

# COMMUNICATION PROTOCOL RS232 IMPLEMENTATION ON FPGA

**MR.SHITAL N.GAVADE,**

*ME Scholar,*

*VLSI & Embedded, DKTE's Textile and Engineering Institute,  
Ichalkaranji, Maharashtra, India*

**PROF.R.N.PATIL**

*Associate Professor,*

*VLSI & Embedded, DKTE's Textile and Engineering Institute,  
Ichalkaranji, Maharashtra, India.*

## ABSTRACT

Communication Protocol RS232 Implementation on Field Programmable Gate Array (FPGA) has been presented in this paper. The Image pixel values are converted into binary and send to the FPGA from PC through Serial Communication Protocol .GUI is designed in MATLAB and is used to interface Personal computer (PC) and FPGA. The image pixels are read through FPGA in binary format.

**INDEX TERMS:** FPGA, MATLAB, Image processing, RS232, VHDL, Verilog. Etc

## INTRODUCTION

FPGA generally consist of a system with logic blocks such as look up tables, gates, flip-flops and some memory blocks all placed in the vast array of interconnects. The FPGA can be reconfigured to a particular logic circuit using hardware description language like VHDL or Verilog. The FPGA architecture allows large variety of logic designs for real time application.

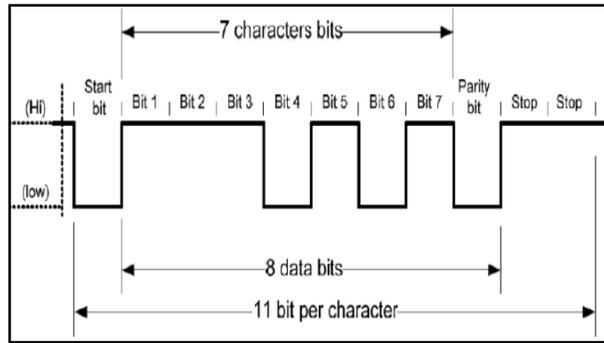
MATLAB GUI is used to communicate with the FPGA board. The Image is converted into binary format and then using RS232 interface the pixel data has been sent to the board. This paper gives a better idea to implement the RS232 Protocol on FPGA board. As the data is in binary format the hardware required is less and processing is fast. Because of binary data one don't need to implement square root algorithm in Verilog or VHDL language.

## DESIGN REQUIREMENT AND IMPLEMENTATION

### 2.1. RS232 PROTOCOL

The RS-232 serial communication protocol is a standard protocol used in asynchronous serial communication [3]. In asynchronous serial communication the data is transmitted without any clock signal to the receiver. Instead, special bits like start bit stop bit and parity bits are sent along with the data bits to synchronize transmitter and receiver. When data has to send

using asynchronous transmission a start bit is added at the beginning of the data and then data bits along with parity bit and stop bit is added. Here parity bit is optional. In ideal condition both Tx and Rx lines are held high. The length of data bits that can be sent are 5, 6, 7 or 8 bits. The start bit is '0' and the end bit might be 1, 1.5 or 2 bits in length with '1' value.



**Figure 1: RS232 asynchronous communication data wave form**

**BAUD RATE CALCULATION**

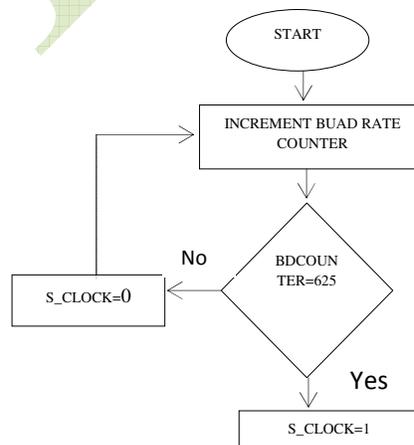
Baud rate is a measurement of transmission speed in asynchronous communication it represents the number of bits that are actually being sent over the serial link

Spartan 3E starter Kit Operating Frequency= 12 MHz

$$\text{Baud Rate} = 9.6 \text{ KHz} = 9000 \text{ Hz}$$

$$\text{Count} = 12 \text{ MHz} / 9000 \approx 625$$

For receiver the sampling is required so the Count is get modified for receiver  
 The flow chart to implement and generate the baud rate is given below. Here counter are initiated to count the values shown in count. After reaching each value the counter will be reset and start counting again. For each time counters reaching the terminal value, the baud clock and sampled baud clock are set '1'

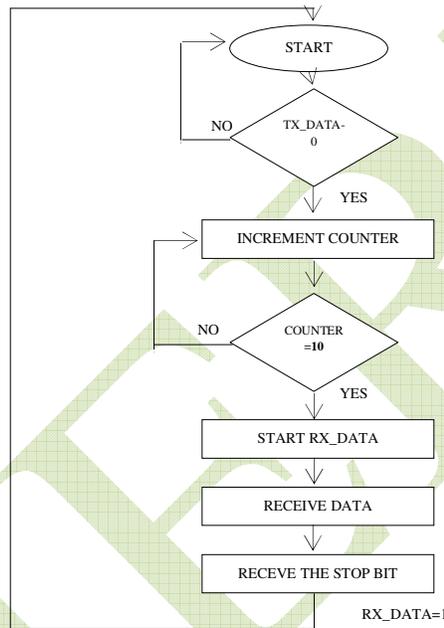


**Figure 2: Flow chart of Baud rate calculation**

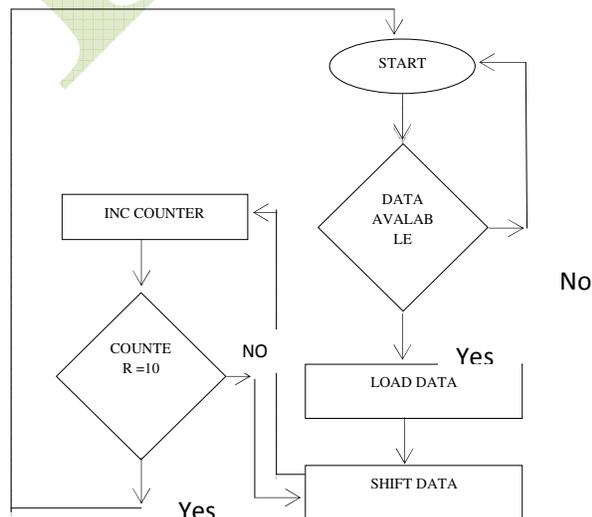
### 2.3 IMPLEMENT RS232 PROTOTYPE USING VHDL

The receiver block of the RS232 serial communication will run on the sampled baud clock while transmitter block will run on the normal baud clock signal. The transmitter can also run on the sampled baud clock but as switching will be more in sampled baud clock to reduce the dynamic power consumption we will run transmitter block on normal baud clock signal.

The sampling can be done with 8, 16 or 32 samples. Here we are doing it with 16 samples. The sampling is required because of the uncertainty of the start bit arrival, as it is an asynchronous communication. The flow chart for receiver and transmitter is shown below. In the receiver the counter need to be synchronized. As we are taking 16 samples the data will be captured at 8th sample, so as soon as counter2 reaches 8th sample the data will be received.



**Figure 3: Flow chart of RS232 Receiver**



**Figure 4: Flow chart of RS232 Transmitter**

## MATLAB GUI IMPLIMENTATION

MATLAB has a very good Graphics User Interface development environment tool to develop reliable and fast user interface. The binary image conversion is done easily with MATLAB in build functions. Also serial communication prototype is implemented using MATLAB to communicate with FPGA board via RS232. The MATLAB graphics use interface window is as shown in below

In that the matlab design the GUI which is shown in the following fig in that the GUI different tools are used like the radio button, push button, Edit button etc. and write the code for the push button .here the two push button are used one for the load the image and another used for the transmit the pixel values those button shown in the following figure

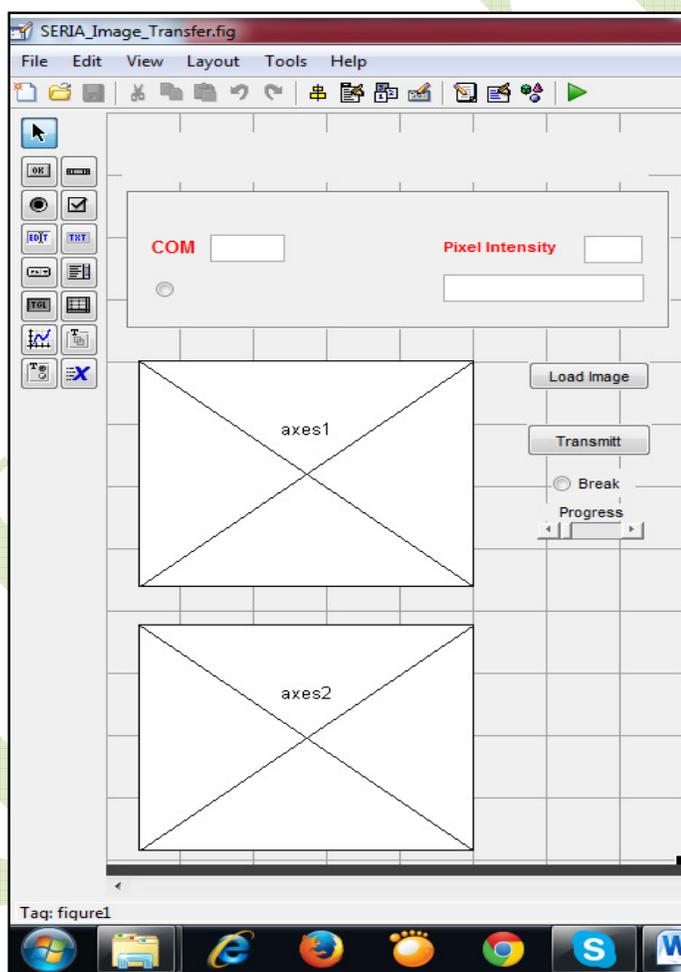


Figure 5: MATLAB GUI

### 3.1 CREATING MATLAB SERIAL PORT

Serial port= serial ('com3') //port creation  
Setting Parameter of the Port

- 1) Set (serial\_port, 'BaudRate', 9600)
- 2) Set (serial\_port, 'InputBufferSize', totalpixels)
- 3) Set (serial\_port, 'OutputBufferSize', totalpixels)

Writing and reading to/from the port

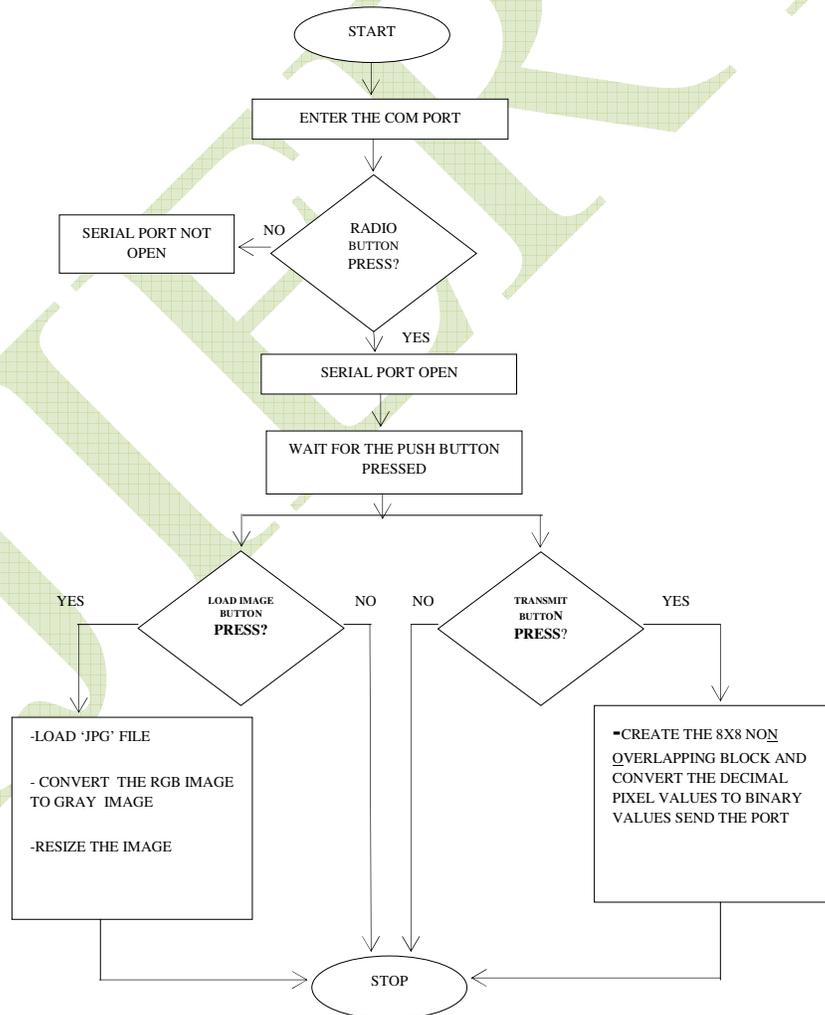
F open (serial port) //opens the port  
Fwrite(serial\_port, [0,12,4,5]) //writing binary data

A= fread (serial port, n) //reading binary data  
//n indicates no. of data

Closing serial port

Delete (serial port), Clear serial port

In the following the flow chart seen that the how the interface the Matlab GUI and the FPGA. Now the you are entered the port are connected FPGA board on MATLAB then you are press the transmit command button then the pixel values are transmitted to FPGA form MATLAB. Before transmitting the image pixel values these image are divided to the 8x8 Non Overlapping Block. Suppose the you can used the 128x128 size of any image then first this image are divided in 8x8 block and then send to FPGA by using the RS232 protocol.



**Figure 6:** Design low of the GUI

### 3.2 STEPEs FOR CREATE THE 8\*8 NON OVERLAPPING IMAGE BLOCK

- 1) Read the size of the image(sz)
- 2) Find the No horizontal block and vertical block

```
Nr =sz(1)/8  
Nc =sz(2)/8
```

- 3) Creating the 8x8 bock

```
for i=1:nr  
    indx_i=(i-1)*8;  
for j=1:nc  
    indx_j=(j-1)*8;
```

```
Block=I(indx_i+1:indx_i+8,indx_j+1:indx_j+8)
```

- 4) These 8x8 block are converting in one dimensional array

```
stream=Block(:);
```

- 5) These array are converting to that ascii values

```
Ch_str=char(stream);
```

- 6) Finding the decimal values of the character

```
D=abs(Ch_str(ptr))
```

- 7) Convert the these decimal values to binary

```
B=de2bi(D,'left-msb',8)
```

- 8) Finally send the decimal values to edit tool by using the command  
Set(handles.edit2,'string',stream(ptr));

And also send the binary string to the edit tool  
Set(handles.edit3,'string',B\_str);

### IMPLEMENTATION AND SIMULATION RESULT

Here we have used 128x128 pixel images. While synthesizing the HDL code, we have used internal FPGA RAM instead of on board memory, so the no. of LUTs used gone up to 90% from the initial 20% value.

128\*128 size of image are transmitted to Spartan 3 board via RS232 implement on FPGA and display the pixel values on the LEDs are available on the Spartan 3 board shown the following Matlab GUI simulation results.

#### 4.1 MATLAB RESULT

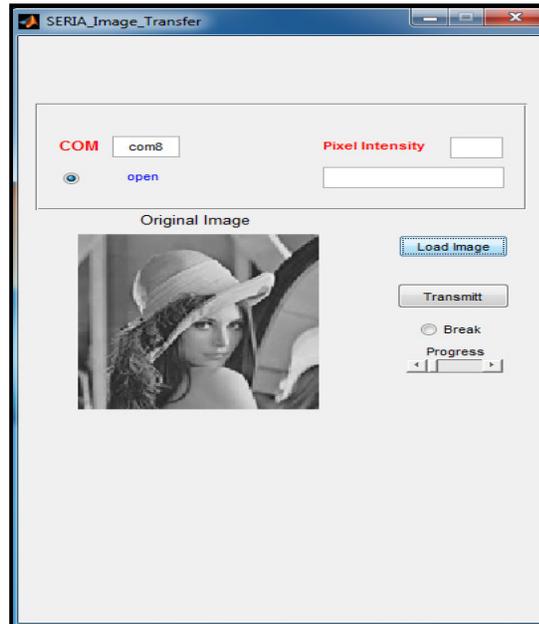


Figure 7: Select the port and Load the image

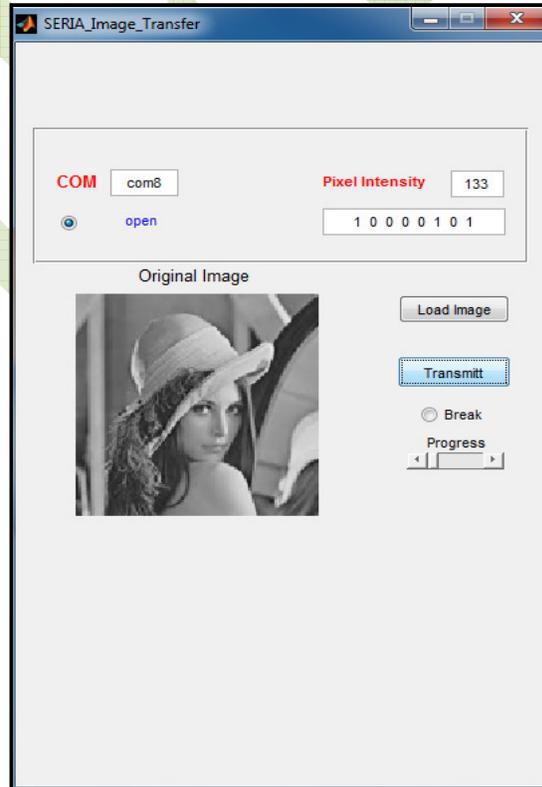
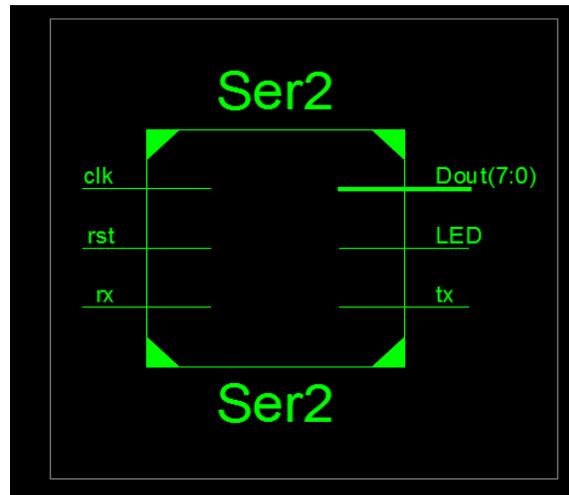


Figure 8: Transmit the pixel values to Serial Port(com8)

#### 4.2 RTL SCHEMATIC FOR RS232



#### 4.3 HARDWARE RESULT

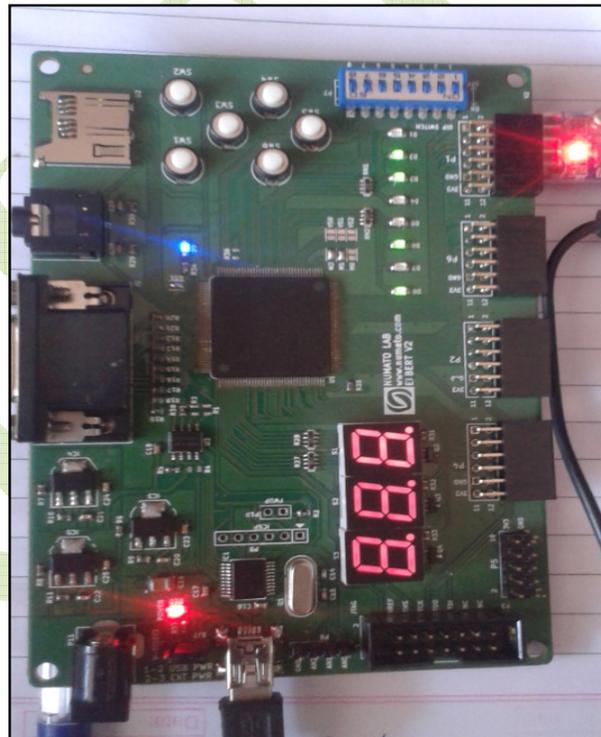


Figure 9: Transmit the pixel values of image to Serial Port (com8) on the hardware sparton3 and displays the pixel values on LED

## DEVICE UTILIZATION SUMMARY

**Table -1: Device xc3s50a-5tq144 Utilization**

DEVICE –xc3s50a-5tq144			
LOGIC UTILIZATION	USED	AVAILABLE	UTILIZATION
NO OF SLICE FLIP FLOPS	655	1,408	46%
NO OF 4 INPUT LUT'S	615	1,408	43%
NO OF USED AS LOGIC	615		
NOOF OCCUPIED SLICE	633	704	89%
NO OF BONDED IOBS	13	108	12%

## CONCLUSION

Converting Image to binary values will increase the efficiency of the system. Image processing applications required large memories, due to this memory control logic become vital in the image processing application. This Memory requirement problem has been reduced in this paper by near about 50%. But RS232 serial communication is simple to implement but the transfer speed is very less as compared to other communication techniques.

## ACKNOWLEDGEMENTS

Shital N. Gavade would like to thank Mr. R. N Patil, sir Lecturer guiding me through out to complete the work successfully and would also like to thank the HOD sir , and other staff members in the electronics and communication department for extending their help and support in giving technical ideas about the paper and motivating to complete the work effectively and Successfully.

## REFERENCES

- [1] Rashmi, Mukesh Kumar, Rohini Saxena “Algorithm and technique on various edge detection,” *An International Journal (SIPIJ) Vol.4, No.3, June 2013.*
- [2] Mr. Manoj K. Vairalkar, Prof. S.U. Nimbhorkar, “Edge detection of images using sobel operator,” *International Journal of Emerging Technology and Advance Engineering, Vol.2, January 2012.*
- [3] Han Xiaoru, Gao Yudong, “Design and Implementation of the Universal RS232-GPIB Interface,” *Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on.*