Novateur Publication's
International Journal of Innovation in Engineering, Research and Technology [IJIERT]
ICITDCEME'15 Conference Proceedings
ISSN No - 2394-3696

REAL TIME BLUETOOTH COMMUNICATION BETWEEN FPGA BASED EMBEDDED SYSTEM AND ANDROID PHONE

Kiran Aghav
Department of Electronics & Telecommunication, Pune University /
Vishwabharti Academy College Of Engineering, Ahmednagar, India

Dr. A.K.Kureshi
Department of E & TC (VLSI and Embedded System),
University of Pune, VACOEA,Ahmednagar, India

ABSTRACT
Applications in areas such as telehealth and household security often require wireless communication between low-power embedded systems and personal smartphones. This paper presents the design and implementation of a project that exploits Bluetooth capabilities in smartphones running the Android Operating System to communicate wirelessly in real-time with an FPGA-based embedded system. The use of FPGAs in such systems promises higher processing capabilities and lower power usage than traditional microprocessors, and has the added advantage of being reconfigurable for future development.

INTRODUCTION
In recent years, smart phones have become many people's primary personal computing device due to their vast capabilities, low cost, and mobility. This lead to a push towards interfacing these devices with other electronics such as medical monitoring and household security equipment.
By exploiting the Smartphone's wireless capabilities, a user is then able to control various other devices and monitor a wide array of sensors remotely [1]. As an example, the advantages of this ability in telehealth applications are numerous. Remote medical monitoring allows patients to collect and view health related data while on the go, improving the quality of life of those asked to wear sensors for tests. This data is then easily accessed by health care providers, given that smartphones are easily synced. that establishes a Bluetooth connection between an FPGA and a smartphone running an Android operating system (OS). The design is ideal for applications that require real-time monitoring of data such as heart rate and communicating the measured values instantaneously. The Android OS is the fastest growing OS on the mobile market. Furthermore, it is open source; thus allowing more ease and freedom when developing applications [2].

SYSTEM ARCHITECTURE
Xilinx's ISE was used to develop the circuit on the FPGA using VHDL. Android SDK as a plug-in, was used to develop the application that is responsible for the communication between the phone and the board. Android applications are written in Java, while some of the user interface features are coded in XML. Figure 1 shows the architecture of the system developed of the system, and can be configured to acquire data from sensors via its many communication interfaces such as UART, RS-232. port. The pins of interest are TX, which serially transmits bits to a receiver who buffers these bits at the RXD pin. RTS and CTS are 1-bit signals sent and received by the sender and receiver to indicate readiness for transmission.

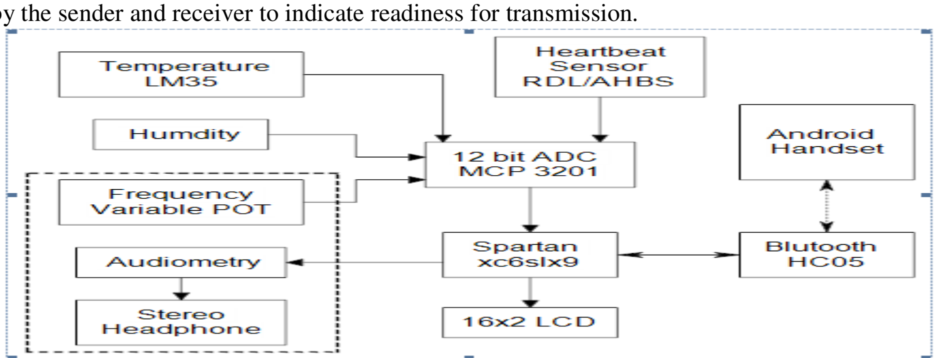

Figure 1: System architecture block diagram.

The block shown in figure 2 contains the code that builds the interface between the FPGA and the Bluetooth module via UART. On the RECEIVER block; TXD will receive the value from the UART interface (from Bluetooth) bit by bit then collect them together and output them as a single 8-bit bus to the FPGA through DATA_IN. DATA_IN_RDY works as a done signal to indicate the data is completely received and ready to be transmitted to the FPGA. On the TRANSMITTER block; DATA_OUT is an 8-bit bus of the data that would be transmitted to the Bluetooth. DATA_OUT_RDY is a 1-bit signal to indicate that there is data to be transmitted. The transmitted data will be sent through the RXD signal bit by bit to the UART interface. CTS is clear to send and used in software handshaking. CTS will not be actively used in the current design. It is put there in case there is a future expansion to the design that needs a CTS signal.
A clock convertor inside each converts the clock speed to support a 9600 baud rate. To limit inaccuracies in the transfer of data from the Nexys 2 board, the sampling baud rate at the UART will be set to 16 times the clock or the rate of the incoming bits. This will mean that each bit is sampled 16 times, decreasing the probability of errors during transfers. The focus of this paper will be on the implementation details regarding the application's receiving of data via Bluetooth.
There, a search for devices in the vicinity is attempted and a "serial" device is seen along with the board's unique ID number. Clicking on this device and selecting pairing prompts a box that requests a PIN. Phone should be connected to the device. Bluetooth connections made between two devices imply that one must act as a server and the other as a client. For this project, the Android device acts as the server. Hence, the application must be configured to hold an open Bluetooth Server Socket.
This is done as such in the Activity. Using the call below, the communication protocol is set to communicate with the board. An instance Bluetooth Socket is acquired, from which connect(), get Input Stream(), get Out put Stream can be called. At this point, the device is communicating with the phone serially as if it were an RS-232 cable, sending 8 bits of data and collecting it in a byte.. Note that the UUID here is not random. It is the one that is needed to connect to a serial Bluetooth device.

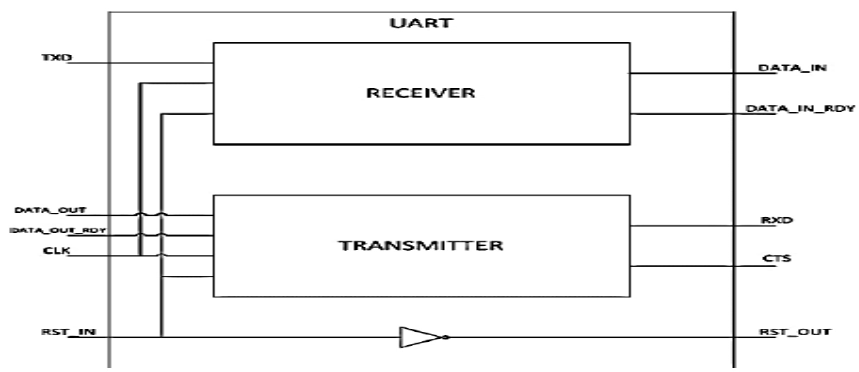

Figure 2: UART block diagram.

The string is an identifiable name of your service, which the system will automatically write to a new Service Discovery Protocol (SDP) database entry on the device (the name is arbitrary and can simply be your application name). The UUID is also included in the SDP entry and will be the basis for the connection agreement with the client device. UUIDs (Universally Unique Identifier) are 128-bit format for a string ID that is used to uniquely identify information and are big enough such that any random string can be chosen without a clash. Once you have established a connection, one must manage it by: · Get the Input Stream and Output Stream that handle transmissions through the socket, via getInput Stream() and getOutput Stream(), respectively. ·
Read and write data to the streams with read Byte[] and write Byte[].the Activity java file also handles some exceptions such as no devices found or no Bluetooth adapter found and attempts to correct them. The second main java file is that of the View. The View file does two things. Firstly it handles some display parameters such as font size and color, background, and where and what to display. The second thing it does is read in the 8 bits of data received by the board, collects them in byte format and finally assigns them to a variable integer called heart_rate. Once this is done, the heart rate sent by the board becomes stored in the phone, and is ready to process and use. The application can now process data in a number of ways, such as storing, uploading and analyzing. The application can issue warnings, control actuators or arrange data in tables for further study.

ISSN No - 2394-3696

CONCLUSION

The communication established between FPGA and smartphone can serve as the infrastructure in many other applications where the high processing capability and low power consumption of FPGAs would like to be exploited. Our work can also serve as instruction for those seeking to establish reliable wireless communication between an FPGA and a smartphone running the Android OS.

REFERENCES

[1] Chalivendra, G.; Srinivasan, R.; Murthy, N.S.; , "FPGA based reconfigurable wireless sensor network protocol," Electronic Design, 2008. ICED 2008. International Conference on , vol., no., pp.1-4, 1-3

[2] Jianping Cai; JianZhong Wu; Minghui Wu; MeiMei Huo; , "A Bluetooth toy car control realization by android equipment," Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on , vol., no., pp.2429-2432, 16-18 Dec. 2011

[3] Martel, M. Normand. "Remote medical monitoring system", US Patent 2006/0122469 A1, Nov 16, 2011

[4] "Nexys 2"
http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,789&Prod=NEXYS2, Accessed May 2015

[5] "Pmod BT"
http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,401,865&Prod=PMOD-BT., Accessed May 2015.