Incorporation of the Internet of Things within the Introductory Course on Microcontrollers

Steve Rowland and Ramakrishnan Sundaram
Gannon University, Erie, PA, sundaram@gannon.edu

Abstract - **This paper discusses the restructure of the introductory course on microcontrollers using laboratory experiments which use the framework of the Internet of Things (IoT) to demonstrate the application of microcontrollers at the system and sub-system level. The course can be taught with compact and portable laboratory components thereby facilitating the on-line approach to instruction. This is especially necessary during the health pandemic which continues to disrupt face-to-face laboratory instruction. Specifically, the hands-on laboratory experiments and project-based experiences introduce the students to the collection of data using temperature and motion sensors, software programs for the microcontroller, and wireless communication between WiFi-enabled modules. The students observe and record the outcomes on personal computers and mobile devices. Rather than use the hardware and software tools from established vendors in the areas of IoT, we chose to design and assemble our own laboratory experiments and projects with simple, cost-effective, off-the-shelf components. The project activities focus on system design and integration based on the distinct laboratory experiences. The students are expected to possess basic knowledge of electrical circuits and electronics, programming skills in the higher level languages such as C/C++, and the fundamentals of test and measurement analysis.**

Index Terms – internet of things (IoT), microcontroller, sensors

INTRODUCTION

In recent years, the internet-of-things (IoT) [1]-[3] has burgeoned at astounding rates. The IoT describes the network of physical devices and machines, loosely labeled "Things" to the internet. These devices are embedded with sensors, software, and microcontrollers to connect and exchange data with the other devices and/or systems of devices over the internet. The devices can span the gamut of basic household appliances to complex industrial tools and machines. The ubiquitous nature of IoT as well as the emergent advanced technologies in wireless data communication and computing are rapidly altering the engineering landscape.

The engineering programs at universities across the world must adapt their courses and curricula to incorporate modern technologies in their courses, classrooms, and engineering laboratories. The engineering students who enroll at these universities expect to be educated and trained with the latest industry-approved tools and technologies to function effectively in the engineering industry.

The traditional approach to teaching the introductory course on microcontrollers takes the *bottom-up* approach i.e., first identify the architecture of the microcontroller (CPU, registers, memory, ALU) and lead up to the assembled or integrated device. Although this is essential for the detailed understanding of the operation of the device, it is not an effective method to train students to recognize the system level applications of the device. In the past, the project-based approach to hardware-software integration [4] and the field programmable gate array (FPGA) method to teach microcontrollers [5] have been implemented. The embedded systems approach to programming courses has also been developed [6]. In order to incorporate the *top-down* approach, this paper focuses on the introduction of microcontrollers at the system or device level with laboratory experiments and project activities relating to the design, assembly, and testing of systems which use microcontrollers, sensors, and wireless connections to user interfaces. Such systems are common in IoT-ready and IoT-enabled devices [7], [8]. This course is not intended to replace the traditional *bottom-up* approach, but to lay the groundwork for the study of the architecture in a future course.

The course is intended for the sophomore-level engineering student who recently completed a first course in programming languages such as C/C++, a course and laboratory on digital logic design, as well as the first course in electric circuits. A student with this background is ill-equipped to comprehend and pursue topics which examine microprocessor hardware at the bit, bus, register, and overall architectural level, and microprocessor software at the assembly level with value-adding knowledge, personal learning satisfaction, and educational success. Instead, a course conceived to expose the applications of microcontrollers to the assembly of simple to complex devices with IoT capabilities has more far-reaching ramifications. First, this approach allows the student to develop the skills at the system and sub-system level by recognizing and building the device required to meet specific objectives. Next, programming the microcontroller based on previous experiences with C/C++ is by far more effective as a teaching and learning tool than compelling the student to explore the bus architecture and assembly-level programming of the microcontroller.

Section 2 overviews the restructured course set up and compares the content of the traditional course with the restructured course. Section 3 provides details related to the delivery of laboratory experiments and project activities in the restructured course. Section 4 documents the learning outcomes assessment process. Conclusions appear in Section 5.

SECTION 2: COURSE SETUP

The course description of the past course on the introduction to microcontrollers was as follows:

This course is designed to give students a basic background in hardware and software aspects of microcontrollers. Contents of the course include core microprocessor architecture, instruction set architecture (ISA); microcontroller architecture, peripheral interfacing designing a C / Assembly software architecture.

In order to better prepare the student to understand the microcontroller at the component level, the syllabus of the introductory course has been revised to read as follows:

This course aims to introduce microcontrollers through laboratory experiments and projects designed to introduce the concepts of microcontrollers at the system and sub-system level. Students will gain hands-on experience assembling and testing devices which incorporate hardware such as sensors and actuators, and software to program the microcontrollers. Students will learn how to achieve node to client communication, node to node communication, and peer to cloud communication through laboratory experiments and projects based on microcontrollers.

The application of the microcontroller to build and test devices which use hardware and software components offers the student the opportunity to recognize the use of the microcontroller in a practical setting prior to the closer examination of the architecture and instruction sets. The latter topics are introduced and developed in a subsequent course.

The course outcomes (COs) of the prior course are as follows:
1. Demonstrate the understanding of the instruction-set and architecture of a microprocessor/microcontroller.
2. Develop an understanding of developing a software architecture for a microcontroller.
3. Develop an understanding of the basic tradeoffs of embedded microcontroller software solutions.
4. Develop an understanding of the challenges of integration between products.

The COs are revised for the restructured course as follows:
1. Demonstrate an understanding of devices with microcontrollers.
2. Demonstrate microcontroller programming skills.
3. Demonstrate how to program, analyze, and test devices with microcontrollers.
4. Understand the appropriate security/privacy solutions for devices with microcontrollers.

Broadly speaking, the restructured course enables the student to apply the microcontroller in specific applications as a black box before engaging in the detailed study of the black box in future courses on the architecture of the microcontroller.

SECTION 3: COURSE DELIVERY

Table I identifies the course schedule for the restructured course. The schedule comprises laboratory experiments suited for the chosen theme. The experiments require hardware assembly of the device using the appropriate components and software programming of the microcontroller for test and validation of device operation to meet the specifications. On the other hand, the traditional approach to the course content used discrete topics for discussion with few, if any, laboratory activity and can be seen in Table II. Each session lasts eighty minutes and there are two sessions in each week.

TABLE I
REVISED COURSE SCHEDULE

Theme	Laboratory	Session #(s)
Course overview: Microcontrollers and IoT	Learning the Arduino microcontroller	1-3
Microcontrollers and data communication: SPI, UART, UDP	Xbee wireless modules	4-6
	LoRa Packet Radio	7-8
	Bluetooth audio	9-10
	UDP transmitter and receiver	11-12
Term project #1	Microcontroller application with IoT	14-15
Microcontrollers and encrypted communication	Encrypted LoRa	16-17
Microcontrollers and WebServer	ESP and Wi-Fi	18-19
IoT data, microcontrollers, and the Cloud	ESP and sensor data on the Cloud	20-21
	Sensor data to Google Sheet	22-23
Final exam/project	Microcontroller-based IoT project	24-28

TABLE II
TRADITIONAL COURSE SCHEDULE

Theme	Topics
Course overview	Microprocessor vs. Microcontroller Review of the C language Introduction to the IDE and development board
Languages Text to machine	Machine and Assembly (ISA) code From Compiler to Machine code
Memory I/O ports	Memory Layout, addressing registers/peripherals, Digital IO Ports
Hardware/Software design	Software Architecture: API vs Applications/Design for reuse
Interrupts	I/O control, performance metrics
Timers	Watchdog, periodic timers, counters
Frequency	PWM, DMA, State machine-Mealy/Moore
Project	Microcontroller system design

The revised course schedule emphasizes the *top-down* approach while the traditional course schedule takes the *bottom-up* approach. The innovative practice is demonstrated by the rang of devices assembled through laboratory experiments and project activities. Samples are summarized below.

Sample of laboratory experiment

Figure 1 shows the wiring schematics for the laboratory experiment titled Server-Client Communication with LoRa Radio Modules. The set up includes the PIR motion sensor and the LoRa module interfaced to the Arduino microcontroller which are first assembled and tested as separate subsystems.

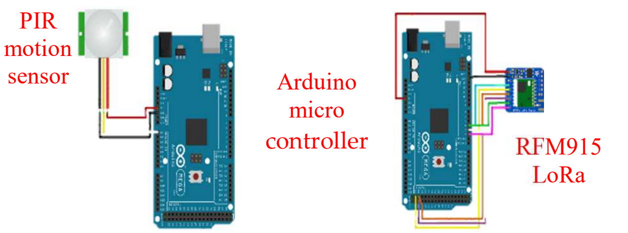

Figure 1: Wiring schematic of the subsystems

Thereafter, the server node and the client node are set up for data communication as shown in Figure 2.

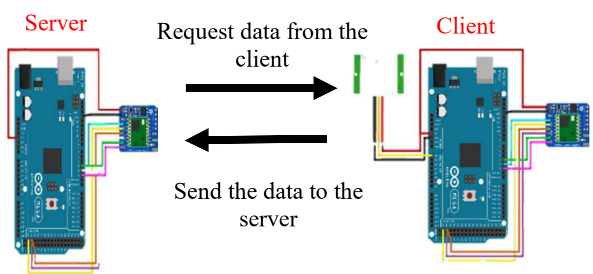

Figure 2: Server-Client setup

The experiment has three phases. The assembly phase leads to the final setup as seen in Figure 3.

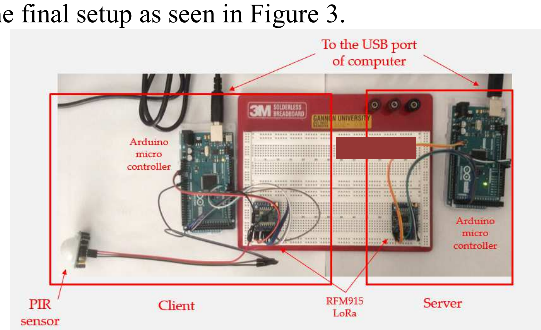

Figure 3: Assembly phase completed

The programming phase consists of writing, compiling, and uploading the C++ code to the Arduino microcontroller. The coiled and uploaded code for the client node is shown in Figure 4. The evidence collection phase comprises viewing and recording the results as seen on the serial monitor of the client in Figure 5.

Figure 4: Programming phase

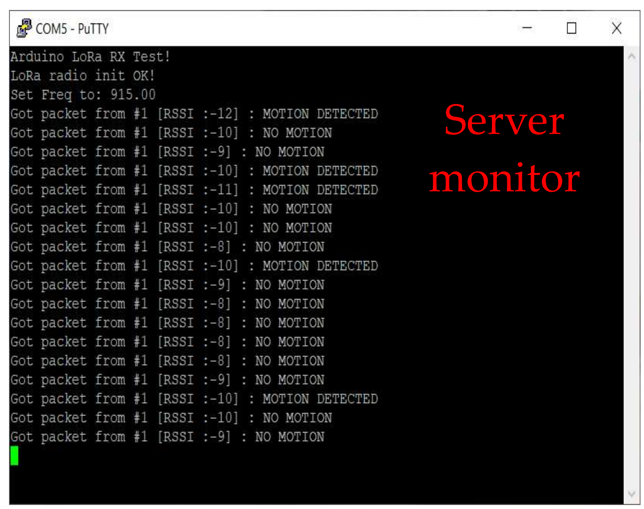

Figure 5: Evidence observation and collection phase

Sample of the project activities

One of the project activities comprised integration of subsystems to create a system for the detection and tracking of moving objects. The subsystems used sensors and bluetooth communication modules as shown in Figure 6. The assembled system to detect motion, track temperature changes, and transfer the data is shown in Figure 7.

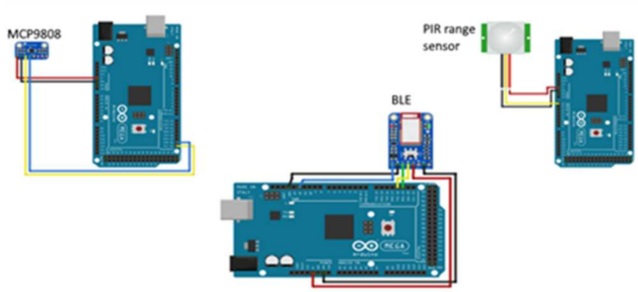

Figure 6: Subsystems of the project activities

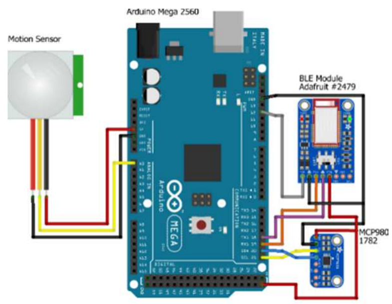

Figure 7: Assembled system for project activities

SECTION 4: LEARNING OUTCOMES ASSESSMENT

The learning outcomes assessment comprised the mapping of the course outcomes (CO) to the ABET student outcomes (SO) through performance indicators (PI). Specific course activities called ‘Key Assignments’ are used to measure the level of achievement of each student based the following scoring: Excellent (E) is scoring 90 or better of the total points possible, Adequate (A) is 75 or better, Minimal (M) is 60 or better, and Unsatisfactory (U) is anything below 60. For instance, the second CO is mapped as follows:

- **Demonstrate microcontroller programming skills (CO_2)**
  PI_1_4: *Select and implement the desirable solution and evaluate the results*
  Key Assignment: Lab experiment 2
  Justification: Lab experiment 2 requires the student to (a) setup node-client communication (b) transmit the state of a sensor from the node to the client. Lab experiment 2 measures the ability to select and implement the desirable solution and evaluate the results (PI_1_4), and the ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (SO_1).

Note: Course will be offered for the first time in the Fall 2021 semester, starting the end of August 2021.Hence there is no student and faculty survey data available during the review process in May-June 2021.

SECTION 5: CONCLUSIONS

The value of integrating the design and testing of IoT devices based on the use and understanding of microcontrollers at system and subsystem level at the sophomoric stage of the undergraduate engineering curriculum cannot be understated. Noting that the preparation of the student is in basic digital logic design and programming in high level languages such as C/C++, it seems prudent to take the *top-down* approach with microcontrollers i.e., what, and why they do it, before delving into *bottom-up* approach i.e., how they do it, in the future, specifically a more advanced course which adopts the schedule shown in Table II. The innovative aspect of the approach proposed here is the project-based integration of the application of concepts across topics from multiple electrical and computer engineering courses to introduce and reinforce the use of microcontrollers in the real word comprising the IoT environments. In addition, this course caters to students from the mechanical, biomedical, and industrial engineering disciplines so long as they understand and have experience with basic circuits and electronics, programming in C/C++, and digital logic design. Clearly, the course prepares the engineering student to recognize the need for subsystem and system design ahead of component or detailed architectural concerns.

REFERENCES

[1] N. Pazos et al., “ConnectOpen - automatic integration of IoT devices,” *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Milan, Italy, Dec. 14-Dec.16, 2015; DOI: 10.1109/WF-IoT.2015.7389129.

[2] L.V. Yaqiong et al., “IoT based Omni-Channel Logistics Service in Industry 4.0,” *Proceedings of the 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, Singapore, July 31-August 2, 2018; DOI: 10.1109/SOLI.2018.8476708.

[3] I. Aris et al., “Impacts of IoT and Big Data to Automotive Industry,” *Proceedings of the 2015 10th Asian Control Conference (ASCC)*, Kota Kinabalu, Malaysia, May 2015. DOI: 10.1109/ASCC.2015.7244878.

[4] J.L. Cruz-Rivera., “A project oriented approach to teaching software-hardware integration of microcontroller-based systems,” *Proceedings of the 28th Annual Frontiers in Education conference,* Nov. 1998. DOI: 10.1109/FIE.1998.736803.

[5] Karyono & A.Wicaksana , “Teaching microprocessor and microcontroller fundamental using FPGA,” *2013 Conference on New Media Studies (CoNMedia)*, Nov. 2013. DOI:10.1109/CoNMedia.2013.6708541.

[6] B. J. Mealy, “Embedded system-based introductory programming course for computer and electrical engineering students,” *Proceedings of the 38th Annual Frontiers in Education conference*, Oct. 2008. DOI: 10.1109/FIE.2008.4720620.

[7] A. Fidai et al., “Internet of Things (IoT) Instructional Devices in STEM Classrooms: Past, Present and Future Directions,” *Proceedings of the 2019 IEEE Frontiers in Education Conference (FIE)*, KY, Oct. 2019; DOI: 10.1109/FIE43999.2019.9028679.

[8] R Ani et al., “An Approach Towards Building an IoT Based Smart Classroom,” *Proceedings of the 2018 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, Sept. 2018;. DOI: 10.1109/ICACCI.2018.8554869.