Journal of Engineering and Applied Sciences 11 (9): 1931-1937, 2016
ISSN: 1816-949X

IOT Based Home Automation Using FPGA

[1]Azeem Mohammad Abdul, [1]B. Mmurali Krishna, [1]K.S.N. Murthy, [1]Habibulla Khan,
[1]M. Yaswanth, [1]G. Meghana and [2]G.L. Madhumati
[1]Department of ECE, KL University, Green Fields, 522502 Vaddeswaram, Andhra Pradesh, India
[2]Department of ECE, DIET, 521139 Vijayawada, Andhra Pradesh, India

Abstract: Internet of Things (IOT) had many applications in several domains; it also strides into smart homes. Controlling appliances with IOT can be easily done using Smart phone through Android apps. Home automation is one of the profound in day to day applications. Due to hasty progress in technology, Wireless Fidelity (Wi-Fi) has brought revolutionary change than compared to Wired LAN communication. Existing wireless communication devices such as Bluetooth, ZigBee and NRF24L01 etc. are limited to short range. IOT uses Wi-Fi to exchange data wirelessly for large distances using Internet. IOT module (ESP8266) is used to control the home industrial appliances in remote areas anywhere in the world. Serial Communication exchanges the data between FPGA and IOT module. Home appliances are controlled using FPGA which receives commands in serial communication from IOT Module through smart phone app. Compare to existing home automation; IOT based home automation can update device status with E-Mail alerts and also in web with IP address which can be password protected. Due to its high precision and smart phone technology helps for physically challenged and senior citizens.

Key words: Home automation, FPGA, IOT (Internet of Things), android, apps

INTRODUCTION

The enormous development in present day developing innovation, people are adjusted to these innovations from multiple points of view. Communication means sharing information from one point to another point. It can be done in mainly two ways, wired communication and wireless communication. Wireless communication plays a dominant role in present scenario because of its high security and easy accessibility. IOT (Atzori *et al*., 2010; Rudramma and Krihna, 2013), technology dominates Bluetooth technology (Sharon Panth and Mahesh Jivani) because it is limited to certain area. But IOT can be accessed by IP protocols and be connected through web and can be accessed throughout the world. Now present Bluetooth technology can be accessed only in terms of metres but IOT technology can be accessed throughout the world by means of IOT modules which are very cheap in the market. IOT is the upcoming trend and future will depends on this. There are number of Bluetooth Modules available in the market but compared to them IOT modules are more reliable and cheap and also had many applications such as home automation (Li *et al*., 2015) intelligent shopping system, etc. Home automation can be done easily with the help of IOT technology. Monitoring and controlling home appliances (Yan and Shi, 2013) can be done using IOT modules. The IOT module used here is ESP8266. The internet of things is the system of physical items or things installed with hardware, programming, sensors and system network which empowers these objects to gather and trade information. Home automation is likewise a noteworthy stride forward with regards to applying IOT. Every one of these advances add to the various rundown of IOT applications. Presently with IOT, we can control the electrical gadgets introduced in our home while we are sorting out our records in office. Our water will be warm when we get up in the morning for the shower. All credit goes to brilliant gadgets which make up the savvy home. Above all else when planning a home automation framework we need to think about as some issues such as the client ought to have the capacity to associate wth that IOT module from any gadget (Android mobile) he would wish to. He ought to have the capacity to change the host from one gadget to other gadget and that module ought to work in like manner. On the off chance that any shortcoming happens it ought to have the capacity to analyze it and the framework to work promptly when a direction is given to enhance the way of remote innovation. FPGA board is utilized here as it can give high security (Han *et al*., 2006) to our framework.

Corresponding Author: Azeem mohammad abdul Department of ECE, KL University, Green Fields, 522502 Vaddeswaram, Andhra Pradesh, India

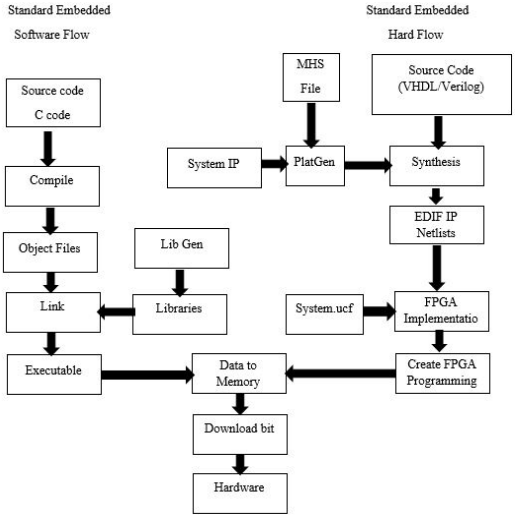

Fig. 1: EDK design flow

EDK design flow of XILINX: December 10, 2016Xilinx developed Xilinx Integrated Software Environment (ISE) is used to generate net lists, Hardware Description Language (HDL) designs can be analyzed which enable the developer to compile user designs, look at the diagrams of RTL schematic, focus on the timing analysis, simulation of developed design and concentrate on device with the designer .Xilinx ISE is generally used for generating net lists and design. Xilinx also contains various tools such as EDK (Kasai, 2013) (Embedded Development Kit), SDK (Viejo *et al*., 2006) (Software Development Kit).

EDK (Embedded Development Kit): EDK provides a set of design tools that happen to be based on a prevalent framework which permit to design a comprehensive system for the execution in a Xilinx FPGA device. It consists of: XPS (Xilinx Platform Studio).

- ESTs (Embedded System Tools suite)
- Software Development Kit (SDK) can be utilized to develop software application based on embedded

EDK has numerous frequently used segments where several varieties of systems could be created by using these kinds off segments. To create each of our own segments EDK uses IPIF (Intellectual Property Interface) library in order to perform similar functionality between various processor segments. EDK also gives us IP Interconnect bus protocols which are simply used instead on the subject of Processor Local Bus (PLB) bus protocol directly. Employing this IPIF module with suited specification matches our requirements which reduce design and test effort by huge amount. Xilinx Platform Studio (XPS) is used by BSB (Base System Builder) for producing a simple processor system which in turn may use custom segment. Hardware Description Language templates could be produced by using create and import (create mode) segment wizard. To open the .npl file that produced in create mode project navigator is utilized. Also add extra pattern to the segment top template file to implement our custom operation in user logic.vhd file (Fig. 1).

Intellectual Property (IP) Core generator stimulates the design time by offering entry to highly segmented IP for FPGA and is usually included in ISE design suite. From frequently used memories and FIFOs these user programmable IP functions range with in complexity. This building designs be quicker with highly optimized IP by designers. Generating HDL to configure FPGA system elements like Multi-Gigabit Transceivers (MGTs), Peripheral Component Interconnect (PCI) and Ethernet convey hard blocks by making use of the logiCORE Graphical User Interface (GUI)-based customizers. Right after generation of IP core, the top component is synthesized after which all of us implement the design that should be done. This design associated with Translate, Map, Place and route. Following implements the design .bit file is to be generated which often will be downloaded to the FPGA board.

XPS (Xilinx Platform Studio): XPS offers an environment for generating both software and hardware specification moves for embedded systems. XPS provides an editor and a project management interface to create and edit source code provides modification of tool flow construction options, furthermore offers a graphical system manager for the connection of processors, peripherals and buses. It has the ability to include and edit core variables. They have the ability in order to generate and modify the Microprocessor Software Specification (MSS) file. That provides an ability to create and view a system block diagram. It facilitates multiple-user software program applications in addition to project management.

SDK (Software Development Kit): SDK is an interdependent GUI to (XPS) and offers a development in order to software program projects. SDK is based on the Eclipse open-source standard, provides rich C/C++ code and compilation environment and project management. It

Fig. 2: ESP-01

offers application build configuration in addition to programmed make file technology. Navigating error is their special feature.

MATERIALS AND METHODS

IOT module (ESP8266, ESP-01): ESP8266 (ESP-01) is a Universal Asynchronous Receiver/Transmitter (UART) to Wi-Fi Module. ESP-01 Module is very cheap, abundant available in the market and easy way to connect any FPGA wirelessly to internet. AT commands are needed to connect wirelessly to internet. By simply connecting FPGA to ESP-01 Module, Serial communication can be done easily using internet. As there are several modules (ESP-01 to ESP-13), ESP-01 Module is used here and firmware is set to 9600 Baud Rate. ESP-01 consists of 8 pins, they are TX, Rx, RST, CH_PD (chip enable) and two GPIOs (GPIO0, GPIO2), Vcc and Gnd. ESP-01 WIFI Module comes with an inbuilt firmware which supports serial interface and can be controlled using AT commands. This module is not just a simple serial to Wi-Fi transceiver. It is comprised of a 32 bit processor (80 MHz), 512 kB SPI FLASH , 64 kB SRAM, 96 kB DRAM, GPIO Pins and Wi-Fi transceiver is shown in the Fig. 2.

To upload firmware a following procedure should follow, i.e., a USB-TTL converter should be in between ESP-01 boar and PC. As the ESP device requires 200 ma current which cannot be sourced from USB-TTL module but using a separate power b source of 3.3 V is reliable CH-PD is CHIP ENABLE which is always directly connected to + 3.3 V. TX of ESP is connected to RX of USB-TTL module RX is connected to TX. Vcc is connected to +3.3 V, GND is made common for all. RST pin should be floating or else can be used to reset the module by pulling it high. GPIO0 pin is connected to GND to upload firmware. GPIO2 also be floating till we upload the firmware After uploading firmware, this GPIO0 connection must be removed from GND make RESET pin high so that it leaves the Flasher mode. Now GPIO pins can be used to control the appliances through relays by FPGA.

After the firmware flashing, ESP8266 module settings changed. The default baud rate now is 9600 and can be modified using AT+CIOBAUD command. The terminal software Putty is opened in PC and set it in serial mode with 9600 baud and select the COM port allotted to USB-TTL converter. The ESP 8266 module is now is connected to PC through USB-TTL module. Type following commands, AT and then the module returns ok, i.e., it is in serial mode. AT+CWMODE = 1, for Station mode. AT+CIPMUX = 1, to enable Multiple connections before starting server. AT+CWLAP, to list the surrounding Access Points near ESP module AT+CWJAP = ”ssid”,”password”, to join the AP using the SSID name and its password. AT+CIPSERVER = 1, 80 to start a server on port 80. AT+CIFSR, displays the IP address, STAIP is the IP address we need to port forward. Now open the app in android mobile and connect by the STAIP and port allotted and make sure the PC and Android mobile connected by same server.

Design procedure: In this study, we show a home automation framework which is secured and adaptable that is controlled utilizing an android application as a part of a mobile phone. FPGA board is utilized to control the home appliances which are associated through transfers by means of FPGA board. The correspondence between the IOT module and the mobile phone is remote and afterward that module is serially associated with the FPGA board through RS-232 for the control of home appliances. The general engineering of the framework is appeared in Fig. 3.

The XPS Generated Block Diagram is appeared in Fig. 4. Android application is produced in JAVA script as indicated by the IOT module details. Adjustments in application can be made. Initially the IOT serial module needs to match it with the close Wi-Fi accessing gadgets. The application created in FPGA to control the appliances and the secret key is utilized to verify to make the correspondence can be set by client and once blending is finished the module gets the orders from the mobile

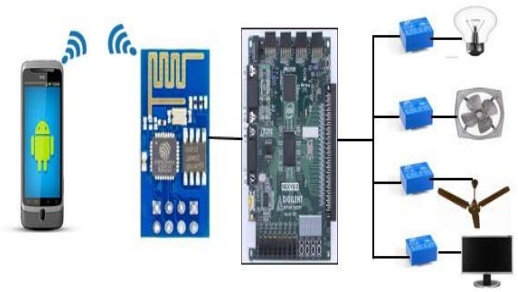

Fig. 3: Block diagram

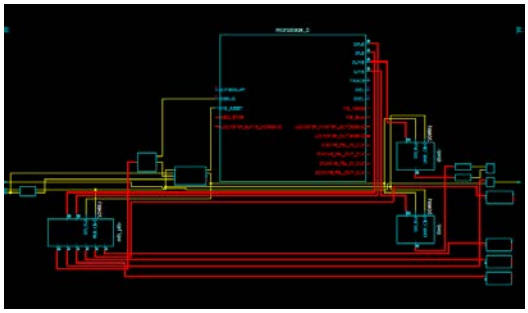

Fig. 4: XPS generated block diagram

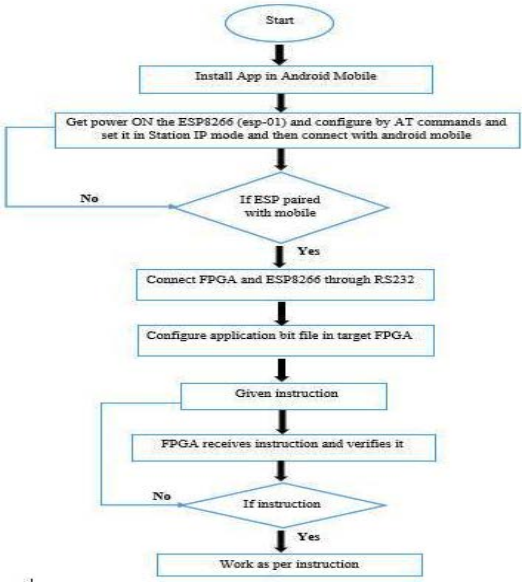

Fig. 5: Flow chart

and afterward it transmits those charges to RS-232. Taking into account orders got through ESP8266, FPGA controls the appliances through relays. This FPGA board comprises of the bit file downloaded into it by the designer. It comprises of all the instructions that are utilized to control the home appliances.

Algorithm: The overall procedure is shown in Fig. 5. At first to begin our procedure we have to introduce the application that we created in our android mobile and afterward switch ON the Wi-Fi in our mobile, ensure that the Wi-Fi is in ON condition or not. On the off chance that it is not ON then turn it ON. Once the Wi-Fi is turned ON give control supply to the ESP8266 IOT module and pair it with the android mobile by means of Wi-Fi. Typing so as to blend between these gadgets is finished the secured Static IP and port number given and created by AT Commands. Ensure that the module is given 3.3 V if not the module might crumple. On the off chance that the IOT module is combined then interface the FPGA and ESP8266 through RS 232. This RS 232 is a serial port gadget that exchanges every one of the directions given to it in serial way. Presently, design application bit file in target FPGA and afterward give the guideline. This direction given to control the appliances that are to be controlled. The software bolsters any kind of guideline, i.e, they can either be Upper characters or Lower character and numbers. Once, if a guideline is given in the application, FPGA gets the direction by means of serial port and after that it checks whether the given guideline has substantial address or not. On the off chance that the direction is substantial then the corresponding appliances works and on the off chance that it is not legitimate, the system will rehash the procedure until a legitimate guideline is given.

RESULTS AND DISCUSSION

Simulation results: Resources are utilized for IOT communication between FPGA and ESP8266 module. Both are communicated through UART (Universal Asynchronous Receiver and Transmitter). The results of transmitter and receiver communication between FPGA and ESP8266 module through several commands that are shown in Fig 6-9. The RTL Schematic of UART Transmitter is shown in Fig. 10.

Table 1 discuss about device utilization summary. Figure 11 shows the graphical representation of device utilization summary or resources utilized for FPGA IOT communication. Figure 12 shows the loading of firmware in to the ESP8266 module and Fig. 13 shows the interface of IOT module with FPGA. Figure 14 shows the Android App to control the home appliances.

Figure 15 shows a relay circuit to connect FPGA and supply mains when a command is received through ESP8266 module through serial communication the relay is activated and home appliances are controlled through relays. From Fig. 16-19, we get to know that one, two,

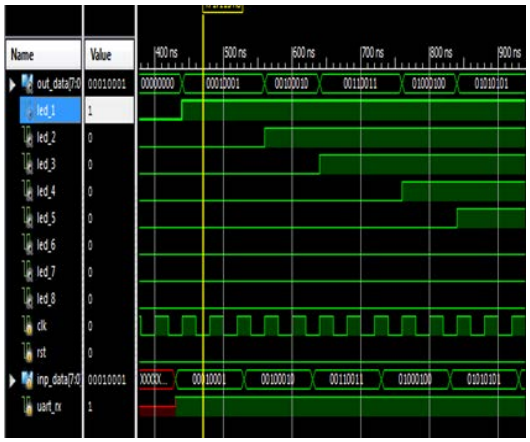

Fig. 6: Receiver results of UART

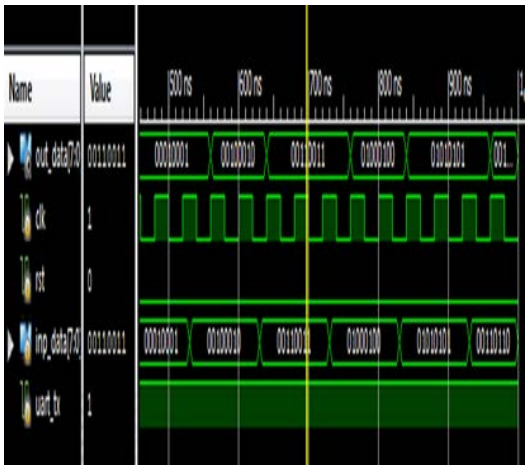

Fig. 7: Transmitter results of UART

Table 1: DUS (Device Utilization Summary) of IOT module

Device utilization summary

Logic utilization	Used	Available	Utilization (%)
Number of slice flip flops	1,475	9,312	15
Number of 4 input LUTs	2,447	9,312	26
Number of occupied slices	1,761	4,656	37
Number of Slices containing only related logic	1,761	1,761	100
Number of slices containing unrelated logic	0	1,761	0
Total number of 4 input LUTs	2,506	9,312	26
Number used as logic	2,032		
Number used as a route-thru	59		
Number used for dual port RAMs	256		
Number used as shift registers	159		
Number of bonded IOBs	35	232	15
IOB flip flops	22		
Number of RAMB16s	16	20	80
Number of BUFGMUXs	2	24	8
Number of DCMs	1	4	25
Number of BSCANs	1	1	100
Number of MULT18X18SIOs	3	20	15

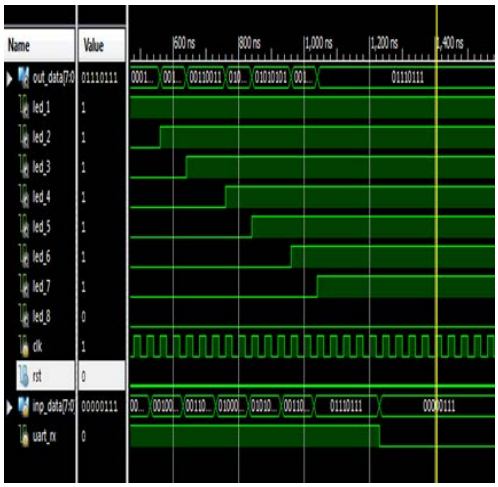

Fig. 8: Receiver results of UART

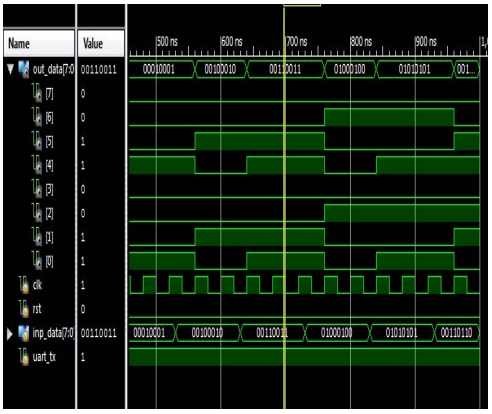

Fig. 9: Transmitter results of UART

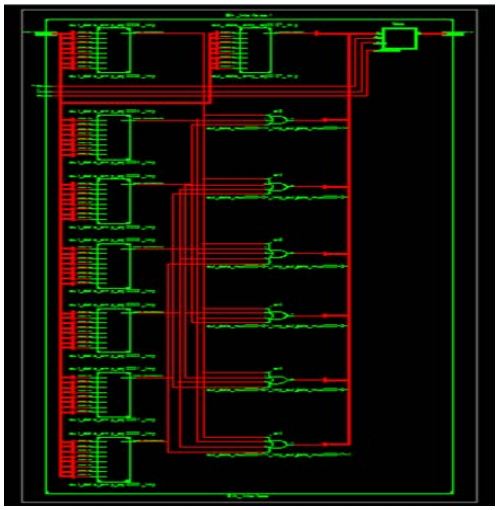

Fig. 10: Transmitter results of UART

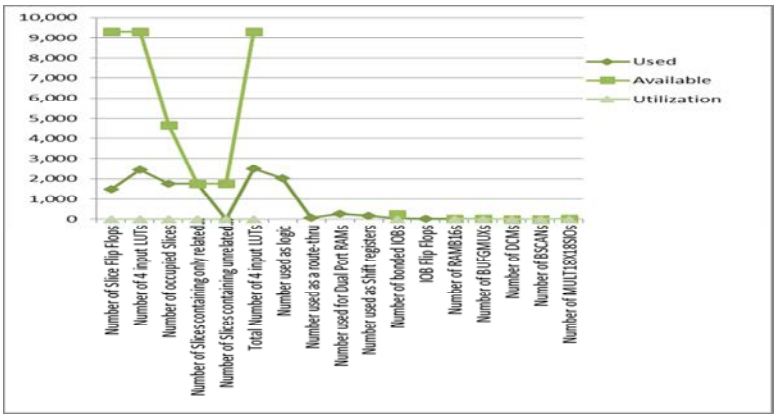

Fig. 11: Graphical representation of DUS

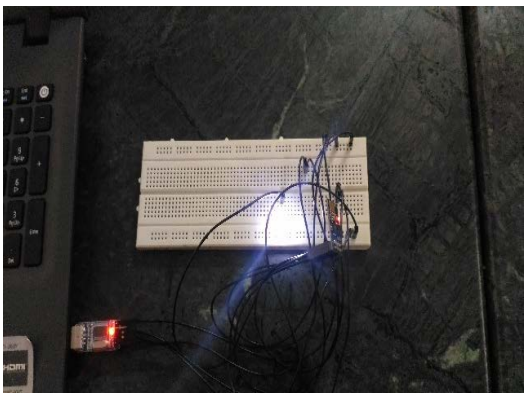

Fig. 12: IOT firmware loading through USB-TTL

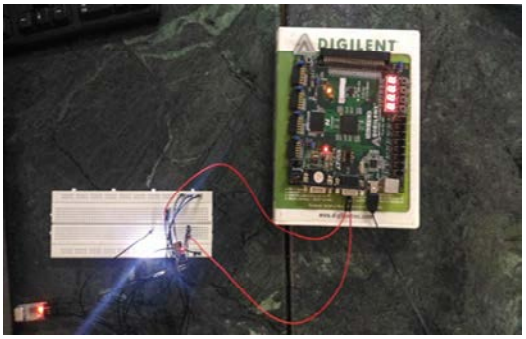

Fig. 13: Interfacing ESP-01 IOT module to FPGA

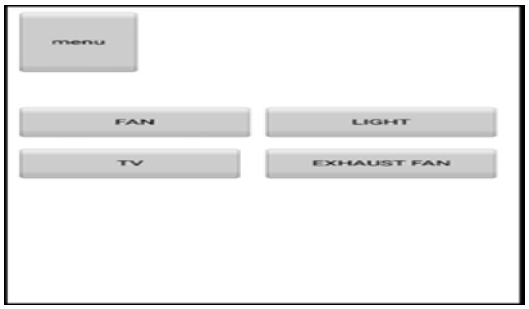

Fig. 14: Android App to control home appliances

Fig. 15: A relay circuit connected to FPGA to control Appliances

Fig. 16: When one appliance is controlled

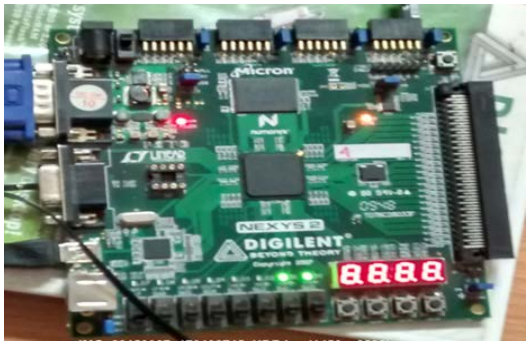

Fig. 17: When two appliances are controlled

Fig. 18: When three appliances are controlled

Fig. 19: When four appliance are controlled

three home appliances respectively are controlled through FPGA by some commands for each home appliances (such as Light, Fan, TV, etc.)

- UART transmitter and receiver results
- FPGA OUTPUTS

CONCLUSION

Gigantic development in innovation and headways in remote correspondence, savvy method for living has ended up being a noteworthy part in the present period of human life. We have proposed a brilliant home automation which is worked with the assistance of android advanced mobile phone by utilizing IOT technology and home automation will be controlled through FPGA through serial communication. Work is actualized progressively and appliances are controlled by commands from android mobile.

REFERENCES

Atzori, L., A. Iera and G. Morabito, 2010. The internet of things: A survey. Comput. Networks, 54: 2787-2805.

Han, I., H.S. Park, Y.K. Jeong and K.R. Park, 2006. An integrated home server for communication, broadcast reception and home automation. Proceedings of the 2006 International Conference on Consumer Electronics, Digest of Technical Papers, January 7-11, 2006, IEEE, Daejeon, South Korea, ISBN:0-7803-9459-3, pp: 253-254.

Kasai, H., 2013. Embedded software development kit for sustainable distributed mobile Geocast. Proceedings of the 2013 IEEE International Conference on Consumer Electronics (ICCE), January 11-14, 2013, IEEE, Tokyo, Japan, ISBN:978-1-4673-1361-2, pp: 524-525.

Li, S., J. Li, X. Nie and L. Kong, 2015. Design and implementation of smart home based on android. Proceeding of the 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS), August 21-23, 2015, IEEE, Harbin, China, ISBN:978-1-4673-7572-6, pp: 32-35.

Rudramma, K.R. and B.M. Krihna, 2013. PS2 VGA peripheral based arithmetic application using micro blaze processor. Intl. J. Emerging Trends Technol. Comput. Sci., 21: 153-157.

Viejo, J., M.J. Bellido, A. Millan, E. Ostua and J. Juan *et al*., 2006. Efficient design and implementation on FPGA of a MicroBlaze peripheral for processing direct electrical networks measurements. Proceedings of the 2006 International Symposium on Industrial Embedded Systems, October 18-20, 2006, IEEE, New York, USA., pp: 1-7.

Yan, M. and H. Shi, 2013. Smart living using Bluetooth-based android smartphone. Int. J. Wireless Mobile Networks, 5: 65-72.