 01 7D 01 02 03 00 04
11 05 12 21 31 41 06 13 51 61 07 22 71 14 E7 00

Similarly, 5th to 12th packet was received.

Last Packet:
0C 00 0C 00 00 28 A2 8A 00 28 A2 8A 00 FF D9 00 98 00

Here, the packet ID 0C 00 0C 00 informs the 13th packet with 12 bytes of image data and 98 00 is the end of packets. So, It was programmed to remove packet ID (e.g. 0C 00), data size (e.g. 0C 00) and verify code (e.g. 98 00) and store image data in an array of size (58*12 + 12= 708, i.e. 02 C4) for all the packets which gives the image of the snapshot. Actually there are 13 packets, i.e. 12 packets of 58 bytes and 1 packet of 12 bytes. The image data stored in the memory of microcontroller was in the following format which can be transfer to PC to view the complete picture. The whole image was stored in the memory of microcontroller but of small resolution, so to store the picture of high resolution the memory card can be used. Image Data in the memory of Microcontroller:

FF D8 FF E0 00 11 4A 46 49 46 00 01 02 03 04 05
06 07 08 09 0A FF DB 00 43 00 10 0C 0C 0E 0C 0A 10 0E 0E 0E 12 12 10 14 18 28 1A 18 16 18 32 24 26 1E 28 3A 34 3E 3C 3A 34 10 00

Where, FF D8 and FF D9 are the start and end of JPEG format respectively [10]. This, can be verify when we view any JPEG picture by HEX editor software that shows the HEX data format of JPEG along with their ASCII form [9].

4. Conclusion

The C328 is a highly integrated camera module featured on-board JPEG compression engine and RS232 serial interface. The goal of this project is to implement a simple embedded vision system consuming low power to replace fairly complicated, expensive system that is out of reach for many developers who have moderate camera interfacing requirements for storing in MMC card, security system, uploading image using GPRS modem, robotics etc. The feasibility was demonstrated by taking the JPEG still image of low resolution and storing the image data in memory of microcontroller which can be easily used for the further
requirements. However, only the lowest resolution of image was stored in microcontroller due to its small size (2K Byte) of internal SRAM and for the higher resolution one can extend memory or interface MMC card to store large number of pictures at high resolution.

References


