# Interfacing Serial JPEG Camera Module with Computer using Zigbee pair

*A Guide to interface Serial JPEG Camera with Matlab*

*By*

*Kaushik Basak Chowdhury*

*Project Staff, IIT Bombay*

*Email id- soumyakbc@gmail.com*

## Contents

## I. Introduction

This document provides a step-by-step approach towards understanding of the camera module hardware, testing the module interface using X-CTU (software from Digi) and finally implementation of Image Acquisition using Matlab.

## II. Camera Module Description

The Camera module hardware is briefly described in the Figure 2.1

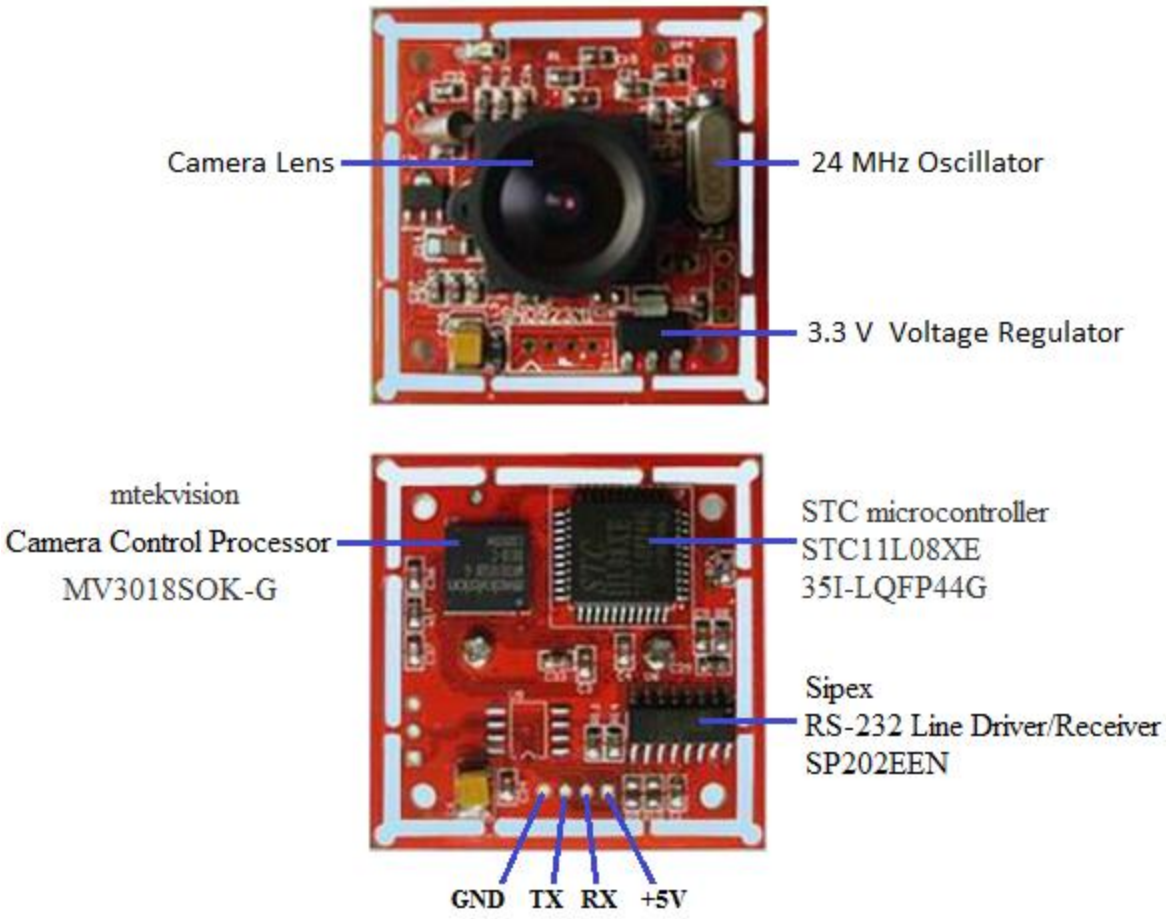



Figure 2.1 Description of Camera Hardware

The functional block diagram of the Camera module is shown in following Figure 2.2.

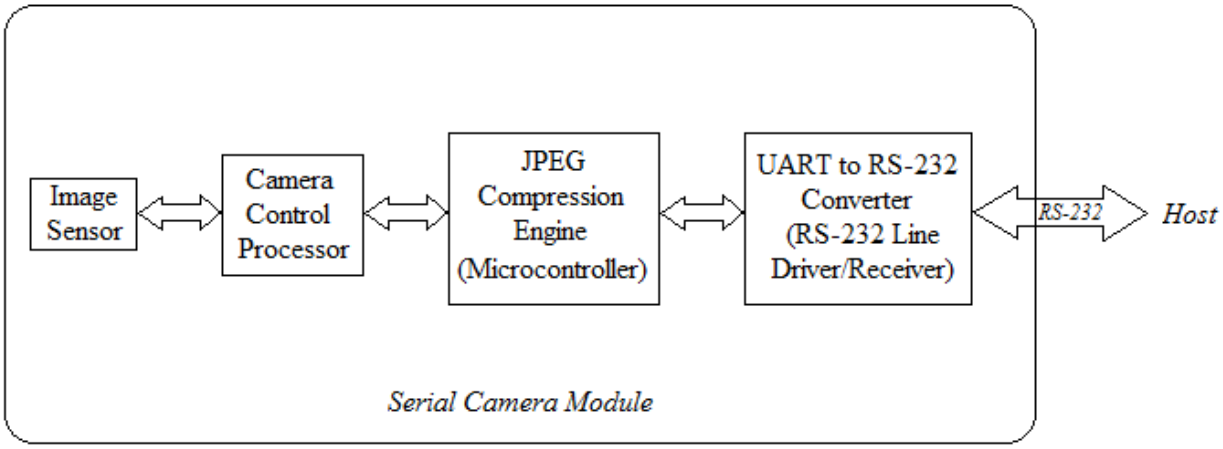



Figure 2.2: Functional Block Diagram of the Serial Camera Module

The camera module consists of a CMOS image sensor which is controlled by mtekvision Camera Control Processor (CCP). The captured image is compressed in JPEG format by the onboard STC microcontroller which uses OV528 protocol, developed by OmniVision. The compressed JPEG image is then divided into packets and transmitted out through UART. The size of the packets can vary from 64 to 512 bytes. In this particular camera module the UART output from the STC microcontroller is converted to RS-232 logic levels by the Sipex RS-232 Line Driver/Receiver. Finally the RS-232 interface is available from the camera module board to connect to a host. This camera module supports Baud Rates from 7200 bps to 115200 bps.

*Note 1: In order to communicate with this serial camera module the Baud Rate of the host should match with that of the camera module and the host should have RS-232 interface. The camera module cannot detect and configure the Baud Rate automatically. One has to manually set the Baud Rate by issuing specific commands.*

*Note 2: The default command set for OmniVision OV528 Protocol is not directly used in this camera. The manufacturer has used a generic command set by changing the firmware and the host driver. The command set used in this camera module is provided in the Appendix of this document.*

## III. Interfacing Camera Module with Computer

### i. Directly using wired connection

The camera module can easily be connected to a computer having RS-232 Serial Port. Since computers today rarely have a Serial Port and on the other hand have multiple USB ports, the need for a *RS-232 to USB converter* arises. The setup is shown in the Figure 3.1

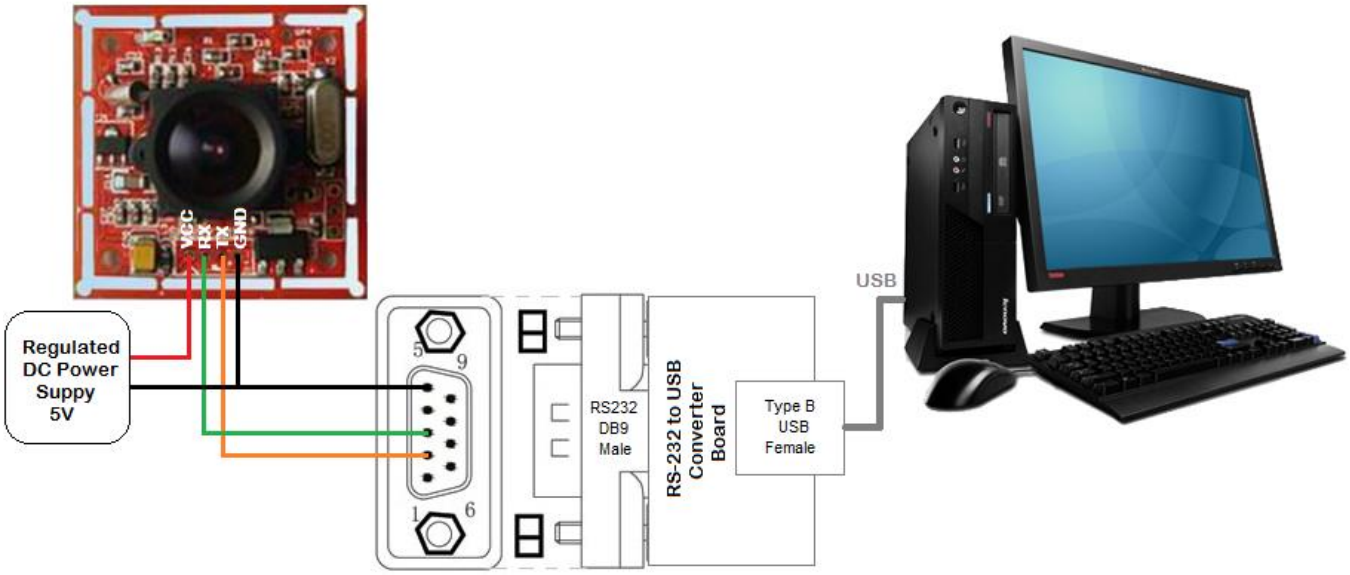



Figure 3.1: Connecting serial camera module with computer using Serial-to-USB converter

*Note: The RS-232 to USB converter board (or will appear as a "COM" Port in the Device Manager in Windows OS. The same COM Port is to be referred to while communicating with the camera through Matlab or X-CTU software.*

### ii. Wireless connection using Zigbee Pair

The camera module can be wirelessly connected to the computer using a Zigbee module pair in between as shown in the Figure 3.2.

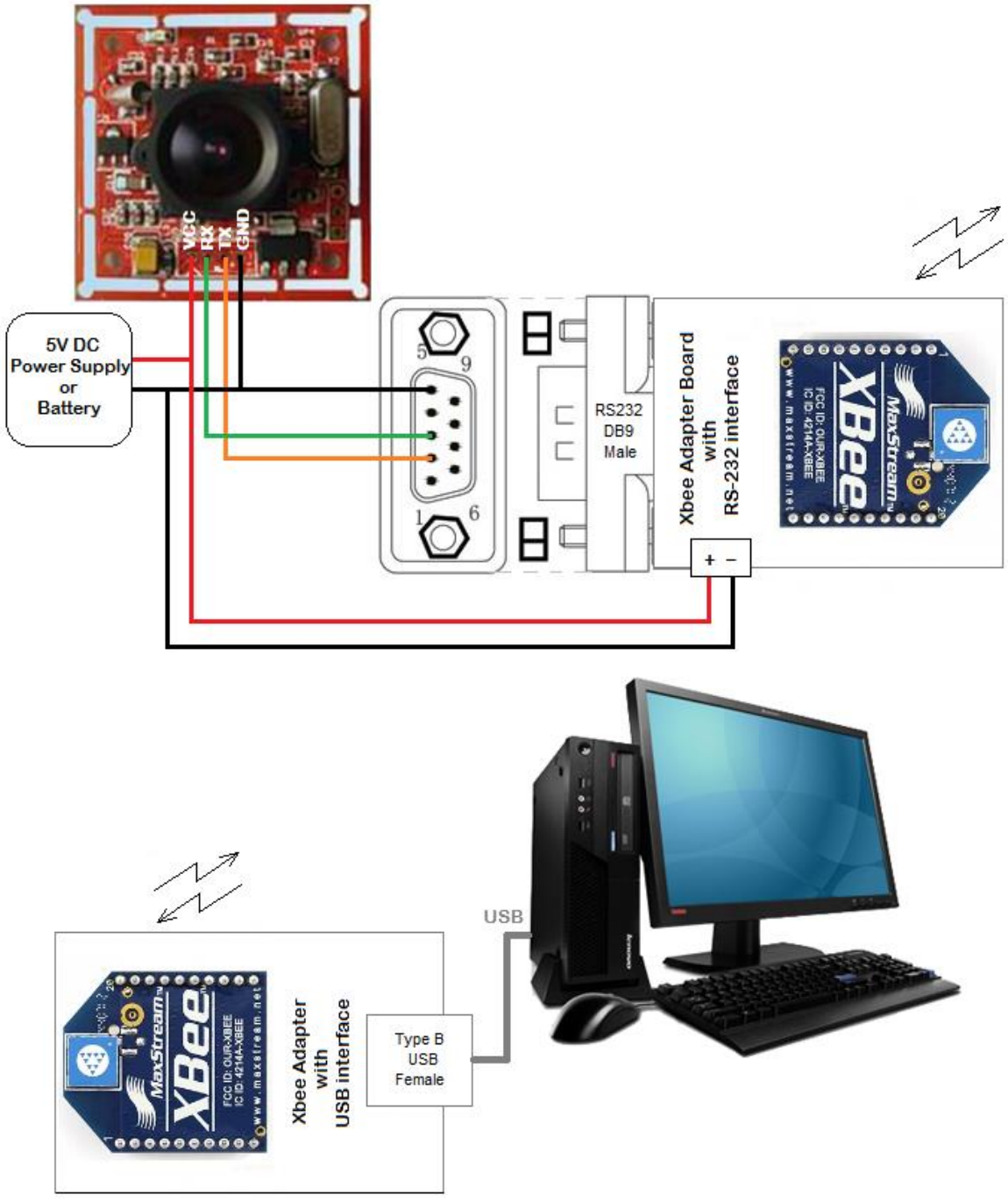
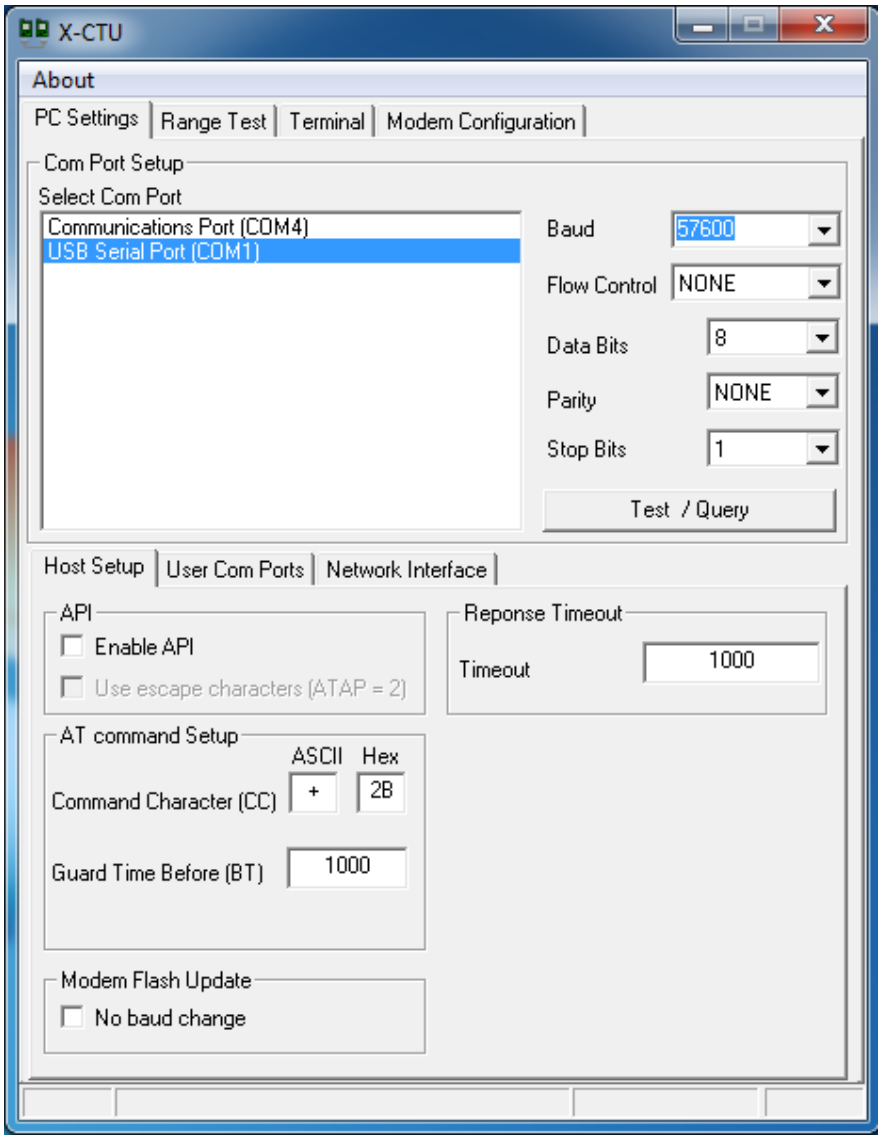



Figure 3.2: Connecting serial camera module with computer using Zigbee Module Pair

*Note: One should make sure that the same Baud Rate is assigned to - the camera module, the Xbee pair, the X-CTU(or Matlab) software. Since, the default Baud Rate for Camera module is 9600 bps; it is convenient to set the Baud Rates of all other interfacing devices to 9600 bps as well.*

## IV. Testing the Camera Module using X-CTU software

The camera module can be tested using the X-CTU software from Digi. The entire method is described using diagrams. In our example the Serial Port is COM1 and Baud Rate is 57600 bps. After opening the X-CTU terminal the following window will show up on the screen:



From the X-CTU window make the following settings:

PC Settings > Com Port Setup > Select Com Port > Select **USB Serial Port (COM1)**

PC Settings > Com Port Setup > Baud > Select **57600**

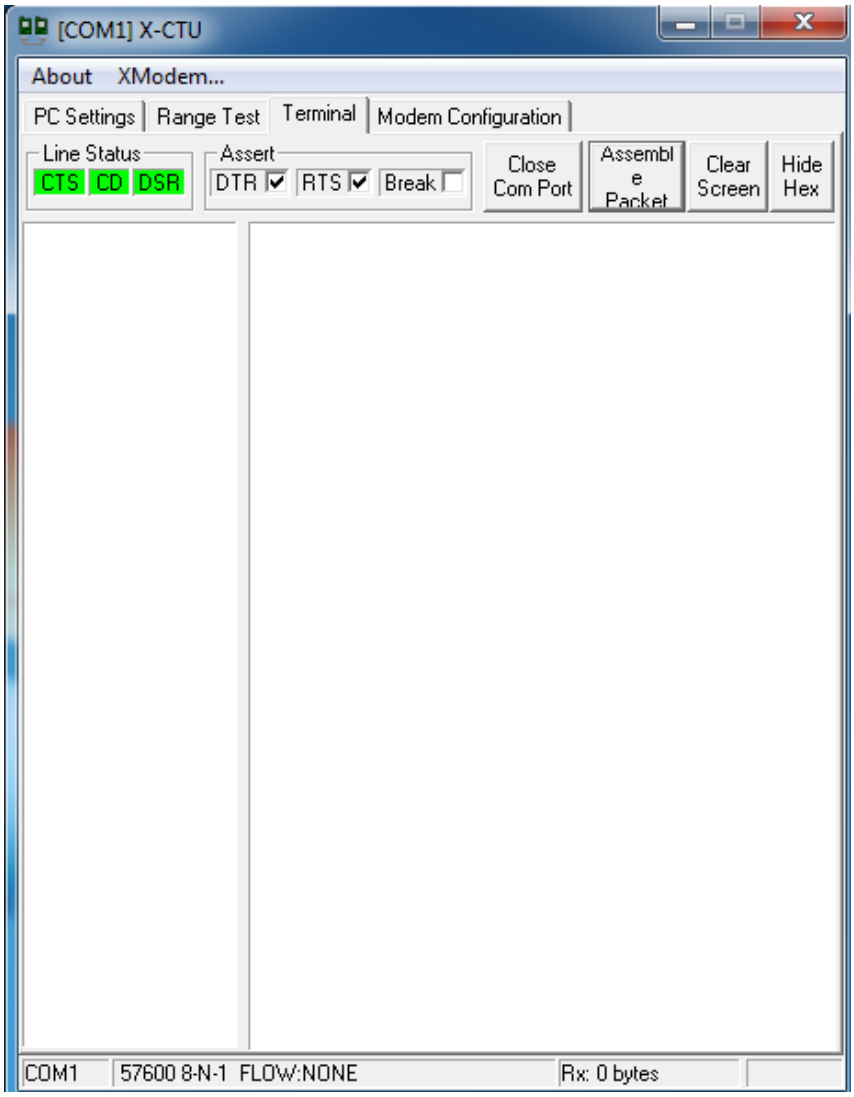
PC Settings > Com Port Setup > Flow Control > Select **NONE**

PC Settings > Com Port Setup > Data Bits > Select **8**

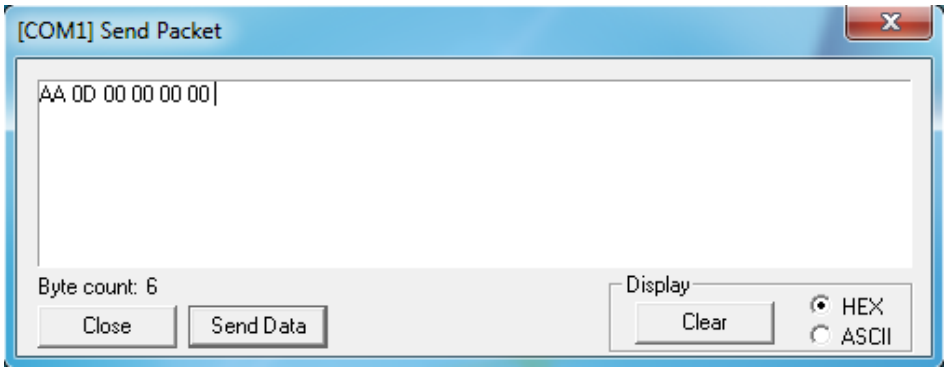
PC Settings > Com Port Setup > Parity > Select **NONE**

PC Settings > Com Port Setup > Stop Bits > Select **1**

Next, go to the tab ‘Terminal’ and the Click on ‘Show Hex’ and the following will appear:



Now click on the ‘Assemble Packet’ and the following window will show up.



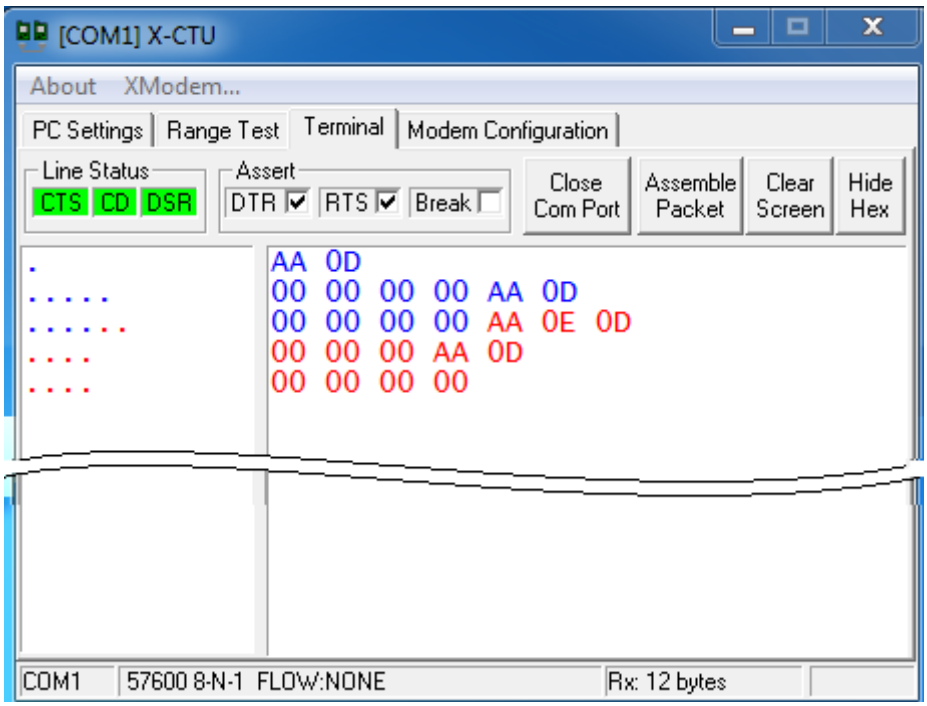

Here one can type the bytes which need to be transmitted out from the terminal. In our example we are sending the SYNC command which consists of the following six bytes –

**AA 0D 00 00 00 00**

As soon as you hit the ‘Send Data’ Button, the X-CTU terminal will send out those 6 bytes and the sent bytes appears in blue color in Terminal Hex screen. On receipt of those 6 bytes or SYNC command the camera sends back an ACK of the following 6 bytes-

**AA 0E 0D 00 00 00**

The bytes received from the camera module are displayed in red in the X-CTU terminal Hex screen as shown in the following diagram-



## V. Image Acquisition using Matlab Software

In general, Image Acquisition Toolbox of Matlab automatically recognizes an imaging device (for example Web-cam, digital cameras etc.) connected to the computer and hence the image acquisition toolbox functions work directly. This is because these cameras have got a device firmware embedded in the controller of the device which has a capability to connect as a USB device and act as an imaging device. The serial camera module in our case does not have any such program which automatically communicates with the host device (computer in this case). Thus, this cannot be termed as an 'imaging device' from the perspective of interfacing with Matlab. The serial camera module has to be supplied with appropriate set of commands for communication. This has to be done by creating a 'Serial Object' in Matlab through which we can communicate with the serial camera module.

As shown in Section III, the serial camera module can be connected to the Computer in two different ways, both of which offers a provision to access the camera module through a 'Virtual Com Port' designated by a specific COM port number. This COM port number has to be used to create the 'Serial Object' in Matlab program.

Now, the entire Matlab code written for image acquisition will be elaborated in detail. The command set is provided in the Appendix.

1. *Creating a Serial Object- 'cam' is the name of the serial object created in Matlab which is connected to the COM port 1.*

```
cam=serial('COM1');
```

2. *Setting the Baud Rate- the Baud Rate of the serial object 'cam' is set at 57600 bps. It is assumed that the camera baud rate is also 57600 bps.*

```
cam.BaudRate=57600;
```

3. *Setting Byte Order of 'cam' object to Big Endian*

```
set(cam,'ByteOrder','bigEndian');
```

4. *Setting Input Buffer Size of 'cam' object to 512 bytes, since data packets received from the serial camera is of size 512 bytes.*

```
cam.InputBufferSize=512;
```

5. *Setting Timeout of 'cam' object to 0.25 seconds, an optimum value for smooth operation*

```
cam.Timeout=0.25;
```

6. *Opening 'cam' object*

```
fopen(cam);
```

7. *Synchronization with the serial camera by sending SYNC commands and receiving ACK commands*

```
for i=1:50
    fwrite(cam,hex2dec({'AA','0D','00','00','00','00'}));
    if cam.BytesAvailable>=1
        %ack=fread(cam);
        break;
    end
    pause(0.2);
end
```

8. *Creating and opening JPEG file for storing the acquired image*

```
jpg_id=fopen('jpeg_snapshot.jpg','w+');
```

9. *Initialization of the camera module*

```
fwrite(cam,hex2dec({'AA','01','00','07','00','07'}));
```

10. *Setting package size to512 bytes*

```
fwrite(cam,hex2dec({'AA','06','08','00','02','00'}));
```

11. *Setting the type of picture which is to be acquired from the camera module to Compressed Snapshot*

```
fwrite(cam,hex2dec({'AA','05','00','00','00','00'}));
pause(0.05);
ack=fread(cam,cam.BytesAvailable)
```

12. *Sending trigger to get picture*

```
fwrite(cam,hex2dec({'AA','04','01','00','00','00'}));
```

*13. Reading the size of image data from the bytes received from the cam*

```
data_size=fread(cam);
data_size_dec=dec2hex(data_size);
```

*14. Calculating the number of 512 byte size packets*

```
h_byte=data_size_dec(11,:);
l_byte=data_size_dec(10,:);
data_bytes_hex=strcat(h_byte,l_byte);
data_bytes_dec=hex2dec(data_bytes_hexi=data_bytes_dec;
p=floor(i/506);
```

*15. Calculating the total number of packets*

```
t=p+1;
```

*16. Calculating the number of compressed image data bytes in the last packet*

```
last_data_bytes=i-(p*506);
```

*17. Creating a zero matrix of rows equal to the number of packets received and 512 columns. This matrix is used to store the raw data packets received from the camera module*

```
Img_data_packet=zeros(p,512);
```

*18. Reading the packets from the camera module*

```
for j=1:t

    id=dec2hex(j-1);
    l=length(id);
    h_id='00'; l_id='00';
    switch l

        case 1
            h_id='00';
            l_id(1)='0';
            l_id(2)=id;

        case 2
            h_id='00';
            l_id=id;
```

```
        case 3
            h_id(1)='0';
            h_id(2)=id(1);
            l_id(1)=id(2);
            l_id(2)=id(3);

        case 4
            h_id(1)=id(1);
            h_id(2)=id(2);
            l_id(1)=id(3);
            l_id(2)=id(4);
    end

    %Send ACK with dataID and get data packets
    ack_dataID=hex2dec({'AA','0E','00',h_id,l_id,'00'});
    fwrite(cam,ack_dataID);
    Temp_data=fread(cam);
    Img_data_packet(j,1:length(Temp_data))=Temp_data;
end
```

19. *The packet matrix which has already been created contains of the following-*

    *1st and 2nd column- Packet ID no.*
    *3rd and 4th column- Number of compressed image data in that particular packet (row)*
    *5th to third from last column- Compressed image data bytes*
    *Last two columns (511st and 512nd in for 512-byte packets) - Verify bits*

    *Now the Compressed image data bytes are to be extracted from the packet matrix.*

```
hi_byte=dec2hex(Img_data_packet(t,4));
lo_byte=dec2hex(Img_data_packet(t,3));
last_pck_data_hex=strcat(hi_byte,lo_byte);
last_pck_data_bytes=hex2dec(last_pck_data_hex);
d=4+last_pck_data_bytes;
Crmpsd_Image_data=Img_data_packet(1:t,5:510);
Crmpsd_Image_data(t,1:(d-4))=Img_data_packet(t,5:d);
```

20. *Writing the JPEG compressed image data matrix into the previously created JPEG file.*

```
for i=1:t
    data=Crmpsd_Image_data(i,:);
    fwrite(jpg_id,data);
end
```

21. *Closing the jpeg file*

```
fclose(jpg_id);
```

*22. Displaying captured image using Matlab*

```
f=figure;
set(f,'renderer','openGL');
imshow('jpeg_snapshot.jpg');
```

*23. Closing, deleting and clearing the Serial object 'cam'*

```
fclose(cam);
delete(cam);
clear cam;
```

## *Appendix – Command Set*

### Serial Interface

1. Single Byte Timing Diagram
   A single byte RS-232 transmission consists of the start bit, 8-bit contents and the stop bit. A start bit is always 0, while a stop bit is always 1. LSB is sent out first and is right after the start bit.

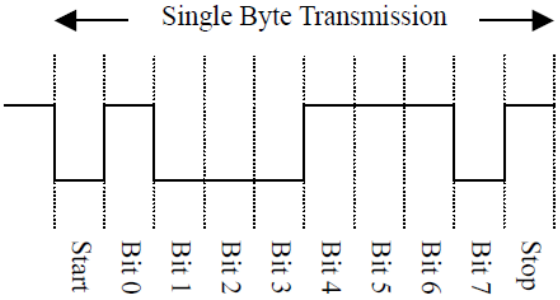



Figure 3 – RS-232 single byte timing diagram

2. Command Timing Diagram
   A single command consists of 6 continuous single byte RS-232 transmissions. The following is an example of SYNC (AA0D00000000h) command.

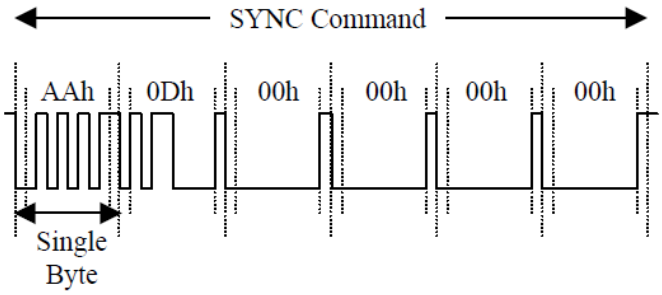



Figure 4 – RS-232 SYNC command timing diagram

## Command Set

The C328-7640 module supports total 11 commands for interfacing to host as following:

| Command | ID Number | Parameter1 | Parameter2 | Parameter3 | Parameter4 |
|---|---|---|---|---|---|
| Initial | AA01h | 00h | Color Type | RAW Resolution (Still image only) | JPEG Resolution |
| Get Picture | AA04h | Picture Type | 00h | 00h | 00h |
| Snapshot | AA05h | Snapshot Type | Skip Frame Low Byte | Skip Frame High Byte | 00h |
| Set Package Size | AA06h | 08h | Package Size Low Byte | Package Size High Byte | 00h |
| Set Baudrate | AA07h | 1st Divider | 2nd Divider | 00h | 00h |
| Reset | AA08h | Reset Type | 00h | 00h | xxh* |
| Power Off | AA09h | 00h | 00h | 00h | 00h |
| Data | AA0Ah | Data Type | Length Byte 0 | Length Byte 1 | Length Byte 2 |
| SYNC | AA0Dh | 00h | 00h | 00h | 00h |
| ACK | AA0Eh | Command ID | ACK counter | 00h / Package ID Byte 0 | 00h / Package ID Byte 1 |
| NAK | AA0Fh | 00h | NAK counter | Error Number | 00h |
| Light Frequency | AA13h | Frequency Type | 00h | 00h | 00h |

* If the parameter is 0xFF, the command is a special Reset command and the firmware responds to it immediately.

### 1. Initial (AA01h)

The host issues this command to configure the preview image size and color type. After receiving this command, the module will send out an ACK command to the host if the configuration success. Otherwise, an NACK command will be sent out.

1.1 Color Type

C328-7640 can support 7 different color types as follow:

| | |
|---|---|
| 2-bit Gray Scale | 01h |
| 4-bit Gray Scale | 02h |
| 8-bit Gray Scale | 03h |
| 12-bit Color | 05h |
| 16-bit Color | 06h |
| JPEG | 07h |

1.2 Preview Resolution

| | |
|---|---|
| 80x60 | 01h |
| 160x120 | 03h |

1.3 JPEG Resolution

Since the Embedded JPEG Code can support only multiple of 16, the JPEG preview mode can support following image sizes. It is different from normal preview mode.

| | |
|---|---|
| 80x64 | 01h |
| 160x128 | 03h |
| 320x240 | 05h |
| 640x480 | 07h |

## 2. Get Picture (AA04h)

The host gets a picture from C328-7640 by sending this command.

2.1 Picture Type

| | |
|---|---|
| Snapshot Picture | 01h |
| Preview Picture | 02h |
| JPEG Preview Picture | 05h |

## 3. Snapshot (AA05h)

C328-7640 keeps a single frame of JPEG still picture data in the buffer after receiving this command.

3.1 Snapshot Type

| | |
|---|---|
| Compressed Picture | 00h |
| Uncompressed Picture | 01h |

3.2 Skip Frame Counter

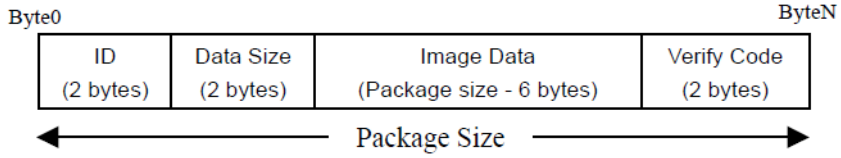
The number of dropped frames can be defined before compression occurs. "0" keeps the current frame, "1" captures the next frame, and so forth.

## 4. Set Package Size (AA06h)

The host issues this command to change the size of data package which is used to transmit JPEG image data from the C328-7640 to the host. This command should be issued before sending Snapshot command or Get Picture command to C328-7640. It is noted that the size of the last package varies for different image.

4.1 Package Size

The default size is 64 bytes and the maximum size is 512 bytes.




| | |
|---|---|
| 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. i.e. verify code = lowbyte(sum(byte[0] to byte[N-2])) |

Note: As the transmission of uncompressed image is not in package mode, it is not necessary to set the package size for uncompressed image.

## 5. Set Baudrate (AA07h)

Set the C328-7640 baud rate by issuing this command. As the module can auto-detect the baud rate of the incoming command, host can make connection with one of the following baud rate in the table. The module will keep using the detected baud rate until physically power off

5.1 Baudrate Divider

Baudrate = 14.7456MHz / 2 x (2nd Divider + 1) / 2 x (1st Divider + 1)

| Baudrate | 1st Divider | 2nd Divider | Baudrate | 1st Divider | 2nd Divider |
|---|---|---|---|---|---|
| 7200 bps | ffh | 01h | 28800 bps | 3fh | 01h |
| 9600 bps | bfh | 01h | 38400 bps | 2fh | 01h |
| 14400 bps | 7fh | 01h | 57600 bps | 1fh | 01h |
| 19200 bps | 5fh | 01h | 115200 bps | 0fh | 01h |

## 6. Reset (AA08h)

The host reset C328-7640 by issuing this command.

6.1 Reset Type

"00h" resets the whole system. C328-7640 will reboot and reset all registers and state machines. "01h" resets state machines only.

## 7. Power Off (AA09h)

C328-7640 will go into sleep mode after receiving this command. SYNC command (AA0Dh) must be sent to wake up C328-7640 for certain period until receiving ACK command from C328-7640.

## 8. Data (AA0Ah)

C328-7640 issues this command for telling the host the type and the size of the image data which is ready for transmitting out to the host.

8.1 Data Type

| | |
|---|---|
| Snapshot Picture | 01h |
| Preview Picture | 02h |
| JPEG Preview Picture | 05h |

8.2 Length

These three bytes represent the length of data of the Snapshot Picture, Preview Picture or JPEG Preview Picture.

## 9. SYNC (AA0Dh)

Either the host or the C328-7640 can issue this command to make connection. An ACK command must be sent out after receiving this command.

## 10. ACK (AA0Eh)

This command indicates the success of last operation. After receiving any valid command, ACK command must be sent out except when getting preview data. The host can issue this command to request image data package with desired package ID after receiving Data command from C328-7640. The host should send this command with package ID F0F0h after receiving a package to end the package transfer. Note that the field "command ID" should be 00h when request image data package.

10.1 Command ID

The command with that ID is acknowledged by this command.

10.2 ACK Counter

No use.

10.3 Package ID

For acknowledging Data command, these two bytes represent the requested package ID. While for acknowledging other commands, these two bytes are set to 00h.

## 11. NAK (AA0Fh)

This command indicates corrupted transmission or unsupported features.

11.1 NAK Counter

No use.

11.2 Error Number

| | | | |
|---|---|---|---|
| Picture Type Error | 01h | Parameter Error | 0bh |
| Picture Up Scale | 02h | Send Register Timeout | 0ch |
| Picture Scale Error | 03h | Command ID Error | 0dh |
| Unexpected Reply | 04h | Picture Not Ready | 0fh |
| Send Picture Timeout | 05h | Transfer Package Number Error | 10h |
| Unexpected Command | 06h | Set Transfer Package Size Wrong | 11h |
| SRAM JPEG Type Error | 07h | Command Header Error | F0h |
| SRAM JPEG Size Error | 08h | Command Length Error | F1h |
| Picture Format Error | 09h | Send Picture Error | F5h |
| Picture Size Error | 0ah | Send Command Error | ffh |

## 12. Light Frequency (AA13h)

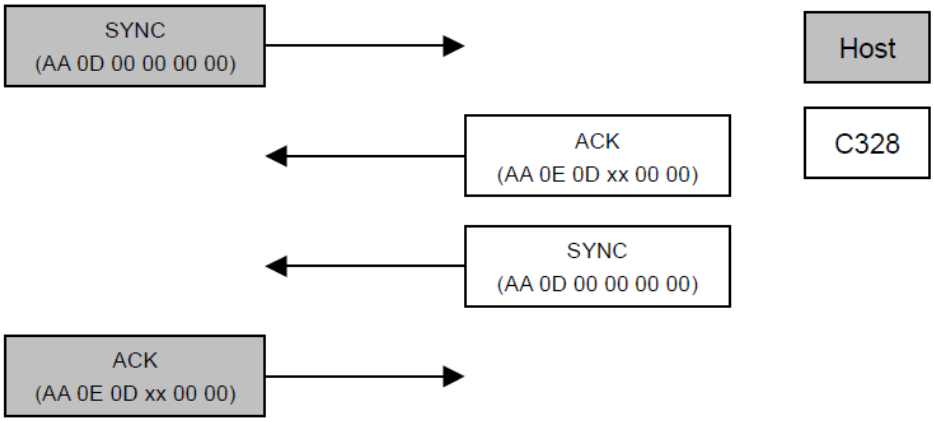
The host issues this command to change the light frequency of the C328-7640.

12.1 Light Frequency Type

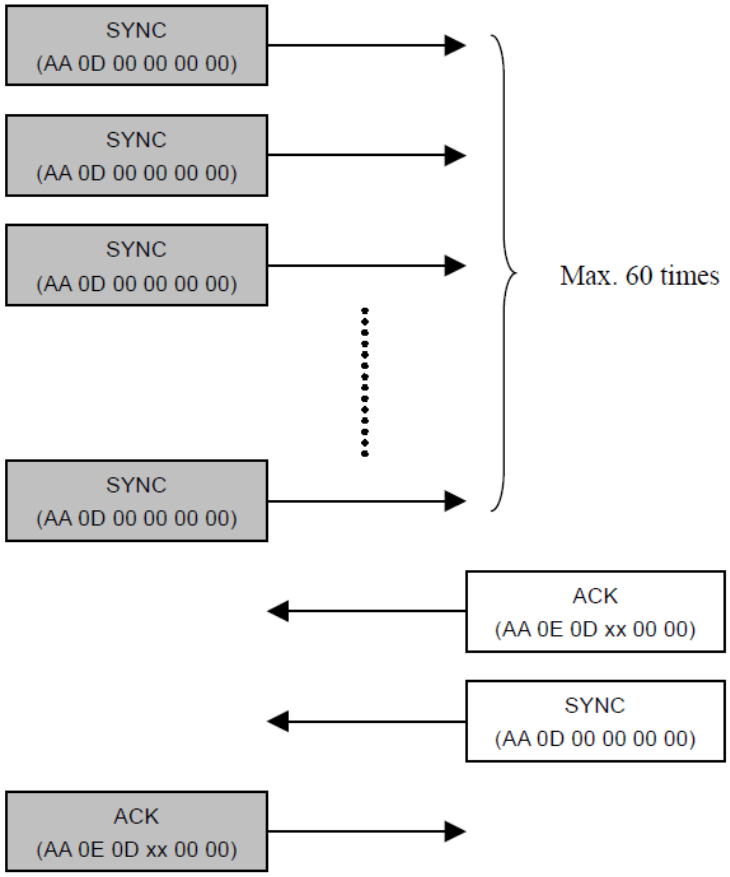
| | |
|---|---|
| 50Hz | 00h |
| 60Hz | 01h |

## Command Protocol

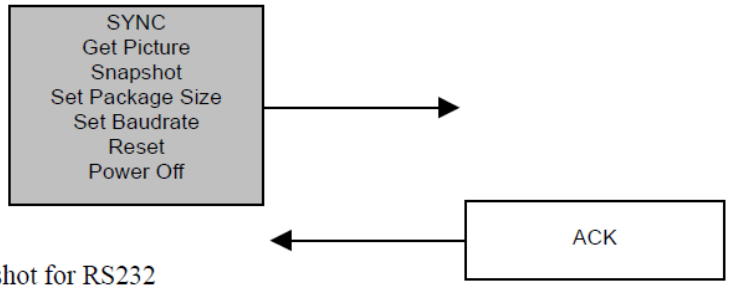
1. SYNC Command




2. Make Connection with C328-7640
Send the SYNC command (at 14400bps) until receiving ACK command from C328-7640 (usually an ACK command is receive after sending 25 times of SYNC command). This must be done after power up.



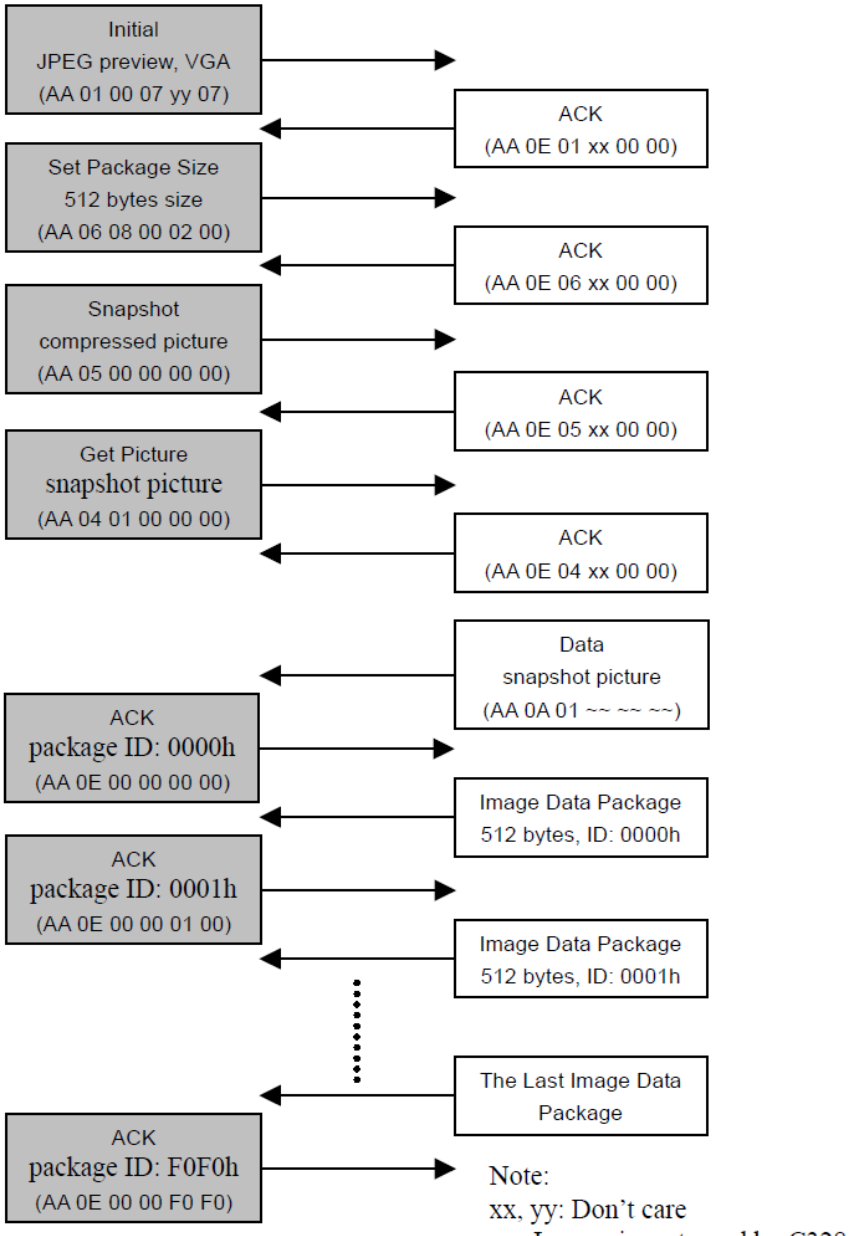
3. Initial, Get Picture, Snapshot, Set Package Size, Set Baudrate, Reset and Power Off Command




4. Getting a Snapshot for RS232

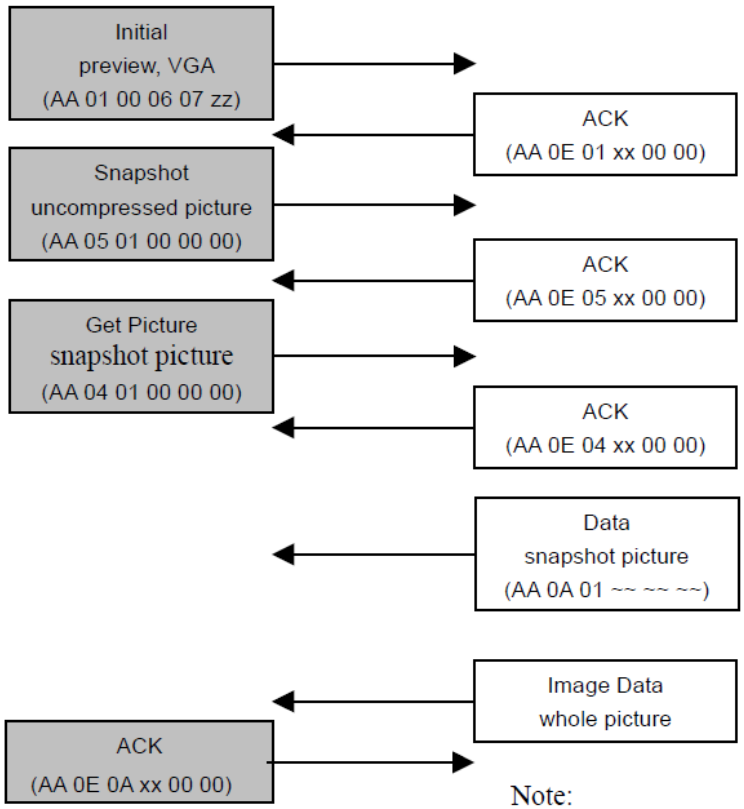
Make sure connection is made before the following communication.

4.1 JPEG Snapshot Picture (eg. 640x480 resolution)




Note:
xx, yy: Don't care
~~: Image size returned by C328

## 4.2 Snapshot Picture (uncompressed snapshot picture)

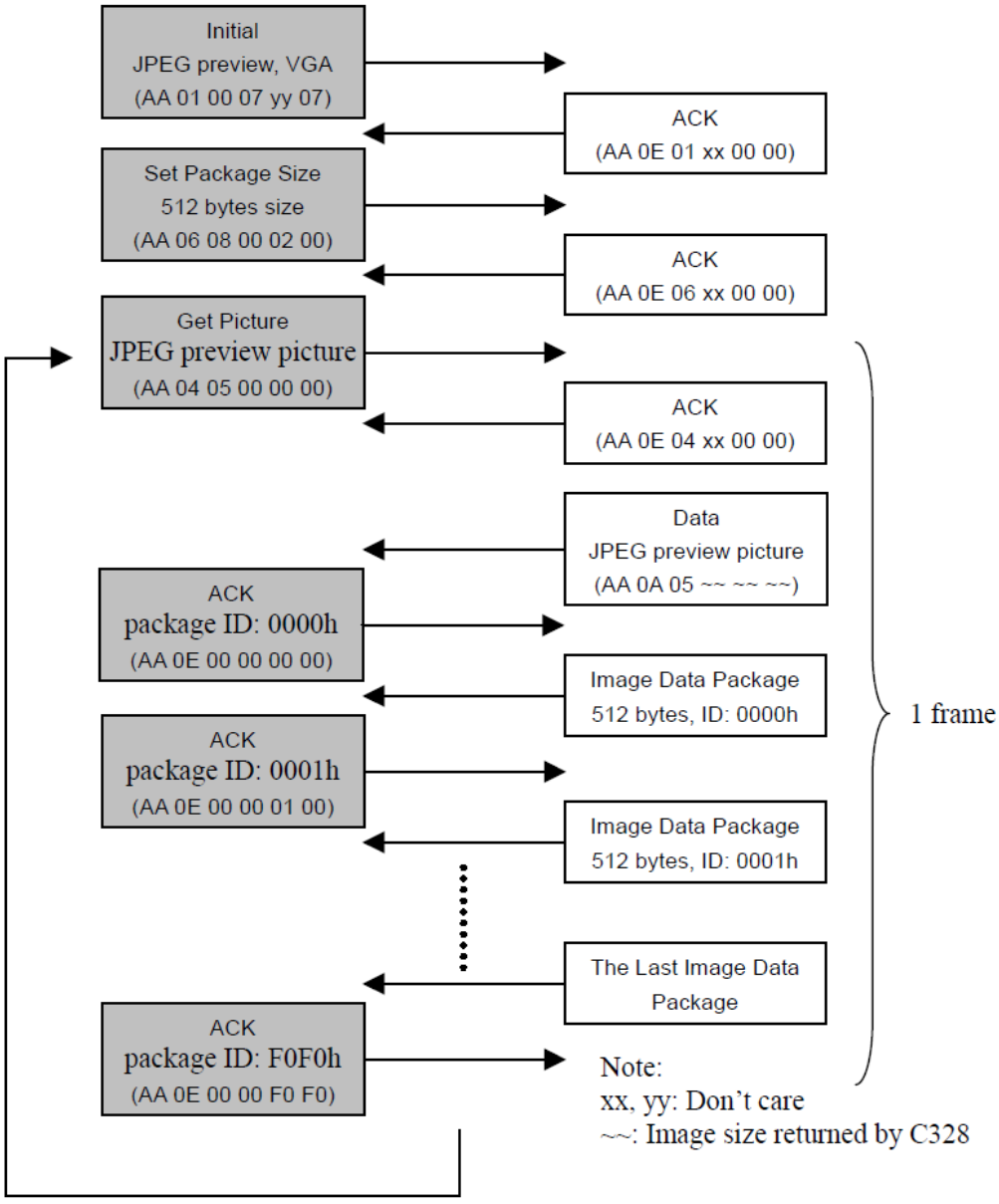



Note:
xx, zz : Don't care
~~: Image size returned by C328

5. Getting JPEG preview pictures (video) for RS232

Make sure connection is made before the following communication.

5.1 JPEG Preview Picture



## 5.2 Preview Picture (uncompressed preview picture)