122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society

Paper ID #14039

Work-in-Progress:Experience of Teaching Internet-of-Things Using TI ARM based Connected Launchpad

Dr. Nannan He, Minnesota State University, Mankato

Nannan He is an Assistant Professor of the ECET Department in Minnesota State University at Mankato. She received the Ph.D. in computer engineering from Virginia Tech. Her teaching and research interests are in safety-critical embedded software, real-time systems, and software verification.

Dr. Han-Way Huang, Minnesota State University, Mankato
Miss Navya Mereddy

Work-in-Progress: Experience of Teaching Internet-of-Things Using TI ARM based Connected Launchpad

Abstract

Most embedded systems design efforts in industry have moved to modern 32-bit devices with networking capabilities. The widespread development of such embedded applications has led to the highly influential concept of the "Internet of Things (IoT)". This paper describes our ongoing work of introducing the IoT technologies in an embedded networking course. We mainly present our experience of applying the TM4C1294 Connected Launchpad with two BoosterPacks - Sensor Hub and SimpleLink WiFi provided by Texas Instruments (TI) to the development of course projects. The Launchpad includes a 32-bit ARM Cortex M4 microcontroller (MCU) integrated with 10/100 Ethernet MAC and PHY. TI's Code Composer Studio (CCS) – an Eclipse-based Integrated Development Environment (IDE) and some open source software like the lightweight TCP/IP stack called IwIP are used for software development. In this course, we introduce students the TCP/IP protocols, and wireless communication technologies like WiFi and Bluetooth. Through course projects, students study the implementation of TCP/IP protocols in practice, and learn how to use drivers of Ethernet port and WiFi connection to develop microcontroller-based networking applications. Our primary experiences indicate that TI's connected Launchpad with various BoosterPack plugin-in modules is a flexible, cost-effective, well-supported platform for teaching embedded networking and developing IoT applications.

1. Introduction

The recent industry survey shows that around 60% of new embedded designs include networking capabilities [1]. As IPv4 is giving way to IPv6, every device on the Internet can have its own unique IP address. The number of connected devices on the Internet is predicted to surpass 50 billion by 2020. Such growing interconnected web of devices is often referred to IoT. These devices vary from personal electronics to industrial machines and sensors. The majority of devices in the IoT contain 32-bit MCUs with large flash memory and the wide range of standard I/O interfaces including networking. As IoT grows rapidly in industry, there is a great need to educate students IoT so that they are capable of developing basic IoT applications.

This new course covers the fundamental IoT technologies relevant topics, such as the evolution of IoT, advanced sensors, TCP/IP protocols, prominent wireless technologies for IoT, and the practical issues of design and development of IoT applications based on MCUs. The majority of devices in IoT are getting wirelessly connected to the Internet. Thus, four wireless technologies and their applications in IoT – WiFi, Bluetooth, ZigBee and 6LoWPAN are taught in detail. To deepen students' understanding of these wireless technologies, the comparison of key technical features and the selection of the suitable wireless technology in applications are discussed through the case study of an example IoT in the smart grid application. Some electronic principles of common sensors in IoT are also introduced in this course, such as gyro and accelerometer sensors for the motion tracking, temperature, humidity and light sensors for the environment monitoring. Through projects, students learn how to program MCU to access these sensors via Analog to Digital converter and interpret the readings. The emphasis of this course is on teaching students to develop the MCU-based IoT projects, which involve the web development, sensors reading, embedded processing, and network connection via Ethernet port or WiFi. This course is for the electrical and computer engineering students at the senior or graduate level. The project-based learning is the main teaching approach.

The TI TM4C1294 Connected Launchpad (i.e., EK-TMC1294XL) is the main evaluation board for practicing the creation of IoT applications. This board features a 120MHz ARM Cortex-M4 microprocessor with 1MB nonvolatile flash memory, on-chip 10/100 ENET MAC and PHY for the Ethernet connection, 8 motion control pulse-width modulation (PWM) channels and a multitude of simultaneous serial connectivity. In the course projects, students learn to utilize these features of MCU to develop the basic building blocks for an IoT application. For instance, using serial inter-integrated circuit (I^2C) interface and serial peripheral interface (SPI) to configure the sensors on the sensor Hub BoosterPack and read the sensing data back to the Launchpad. This Launchpad has two stackable BoosterPack connection sites which offer the flexible hardware extensions.

Besides the rich hardware features for networking and the machine to machine communication, a set of free software library packages like the EK-TM4C1294XL TivaWare and utility software is available for facilitating programming this board. Moreover, TI provides nearly thirty example projects for this board to demonstrate the development of from fundamental applications, such as programming interrupts, timers, general purpose I/O to more advanced applications like accessing universal serial bus (usb), connecting to the Internet via Ethernet port or WiFi. These examples are the basis to help us design course projects. Students have shown great interests in these new course topics and are capable of developing IoT relevant capstone projects for home automation by the end of the course.

This paper presents our ongoing work of teaching advanced IoT technologies to electrical and computer engineering students, with the emphasis of how we develop the lab projects by using TI's latest Connected Launchpad EK-TMC1294XL and its associated software packages. An overview of IoT technologies including its evolution is first briefly introduced. Then, the course description including the learning outcomes and lecture and lab contents is given, especially the organization of lab projects and their learning usage. Then, we present the future work of integrating real-time operating systems (RTOS) into the IoT application development. Finally, the paper is ended with conclusions.

2. Overview of the Internet of Things (IoT)

The Internet of Things (IoT) is generally thought of the interconnection of uniquely identifiable embedded computing devices within the Internet infrastructure to provide remote monitoring or control services. The IoT is grounded in the recent advances in microelectronics, communications and information technology. Due to their decreasing size, falling prize and declining power consumption, microprocessors, communication modules and other electronic components are being increasingly integrated into one device. Things, in the IoT, refer to a wide range of such "smart" devices being "computerized" and equipped with network interfaces. These devices typically have not only basic sensory and actuator capabilities to perceive and act on the environment, but also have built-in networking capabilities to communicate with each other, access Internet services and interact with people [2]. Analysts estimate that more than 50 billion devices will wirelessly connected for the IoT by 2020 [3]. The IoT is expected to transform how we live, work and play by bringing remote automation to every facet of our lives, while also deliver advanced industrial applications like a smart grid to enable more connectivity throughout consumers, power grid infrastructure and utility providers.

Origin and evolution of the IoT

Mark Weiser's 1991 seminal paper on ubiquitous computing, "The Computer of the 21st Century" and academic convention such as the annual ACM Ubicomp conference on ubiquitous computing produced the contemporary vision of the IoT [4]. The concept of "IoT" became first popular by the work of the radio-frequency identification (RFID) research center called Auto-ID at Massachusetts Institute of Technology (MIT), which started to develop a cross-company RFID infrastructure in 1999. Kevin Ashton, the cofounder of Auto-ID, is known for inventing the term IoT in the same year [5]. And He was quoted in an article entitled "The Internet of Things" in Forbes Magazine as saying "We need an internet of things, a standardized way for computers to understand the real world". This was first documented used of the term in a literal sense [6].

The IoT is constantly evolving and the industry of IoT is still in its infancy. Many technological challenges are faced in industry. For instances, embedded processors and MCUs in the IoT need to address the wide spectrum of applications from controlling a simple sensor node to providing the high performance real-time embed processing functions; and various wired and wireless connectivity technologies should meet the needs in different IoT applications; and a large selection of sensors and power-management technologies should provide the friendly user interface to the IoT and energy-saving designs. Technically speaking, the integration with the Internet implies that devices will utilize an IP address as a unique identifier. They need IPv6 to accommodate the extremely large address space required, due to the limited address space of IPv4. To a large extend, the global adoption of IPv6 in the coming years will be critical for the successful development of the IoT.

Building blocks of the IoT

Figure 1 illustrates the engineering overview of the IoT. Three fundamental building blocks of the IoT are the sensing, embedded processing and connectivity. The IoT enabling technologies are grouped under these three building blocks. At the same time, the wide range of potential IoT applications needs a software development environment, which ties together the applications, sensing, users command, control and routing processing and the security of each IoT device and the entire system [7]. As the importance of the software in the MCU based IoT has increased recently, more software and utility tools are needed. Figure 1 also lists some example IoT applications.

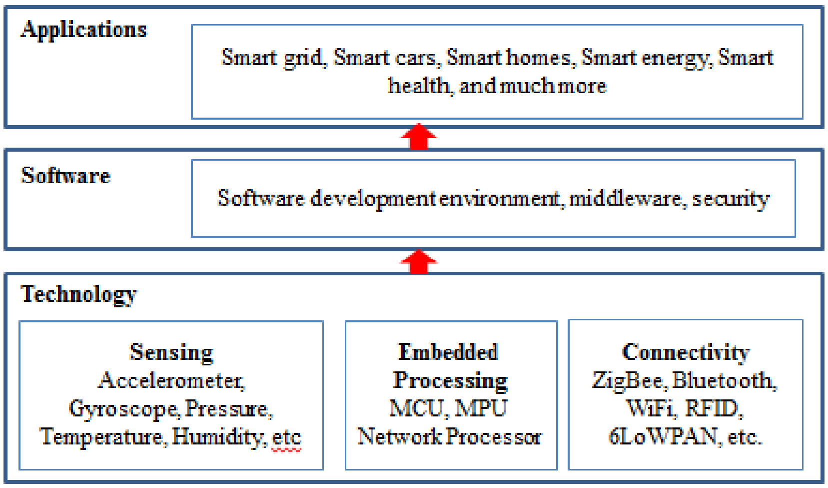

Figure 1. Engineering overview of the IoT

In the rapidly growing IoT, various smart devices from simple sensor nodes to more complex embedded processing nodes are getting wirelessly connected to the Internet. We briefly review four popular wireless connectivity technologies that we have covered in the course. More detailed discussions can be found in a TI's white paper [8].

<u>Wi-Fi</u>

Wi-Fi technology, based on the IEEE 802.11 standard, was initiated as a wireless replacement for the wired IEEE 802.3 Ethernet standard. It is integrated into all new laptops, smartphones and tablets. Wi-Fi has become so ubiquitous that people often refer to it as "wireless". Wi-Fi Access Pointers are deployed in most homes and enterprise. Wi-Fi and TCP/IP software are fairly large and complex. Adding Wi-Fi connectivity to devices with little processing power was almost infeasible before. However, new silicon devices in which the Wi-Fi software and the TCP/IP software is embed has come out. This increasing level of integration in these devices enables wireless Internet connectivity with most MCUs and also eliminates all required radio design experience. Wi-Fi radios generally have fairly large power consumption. With the advanced sleep protocols, the average power consumption of Wi-Fi radios has been reduced dramatically.

In summary, Wi-Fi is the most ubiquitous wireless technology. Latest advances have reduced the barriers of high power and complexity for applying WiFi into emerging IoT applications and battery-operated devices.

Bluetooth

Bluetooth technology is a Personal Area Network (PAN) technology mainly used as a cable replacement for short-distance communication. It became very popular in mobile phones. It is related to the IoT mainly because it supports connecting wireless accessories the last 10 meters to a smartphone or tablet. Bluetooth Low Energy (BLE) is a recent addition to the Bluetooth specification, which is designed for lower data throughput. This addition can significantly reduce the power consumption of Bluetooth devices and enables years of operation using coin cell batteries. BLE has enabled various new applications spanning smart health, fitness, and industrial spaces.

ZigBee

ZigBee is a low-throughput, low-power and low-cost wireless technology. Similar to Bluetooth, it also has capability to maintain long sleep intervals and low operation duty cycles to be powered by coin cell batteries. Moreover, recent ZigBee devices can even allow energy harvesting techniques for battery-less operation. As the IP specification in the ZigBee standard has not got much attention in the industry, ZigBee networks often requires an application-level gateway to connect to the IoT.

6LoWPAN

6LoWPAN is the first wireless connectivity standard that was created for the IoT, typically used to form Local Area Networks (LANs). Its goal is to apply IP to the smallest, lowest-power and most limited processing power device. IPv6 was chosen as the only supported IP in 6LoWPAN as it supports a much larger addressing space, hence much larger networks compared with IPv4. This technology is new to the market. Its main applications are positioned in the rapidly increasing market of internet-connected sensors and other low data throughput and battery-powered applications.

Recently, ARM, a famous British semiconductor company, has collaborated with leading embedded and cloud technology companies to develop the *mbed* IoT device platform [9]. This platform aims at providing basic ingredients to enable the creation and deployment of standard-based IoT devices based on 32-bit ARM Cortex-M microcontrollers. Major ingredients include an operating system, device server, software tools and ecosystem. The mbed demo-board is a NXP MCU - mbed NXP LPC17680. The software development tool for mbed is the cloud-based C/C++ compiler. It can run under any PC platform with a web browser, like Firefox. The mbed had been adopted for teaching the IoT in universities [10]. Table 1 shows the major features comparison between mbed and TI's connected Launchpad employed in this course. The TI connected Launchpad has relatively higher computing power.

	ARM mbed demo-board	**TI connected Launchpad**
MCU	32-bit 96-MHz ARM Cortex M3	32-bit 120-MHz ARM Cortex-M4
Memory	64KB RAM, 512KB Flash,	1MB Flash, 256 KB SRAM, 6KB EEPROM
Peripheral interfaces	Ethernet, USB, CAN, SPI, I2C	Ethernet, UART, I2C, SPI, CAN, USB
Software development tool	Cloud-based IDE	Eclipse-based IDE – Code Composer Studio, TivaWare

Table 1. Engineering overview of the IoT

3. Course description

The ultimate goal of this ongoing teaching practice is to equip students with the knowledge of advanced IoT technologies and developing basic MCU-based IoT applications. Several research graduate students helped prepare the course materials, especially projects. This course is organized as three hours of lecture and two hours of laboratory per week. It has three major objectives.

- To improve students' awareness of origin, current status and future directions of the IoT.
- To introduce students advanced technologies that enable the emerging IoT.
- To teach student to be capable of developing the basic MCU based IoT applications.

Course learning outcomes

In the preparation of this course, we derive the following course learning outcomes under the above three major objectives.

1. To demonstrate the knowledge of the evolution of the IoT.
   1.1 To understand the origin and current status of the IoT in industry and academy
   1.2 To understand the major technology challenges for the promise of the IoT
2. To demonstrate the knowledge of the fundamental IoT technologies and software
   2.1 To get familiar with the common sensor techniques in the IoT. (i.e., accelerometer, temperature, and humidity, etc.)
   2.2 To comprehend the communication protocols related to the IoT networking connectivity. (i.e., HTTP, TCP, IPv4, IPv6, and WiFi, etc.)
   2.3 To understand major wireless technologies for the IoT. (i.e. Wi-Fi, Bluetooth, ZigBee, and 6LoWPAN, etc.)
   2.4 To get familiar with software tools and libraries for programming the MCU-based embedded processing nodes and sensing nodes for IoT applications.

3. To demonstrate the capability of developing MCU-based IoT applications
   3.1 To utilize 32-bit ARM MCUs, sensors and wired connection to develop basic IoT applications.
   3.2 To utilize 32-bit ARM MCUs, sensors and wireless connection (i.e., WiFi) to develop basic IoT applications.

Course contents

This engineering overview in Figure 1 guided the organization of teaching materials for this new course. In the course, we aim at covering each technological building block in lectures, and introducing necessary software tools for students to practice the IoT applications development through course projects. Course materials were drawn from textbooks [11, 12], industrial white papers, TivaWare software, workshop [13, 14] and example projects for the TI Connected Launchpad. Table 2 lists all topics covered in this course.

IoT Introduction
1. Origin and evolution of the IoT
2. Multidiscipline technology challenges of the IoT development
Building blocks of IoT Technology
3. Basic electronic principles of common sensor technologies in IoT
4. Reading sensors data via programing 32-bit MCUs.
5. Application layer protocols (DHCP, WWW and HTTP)
6. Transport layer protocols (TCP and UDP)
7. Network Layer protocols (IPv4 and IPv6)
8. Data link and physical layer protocols (IEEE 802.11)
9. Comparison of wireless connectivity technologies for the IoT
10. Wi-Fi wireless connectivity
11. Case study of smart grid as an example IoT application
Development of basic IoT application as course projects
12. Generation of HTTP or CGI/SSI requests from Webpage
13. Development of MCU-based Ethernet application using open source software IwIP
14. Development of device to device communication via Wi-Fi connectivity

Table 2. Course topics

In this course, each network protocol covered in the topic 5-9 is introduced from the practitioner pointer of view in about one week. Besides the basic concepts like standards, message formats, addressing, we also aim at introducing students the software implementation and application issue of a protocol. An Ethernet example project *enet_io* provided by TI for the Connected Launchpad is applied for achieving this teaching goal. This project demonstrates the web-based I/O control using the Tiva Ethernet Controller and the open source LwIP TCP/IP stack. As an example, while we introduce students the host configuration protocol DHCP in the application layer, we also refer to its implementation in the project to study how DHCP is used to obtain an Ethernet address, and discuss the practical issues in the development, such as how to address the scenario if DHCP times out without obtaining an address.

ID	Project description	Learning usage
1	- Blink on-board LED at fixed rate. - Display messages on a terminal application via serial UART interface	- Create, build and debug projects for the Connected Launchpad in TI Code Composer Studio; - Get familiar with TivaWare software - Program GPIO of ARM Cortex-M4 microprocessor.
2	- Generate the PWM waveform to control a servo motor of a fan using the on-chip PWM block.	Study the TivaWare library for PWM - Use API functions to initialize, configure (duty cycle and frequency) and enable the PWM module.
3	- Read data from the sensors on the SensorHub BoosterPack connected to the Launchpad	Study the TivaWare Sensor Library - Use API functions to configure the I2C bus, assemble the sensing data read from the I2C bus.
4	- Set up the HTTP server and CGI /SSI handlers from the MCU's flash memory. - Use JavaScript running on the web browser to send HTTP request of toggling an on-board LED and return the board response to the browser. - Use HTML forms to pass a LED control command to CGI handlers and send a response to the browser after processing the command.	1. Study the implementation of network protocols using open source lwIP TCP/IP stack. 2. Study to use lwIP for setting up the Ethernet application. - Use API functions ControlCGIHandler() and CSSIHandler() to accept, process user commands from the web browser. - Program javascript to manage new HTTP requests and responses.
5	- Set up two C3100 simpleLink WiFi BoosterPacks Launchpad on two Connected Launchpads respectively. One is in WLAN-Station mode; another is in the client mode. - Connect them via a Wi-Fi AP and UDP socket.	1. Study the implementation of Wi-Fi protocol via the hardware/software platform C3100. 2. Use the development kit for this Wi-Fi device. 3. Use the host driver to develop this kit.

Table 3. Lab projects

Lab projects and capstone project with TI's connected Launchpad

Table 3 summaries the major lab projects in this course and their learning usage. In our department, the ARM Cortex-M4 MCU has been taught for electrical and computer engineering in the microprocessor course for two years. Students who take this embedded network course are assumed to have the basic knowledge of programming ARM Cortex-M4. But they might have used other development boards other than TI's Launchpad. To ease the immigration to this Launchpad, we first prepared three basic projects on the programming Connected Launchpad.

Students could get familiar with the development environment Code Composer Studio. They also learn to utilize the TivaWare software package to program the MCU. TivaWare is the royalty-free software designed to simplify the development of TI ARM Cortex-M series MCU applications. It includes the peripheral driver library for controlling the peripherals; and the Sensor library with an I2C master driver for communicating over the I2C serial bus with the sensors on the SensorHub BoosterPack. Moreover, these three blocks are the building blocks that could be integrated to construct the IoT applications.

The 4th lab project in the table has the rich features and demonstrates a basic IoT application with wired connectivity. The capstone project developed in this course uses this project as the starting point. We derive multiple lab exercises from this project for different learning purposes. Firstly, as lwIP TCP/IP stack is utilized for Ethernet connection, students could analyze its source code to examine how the network protocols covered in the lecture can be implemented in the software. Secondly, as the HTTP server and CGI (Common Gateway Interface) handler are set up in the Flash of this Launchpad in this project, students could study how to edit the html web page, program the Javascript or Server Side Includes (SSI) script to pass the users requests from the web browser to the server and then dynamically insert or update the server response on the web page. Thirdly, students could learn how to write C code to process the commands in the server side, especially the usage of two handler functions *ControlCGIHandler*() and *SSIHandler()* for dealing with CGI/SSI requests.

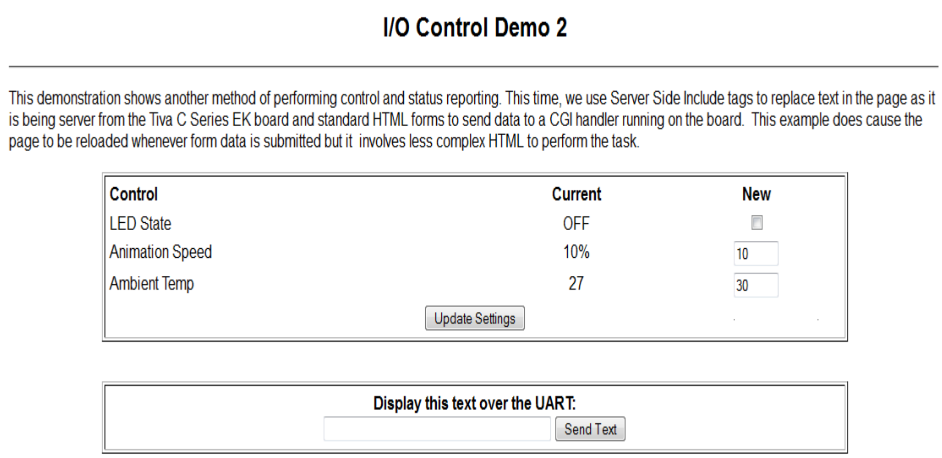

Figure 2. Control page of the Capstone project

Capstone project

This project is an IoT application for room temperature and humidity monitoring and control [15]. It makes use of the SensorHub to monitor the temperature and humidity. A server is set up in the memory of the Launchpad. With DHCP, the router will assign a unique IP to this launcpad. These sensing data will be displayed on a webpage associated this this IP via the Ethernet port of the Launchpad. User could set a temperature threshold on the web page. An LED and PWM generator will be enabled if the ambient temperature is bigger than the threshold. A servo motor controlled fan will start automatically. As the webpage can be accessed in any platform and physically any location as long as a web browser is available, users could remotely access this IoT application.

4. Future work

The reported work is still in progress. We present the two main directions that we will put more efforts. Firstly, the lab project on the wireless connectivity with Wi-Fi is primary. It is a machine to machine communication via a WiFi AP. We will improve this project as the capstone project to set up a HTTP server. We will also apply other wireless technologies like ZigBee as the wireless connectivity and develop more capstone projects. Secondly, we will employ the TI-RTOS (Real-time operating system) and TI Network Development Kit into the development of network applications. The industry survey indicates that around 70% of current embedded system designs have real-time capability to improve the performance and make the system more deterministic.

5. Conclusions

This paper presents our ongoing work of teaching engineering and technology students the IoT technologies in an embedded networking course. The future work directions are also discussed. The TI's low-cost connected LaunchPad, two BoosterPacks for sensing and Wi-Fi, Tivaware software library are employed in this course to develop basic IoT applications. Our present experiences indicate that TI's connected Launchpad with various BoosterPack plugin-in modules is a flexible, cost-effective, well-supported platform for teaching embedded networking and developing IoT applications.

Reference

1. David Blaza. "2013 Embedded Market Study", Design West conference, San, Jose, CA. 2013. http://www.slideshare.net/StephanCadene/embedded-mar1913
2. Mattern, Freidemann, Christian Floerkemeier. "From the Internet of Computers to the Internet of Things", Informatik-Spektrum 33 (2), pp 107-121. 2010.
3. Cisco Internet Business Solutions Group (IBSG). "The Internet of Things", April, 2011. https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.
4. Weiser, Mark. "The Computer for the 21st Century". Journal of Scientific American 265 (3), pp 94-104, 1991.
5. Schoenberger, C. R. "Internet of Things", Forbes Magazine, March, 2002.
6. Kevin Ashton, "That 'Internet of Things' Thing", RFID Journal, June, 2009.

7. Kaivan Karimi, Gary Atkinson. “What the Internet of Things (IoT) Needs to Become a Reality”, Freescale white paper, 2013. http://www.freescale.com/files/32bit/doc/white_paper/INTOTHNGSWP.pdf.
8. Gil Reiter. “Wireless connectivity for the Internet of Things”, Texas Instruments white paper, 2014. http://www.ti.com/lit/pdf/slay028.
9. C. Styles, “mbed educational program,” Arm Holdings, Cambridge, U.K., Mar. 2012 [Online]. Available: http://mbed.org/handbook/Education.
10. J.O.Hamblen, and G.M Van Bekkum, “An Embedded Systems Laboratory to Support Rapid Prototyping of Robotics and the Internet of Things”, IEEE Education, 2013. (06375774)
11. Behrouz Forouzan. “TCP/IP Protocol Suite, 4/e”, Mc-Graw Hill publisher, 2009.
12. Chwan-Hwa Wu and J. David Irwin. “Introduction to Computer Networks and Cybersecurity”, CRC Press Taylor&Francis Group publisher, 2013.
13. Jonathan W. Valvano, course website “how and why to teach wireless connectivity in embedded systems classes.” http://users.ece.utexas.edu/~valvano/arm/wirelessconnectivity.html
14. TI workshop. “Creating IoT Solutions with the TM4C1294XL Connected Launchpad Workshop”. http://processors.wiki.ti.com/index.php/Creating_IoT_Solutions_with_the_TM4C1294XL_Connected_LaunchPad_Workshop
15. Capstone project, http://jppnf1.wix.com/programmingtools#!wwwglobocombr/c20q4