ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Software Defined Radio with Ethernet Interface

Ligi K, Chandrasekar P
Vel Tech Dr.RR & Dr.SR.Technical University, Chennai, India

Abstract*—initialization of transmitter parameters can be done locally by setting values on the transmitter itself or remotely using a pc. a radio transmitter design has to meet certain requirements. these include the frequency of operation, the type of modulation, the stability and purity of the resulting signal, the efficiency of power use and the power levels required to meet the system design objectives. local control of a transmitter is not sufficient in defense and other security purpose. This project aims to develop a suitable gui using labview and create an Ethernet ieee 802.11 interface to communicate with the transmitter in the client end. transmission control protocol (tcp) is used as the communication protocol. All signal parameters such as frequency, transmission power and modulation scheme can be set using the gui. a raspberry pi board acts as the server. An adf7020 transceiver is used as the radio whose parameters is setting through gui and is interfaced with raspberry pi board through usb port.

***Index Terms*— Software Defined Radio, Raspberry Pi, TCP, ADF7020 Transceiver, PIC18F4620, Client server Communication.**

I. INTRODUCTION

Software defined radios are radio communication systems whose hardware are implemented and replaced with software [1]. Parameters settings of a transmitter defined by software are the proposing method that avoids hardware parts like knobs and meters in the transmitters. Radio transceiver ADF7020 is a programmable transceiver which is controlling using LabView software and Ethernet interface. Ethernet interface is done using TCP communication protocol that ensures reliability in communication. Thus LabView PC acts as a client and transceiver end acts as a server. Ethernet interface between LabView and transceiver enables the user to control the transceiver remotely. There are two transceiver modules of ADF7020. Both are programmed and controlled using PIC18F4620 microcontroller. ADF7020 transceiver will work fine from 433MHZ to 960MHZ and 63 power levels that range from -16dBm to +13dBm with each step size of 0.45dBm.Transceiver are programmed for 21.6KHZ deviation of GFSK, 1.416Kbps data rate, Bandwidth of 100KHZ and phase frequency detector as 11.0592MHZ. The parameters like frequency, power level, data rate, sync word etc can be changed. The proposed model aims in changing the two parameters like frequency and power level of transceiver in the transmitter section. Software control of the parameters is done using LabView [7]. Frequency of transmitter can be varied from 433.1MHZ to 433.9MHZ. Transceiver frequency should be 200KHZ less than transmitter frequency for better communication. Thus the frequency range of receiver varies from 432.9MHZ to 433.7MHZ. Power levels like 3rd, 33rd, 63rd are programmed for simulation. Radio control using LabView program is done through Ethernet interface. LabView software develops a message packet and sends to transceiver using TCP communication protocol [7]. Thus a LabView PC act as a client and Raspberry Pi board model B+ act as server which as an Ethernet 10/100Mbps. Transceiver nodes receives the data frame through the serial port from the Raspberry Pi. TCP communication is established between Raspberry Pi and LabView PC by assigning static IP and port number in the range of 1000s. Receiver module is programmed using MPLab and run in debugger mode using PICkit2. Upon data reception receiver module will blink green led and a yellow led for error condition.

II. SOFTWARE DEFINED RADIO DESIGN

The proposed software defined radio design has two sections: a) The transmitter section and b) The Receiver section. Both sections have a PIC transceiver node of ADF7020 whose parameters are programmed and set using LabView.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

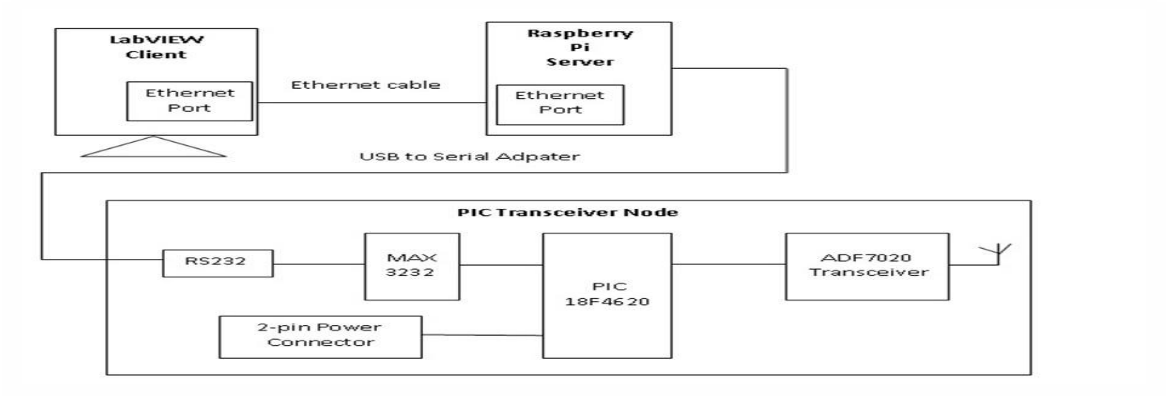

Fig 1: Transmitter Section Block Diagram

The transmitter section block diagram in Fig 1 shows the hardware connection details. LabView client PC uses LabView 8.1 evaluation software to create codes and blocks to control the parameters such as frequency and power levels of the transmitter. Client is implemented with TCP protocol configured with an ip address of server and port number. Server role is done by the Raspberry Pi board which has Ethernet of 10/100Mbps. An Ethernet cable interconnects the LabView PC and Pi board.

In PIC transceiver node PIC18F4620 is connected with the ADF7020 transmitter with SPI bus and RS232 serial port is interfaced using MAX3232 logical level converter. Power connecter pins are provided to power up the node using battery. Raspberry Pi and PIC node is interconnected in serial port. USB port of pi board is connected with RS232 port in PIC node using USB to serial adapter. Serial communication through USB in Pi is done with 9600 baud rate. The Fig 2. shows the Receiver block diagram. PIC transceiver node is same as that in the transmitter section. PIC node is connected with the PC using PICKit2 debugger in ICSP pins to create target and then run the MPLAB coding. A USB to serial adapter is connected in PC with RS232 port to obtain the received data frame in HyperTerminal window. In order to verify the frequency settings in transmitter section, the receiver module should be programmed for the corresponding frequency value ie 200KHz deviation is adequate. After transmission of data the receiver collects the data frame and blinks green led. This green led blinking indirectly verifies the frequency settings in transceiver. Power levels settings can be verified by measuring the maximum distance between transmitter and receiver such that the receiver should receive the data frame and show the green led blinking at that position.

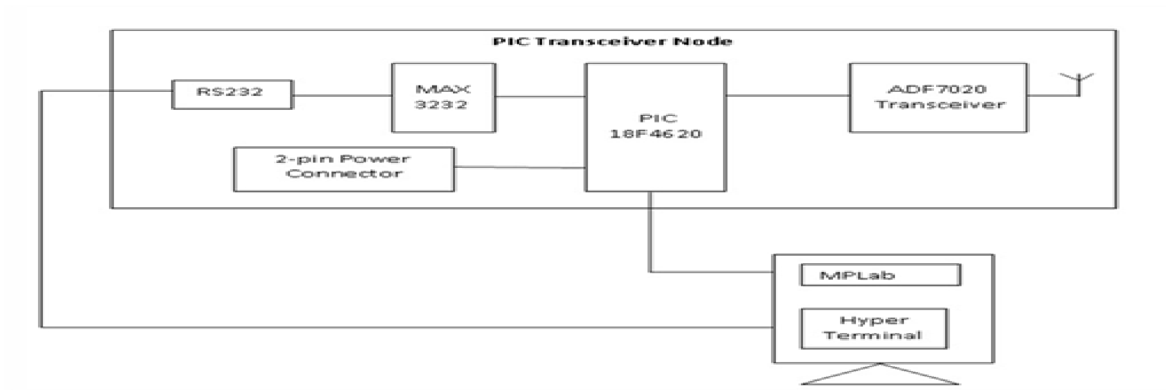

Fig 2: Receiver Section Block Diagram

A. ADF7020-1 Transceiver

The ADF7020-1 ISM band transceiver is a highly flexible transceiver which allows the programmer to change the physical layer parameters, including operating frequency, modulation, transmission power and data rate. Table 1 shows the capabilities of the ADF7020-1 RF transceiver [8].

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Table 1. ADF7020-1 Capabilities

S.No	Features	Capabilities
1.	Frequency bands	135 MHz to 650MHz (direct output) 80MHz to 325MHz (using divide by 2 mode)
2	Data Rates	0.15 kbps -200kbps
3.	Output power	-16dBm to +13dBm
4.	RSSI	7 bit readback
5	Modulation	ASK/FSK/OOK/GFSK
6	Temperature Sensor	Range: -40°C to +85 °C

The ADF7020-1 has an on chip 7 bit Analog to Digital Converter (ADC). As the 7 bit ADC is used to sample the RSSI signal, the signal strength reading is accurate up to 0.5dB. ADF7020 registers like registers 0 to registers 13 can be programmed to change the parameters of the transceiver. Frequency parameters are defined in register0, power levels and data rate are defined in register2, sync word is defined in the register5 and RSSI settings are in register7. The proposed model is changing the two parameters frequency and power levels. In register0 DB 4 to DB 18 represents fractional_N value and DB 19 to DB26 represents integer_N value which is programmed to change the frequency values [3]. Bit values can be found using the formula

Frequency= Crystal frequency/R*(Integer_N+Fractional_N/2^15)

Where, R (R counter value) =1 and crystal frequency=11.0592MHz.

To establish communication between transmitter and receiver frequency deviation among them must be 200KHz. Table 2 shows the different register0 values for transmitter and receiver.

Table 2. Adf7020 Register0 Values

Transmitter frequency	Register0 values	Receiver frequency	Register0 values
433.9	7139E390	433.7	7939BE80
433.7	7139BE80	433.5	793995C0
433.5	713995C0	433.3	793970C0
433.3	713970C0	433.1	79394BB0
433.1	71394BB0	432.9	793926A0

Power levels are set using DB 9 to DB 14 bits in register2. Power value in dB is given as power in dB= -16+(0.45*power level) [3]. We have taken five frequency values and four power levels for the simulation. ADF7020 evaluation software of Analog devices is used for the register value verification. We set modulation scheme as GFSK, data rate as 1.416 kpbs, IF bandwidth of 100KHz and charge pump current (icp) as 0.9mA.

B. PIC Transceiver Nodes

The Fig 3 shows the picture of the PIC TRANSCEIVER node with battery. The core of the PIC TRANSCEIVER node is the PIC18F4620, an enhanced 8 bit flash microcontroller from Microchip. The features of the PIC18F4620 are Operating Frequency of 40 MHz, Program Memory (Flash) of 64 KB, Data Memory (SRAM) 4 KB, thirteen 10 bit ADC Channels, sixteen I/O Lines, four Timers (8/16bits), two Comparators, one EUSART and one CCP [6][2]. The Analog Devices ADF7020-1 is interfaced with the processor through the Serial Peripheral Interface (SPI). The node has an RS232 port which has MAXIM MAX3232 to convert 0/3.3V signals of the processor to the RS232 level signals (-12V/12V) for the computer interface.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Fig 3: PIC Transceiver Node

The transceiver node has red, green and yellow LEDs, which are used to indicate the RF transmission of a packet, the RF reception of a packet and the reception of an erroneous packet respectively. The In Circuit System Programming (ICSP) connector in the node is used to program the processor by the PICKit2 programmer [8]. One node is programmed as transmitter and other as the receiver to implement software defined radio communication between them.

C. TCP Client Server

The Transmission Control Protocol is a connection-based protocol that establishes a connection before data transmission. TCP ensures reliable transmission across networks, delivering data in sequence without errors, loss, or duplication rather than UDP. TCP reassembles the packets the receiver end. In establishing TCP connections a specific address and a port number is necessary. A number between 0 and 65,535 represents a port. The process involves opening the connection, reading and writing the information, and closing the connection. The LabView PC realizes a client that initiates a connection to the Raspberry Pi server.

D. Raspberry Pi

Raspberry Pi is a computer with basic level of functionality. Instead of having a separate CPU, a GPU, a USB controller, and memory chips, Raspberry Pi uses a systemona chip with all the components on a single chip which boots up from a micro SD memory card. It has 256MB of RAM and a 700MHz ARM-11 processor. The Model B+ also sports four USB ports, HDMI out, a 10/100 Ethernet port and 3.5mm audio jack. We are using Raspbian OS in the Raspberry Pi. A server application using Ethernet port is developed in the Raspberry Pi using linux functions. To establish serial communication between pi board and PIC TRANSCEIVER node we disabled the serial port console login which is the default setting [9][10].

III. SIMULATION RESULTS

Simulation of the proposed system can be done with the hardware set up shown in Fig 4 and with software's such as LabView, embedded Linux and MPLab. LabView creates a GUI to set parameter values and establishes a TCP Ethernet interface between LabView and Raspberry Pi. Embedded linux coding in Pi board establishes a server and serial communication in USB to serial port in PIC node. MpLab IDE develops the PIC code that functions as the transmitter and receiver module with ADF7020 interfacing.

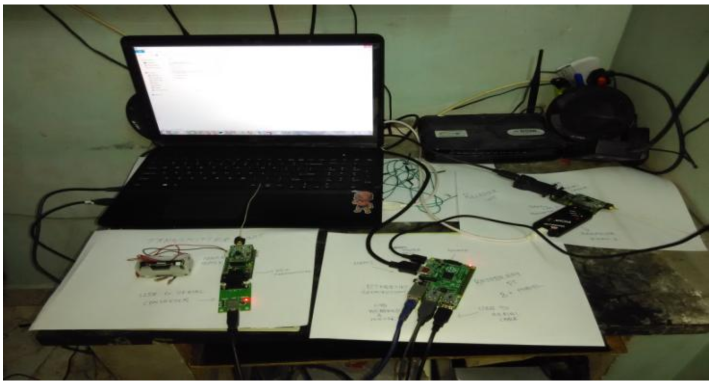

Fig 4: Hardware Setup

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

A. TCP Client Simulation in LabView

TCP/IP functions are built into LabView to give a way of transmitting information across the network reliably. Function blocks prerequisite for TCP client creation are TCP Open, TCP Write, TCP Read and TCP Close Connection function. Block diagram VI establishes a TCP connection between PC and Raspberry Pi board using a static IP and a port number. Values from the control buttons and message are framed to form a packet and are written to the port specified using TCP write function. After data sending operation communication is terminated using the TCP close function. Front panel constitutes two control icons to vary frequency and power levels with step increment size of 0.2 MHz and 30 steps respectively. Default values are 433.1MHz and 3rd power level. An OK button is used to send the frame over the network.

1. Packet Format Created in LabView

Packet format to transmit frequency and power values is framed in LabView and the same is received in the Raspberry Pi server. PIC TRANSCEIVER node changes the ADF7020 parameters through the packet received in the RS232 from the Raspberry Pi USB. Packet format is shown in the Fig 5.

Frequency parameter	Power variable	Message	Carriage return
One byte	One byte	X bytes	One byte

Fig 5: Data Packet Frame

Each frequency value and power level has corresponding variable in the packet frame. For example the packet for 433.1MHz frequency and 33rd power level is 11hello\r.

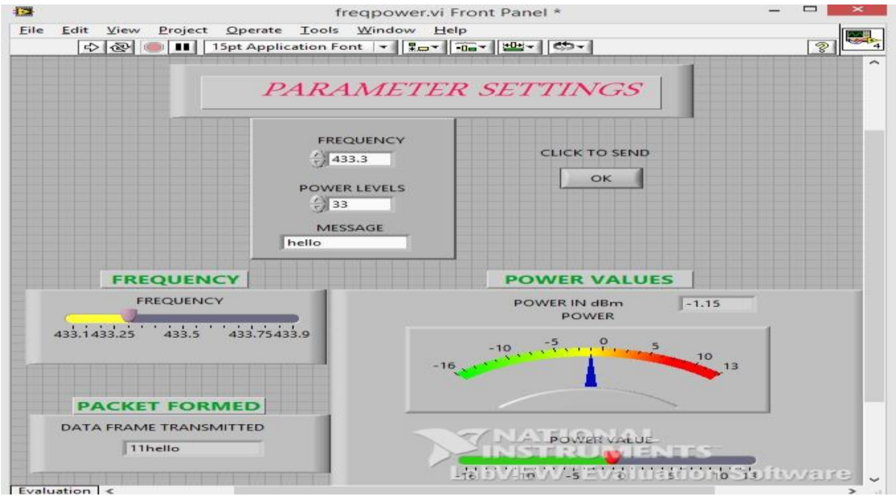

Fig 6: LabView Front Panel for Parameter Settings

B. Server Simulation in Raspberry Pi

In build Ethernet port in Raspberry Pi makes it suitable of being a server to receive packet data from LabView. A suitable server coding is build using system calls such as socket () to create a socket, the bind () system call is used to bind the socket on particular address, the listen () system call listens the remote computer and the accept () system call accepts the client connection. Compiled and build the server coding in the Geany IDE installed in the Raspberry Pi. Thus client – server communication using TCP can be established between LabView PC and Raspberry Pi. TCP server simulation result of receiving packet frame send by the LabView client is shown in Fig 7. To perform serial communication between pi and PIC node serial console login is disabled by editing two files

a) ./etc/inittab to add # character to the beginning of line.

#T0:23:respawn:/sbin/getty L ttyAMA0 115200 vt100

b). /boot/cmdline.txt to remove all AMA0 parameters

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Testing of the serial port can be done using the minicom, HyperTerminal on PC and USB to serial adapter. Setup a connection at 9600 baud and run minicom using the command: minicom b 9600 - o D /dev/ttyUSB0 and serial port communication can be verified. Command configures USB0 device with 9600 baud rate to communicate through minicom. In linux all devices are treated as files. Functions like serialOpen(), serialGetchar(), serialPuts() etc will perform opening of a serial port in specified device with specified baud rate, read the characters and write the string on the port. Program is compiled and executed with gcc compiler. Serial communication code will write the data received in server buffer to the USB0 port and USB to serial adapter will transmit the same to the PIC TRANSCIEVER node.

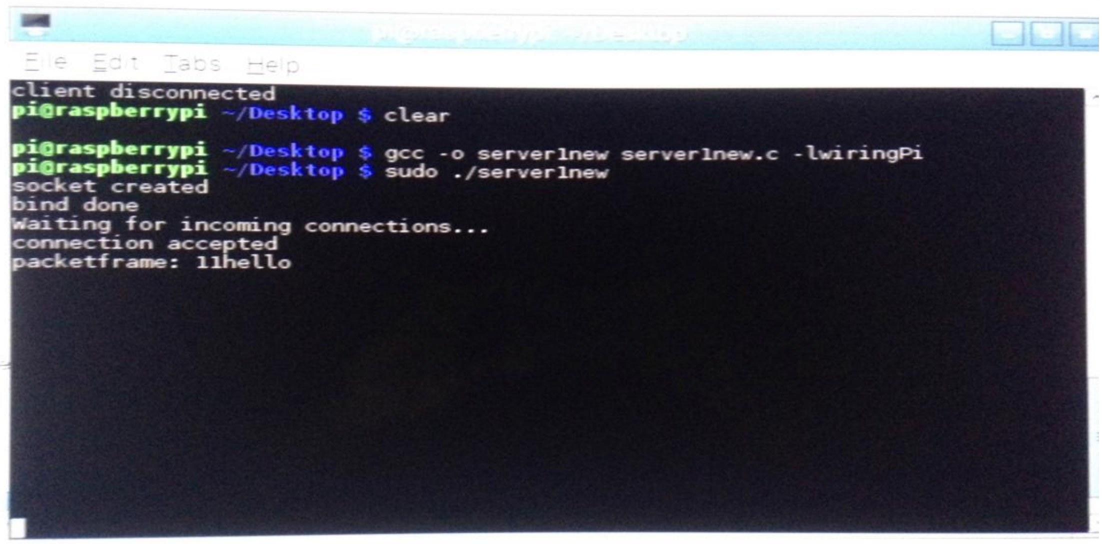

Fig 7: TCP Server Simulation in Raspberry Pi

C. Transmitter Node MpLab Simulation

In MPlab device is selected as PIC18F4620, configuration bits as WTD disable and HS oscillator enabled. In transmitter section five frequency values have selected and serial interrupt in UART occurs at reception of single bit of data from raspberry pi. ISR will perform transmit mode settings like register0 and register2 values for frequency and power according to variable values in the packet frame received in the buffer. ADF7020 transmitter will send the data or message frame stored in buffer in the selected frequency and power level. Red LED blinking indicates transmission of each packet.

D. Receiver Simulation Result in HyperTerminal

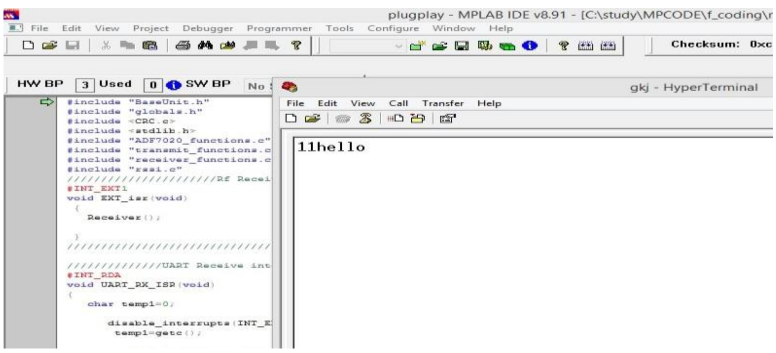

Fig 8: Received Data Frame in Hyper Terminal

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Receiver section is programmed to receive the data frame when an RF receive interrupt occurs. The ISR will send the buffered data to serial port and blinks a green led. Transmitter is programmed in programmer mode using pickit2 and later the module is powered using battery. Receiver section is programmed and run in debugger mode using pickit2. In order to receive the data frame from the transmitter, set ADF7020 register 0 values according to Table 2. For instance to receive data from transmitter which is working in 433.1MHZ, the receiver ADF7020 register value should set as 433.9MHZ. Received data frames can be viewed and verified using hyper terminal window opened in particular COM port , 9600 baud rate, 8-bit, no parity bit configuration as in Fig 8. Power lever can be verified by measuring the range of signal from transmitter. Increase in the power level will produces the maximum range of communication.

IV. CONCLUSION

A software defined radio using LabView software along with the Ethernet interface between LabView and transceiver through the Raspberry Pi board has discussed. Verification of the operations of two parameters like frequency and power level of transceiver ADF7020 were successful. Simulation and implementation of hardware were also done as per the block designs. Frequency was verified using the receiver module which is already set with the corresponding frequency for transmitter module. Power levels were verified by measuring the range of reception in the receiver module. The range is this maximum distance from the transmitter module such that the receiver node should blink green led as the reception indication. The higher the power level the greater the range. The proposed system can be enhanced with wireless communication function in LabView software. With further experiments more programmable parameters of the transceiver can also be controlled.

REFERENCES

[1] Software Defined Radio: Architectures, Systems and Functions (Markus Dillinger, Kambiz Madani, Nancy Alonistioti) Page xxxiii (Wiley & Sons, 2003, ISBN 0-470-85164-3).

[2] Microchip Technology Incorporated, Printed in the U.S.A, 2004, Internet, http://ww1.microchip.com/downloads/en/DeviceDoc/39626e.pdf.

[3] Analog Devices, RF transceiver ADF7020-1, U.S.A. 2005,Internet,http://www.analog.com/static/imported-files/data_sheets/ADF7020-1.pdf.

[4] Mitola III, J. (1992). Software radios-survey, critical evaluation and future directions. National Telesystems Conference. pp. 13/15 to 13/23.

[5] First International Workshop on Software Radio, Greece 1998.

[6] Muhammed Ali Mazzidi, Rolin D Mckinlay, Danny Causey,"Pic Microcontroller and Embedded Systems: Using Assembly and C for Pic 18", United States, Pearson Publication.

[7] "Software defined radio resources", [08/Sep/2014].

[8] M. Ramakrishnan and P. Vanaja Ranjan, "PICSENSE – A Wireless Sensor Network Testbed", International Journal of Recent Trends in Engineering, Vol 1, No. 4, May 2009.

[9] Eben Upton, Gareth Halfacree,"RaspberryPi",www.cs.unca.edu/~bruce/Fall14/360/RPiUsersGuide.pdf [2015].

[10] Charles severance "Eben Upton: Raspberry Pi" Computer, vol 46(10), 2013, pp.14-16, DOI: 10.1109/MC.2013.349 Software Defined Radio: Architectures, Systems and Functions (Markus Dillinger, Kambiz Madani, Nancy Alonistioti) Page xxxiii (Wiley & Sons, 2003, ISBN 0-470-85164-3).

AUTHOR BIOGRAPHY

K.Ligi is currently pursuing her master's degree with specialization in embedded system technologies at Vel Tech Dr.RR & Dr.SR.Technical University, Chennai, India. She received her bachelor's degree with specialization in Electronics and Communication Engineering from Rajiv Gandhi Institute of Technology Government Engineering College, Kerala. She worked as Guest Lecturer in the Department of Electronics and Communication Engineering at College of Engineering Kallooppara, Sylvan Hills of Pathanamthitta, Kerala, India. Her research area includes embedded system design and Embedded C programming.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 4, Issue 2, March 2015

Dr. P. Chandrasekar was born in Rajapalayam, Tamilnadu in 1976. He received Bachelor of Technology in Electrical and Electronics Engineering from Pondicherry Engineering College, Pondicherry in 1997 and Master of Engineering in Power Systems Engineering from College of Engineering, Guindy, Anna University, Chennai in 2002. He has been working in the teaching field for about 17 years. He received PhD in Power Quality from Anna University, Chennai in 2013 and presently Head of Department in Electrical and Electronic Engineering at Vel Tech University, Avadi, Chennai. He is also associated with professional societies namely Indian Society for Technical Education (ISTE) and Institute of Electrical and Electronics Engineers Inc., (IEEE), USA. His research interests include signal processing applications of power quality monitoring and improvement techniques, Wavelet analysis, artificial intelligence to power system and Renewable Energy Systems.