FALL 2019

Integrating Open-Source Tools for Embedded Software Teaching: a Case Study

MUHAMMAD USMAN RAFIQUE
AQUIL MIRZA MOHAMMED
SHUAI LI
AMEER TAMOOR KHAN
Hong Kong Polytechnic University
China

AND

SEIFEDINE KADRY
Beirut Arab University
Lebanon

ABSTRACT

Consumer electronics are creating huge job markets for graduates with the programming back-ground, and more and more computer science departments are launching embedded software subjects to meet this demand. However, most students majoring computer science do not have the background in electronics, or even circuits. Due to this reason, how to convey the abstract knowledge on embedded software tangibly to students in computer science is a significant challenge in teaching to bridge the gap between the limited background and curriculum requirements. This paper addresses this challenge by proposing a student project to develop and pilot test a mobile robot platform, and a peer competition with the robots developed by other students. The project requires students to develop a mobile robot using commercial sensors and actuators for simultaneous light source seeking and obstacle avoidance. The development and assembly of such a system stimulate students' interest in this subject and provide them with a concrete understanding of embedded systems. The algorithm design and implementation enhance their theoretical knowledge gained from lectures. To facilitate the development, this project is partitioned into multiple parts and we provided a one-hour tutorial every week for all parts. The components include basic simple concepts such as input/output programming, intermediate tasks such as reading sensor data and controlling motors, and advanced topics such as multi-threading to simultaneously control multiple software methods. The final competition motivates students to develop robust and optimal programs. It helps them accumulate practical knowledge about performance limits of some electronic devices and calibration techniques to improve the performance. The pilot testing of this course has been conducted for two semesters and all major issues have been addressed. A comprehensive survey was conducted at the end of the course to get student feedback on different pedagogical aspects of the course. The evaluation results demonstrate the effectiveness of the project in enhancing students' understanding of electronics, knowledge of programming to support embedded systems, and critical thinking.

Key words: Design Competitions, Rapid Prototyping, Design-based Research.

INTRODUCTION

The recent rise of open-source and easy-to-use software and hardware tools has brought two drastic changes in learning of embedded systems in software engineering. First, the ease of use has enabled people with minimal background knowledge to quickly learn and start developing software. Second, there is the rise of powerful new tools that enable development of embedded systems with the minimal understanding of hardware. The recent success of Arduino hardware boards and programming language has added a new dimension in learning and solution development [1], [2]. Instead of long and arduous formal learning path, people are engaged in the activities of simple tutorials, discussion forums and explanatory videos on the internet. A key advantage of Arduino platform is the access to ready-made hardware: it enables people without hardware expertise to engage in embedded systems development easily. Other than learning, this new ecosystem of teaching and learning has successfully engaged a large body of developers who are actively imparting their acquired knowledge through electronic media. Naturally, there are some limitations and bottlenecks in the adoption of these quick and easy-to-use tools by self-learners: they can only achieve to a basic limit and scope of solutions is generally limited to amateur projects mainly done as a hobby. On the other hand, we have a formal learning path through advanced university courses which gradually prepare students to pursue professional careers in software engineering. In this paper, we propose to bridge the gap between formal learning methods and a growing community of easy open-source tools. Our initial testing suggests that by careful course design, hands-on tutorials and proper motivation and mentoring, we can harness the best of both worlds: students can be equipped with the formal knowledge to pursue professional careers while learning at a very fast rate. We have designed a graduate level course of embedded software engineering that includes community-based open-source tools for students without requiring prior expertise in hardware.

In addition to the knowledge-based (or assessment-based) teaching, we incorporate challenge based learning to equip students with more relevant and practical skills [3], [4], [5], [6], [7]. Since the goal of challenge based learning is to develop a solution in terms of financial and timing constraints, it familiarizes students with core practical issues and prepares them for professional careers. The central part of this method is to develop and pose a practical challenge to students which requires them to brainstorm ideas and then actually develop feasible solutions. In this course, the grand challenge is to program a small robot to reach the light source while avoiding obstacles, and other robots. Every group of up to three students has to assemble, and program the robot to accomplish the desired goal. To further motivate students, a robot contest is held at the end of the semester where all robots compete to reach the light source before others. Pilot testing of this approach for two years has revealed promising results and the majority of students have been observed to go beyond the normally expected efforts and results.

The field of embedded systems is an emerging field in the areas of information technology (like computer and communications applications), information systems and electrical and multimedia communications, and the like. Recently, the trend has embarked upon the integration of various techniques of embedded systems for robotics and computing devices. This is due to the fact of increased applications and advantages of these computing systems involving the real-time computation, power awareness, complex functionality and interdisciplinary handshake character-istics. These advantages lead to motivational and creative designs in the fields of engineering, like computer and communications engineering, robotics, multimedia and information sciences and systems.

Presently, the discipline of robotics engineering is considered as an interdisciplinary research with computers and embedded systems engineering. In general, the courses offered by the universities are broadly classified into one of the two types. One type is theoretical which deals with the fundamentals and automation of robotics kinematics and dynamics and control. The second type is practical or industry-related which focuses on the applications and operations of embedded accessories, such as using sensors, actuation, motors, microprocessors with support-ing and controlling mechanisms. The theoretical courses are mostly prescribed in mechanical engineering curriculum whereas technology-related is always a state-of-the-art topic in the field of electrical and electronics engineering.

Although this course is particularly designed to teach the students about embedder systems via robotics, but it doesn't mean that we cannot expand the contents of this course for other examples as well, including IoT, telecommunication, and signal processing in embedded systems. Since this was our test course to see how well the content of the course is organized and how beneficial this methodology is that is why we kept the content short and to the point. Above mentioned courses may or may not require some prerequisites for example, in IoT students need to have a sound knowledge related to different communication protocols same is the case with the telecommunication, which means that the content of the course needs to be more elaborative and requires more planning. On the other hand, robotics is easy because everything on wheels is a robot, pretty easy to build-up the structure over this foundation.

In this paper, we examine the pedagogical considerations of designing a graduate course in embedded software engineering. We detail the process and challenges of adopting challenge based approach by introducing a robot programming project in the course. Working on an actual hardware, including sensors and multiple software modules, formally introduces the problems and key knowledge domains of low-level embedded engineering. We examine the theoretical concepts of the course, with an emphasis on specially-designed tutorials, to assist students in accomplishing the course project. We have conducted the pilot testing of the course for two semesters: fall 2015 and fall 2016. Based on feedback from students, we learn that students find this course motivating, interesting and helpful in learning of core concepts of embedded software. Successful completion of the project by all students in fall 2016 semester highlights the engagement and positive contribution from students.

COURSE DETAILS

Introduction to Course

Embedded Software Engineering (COMP 5228) is a graduate level course offered by the Department of Computing at Hong Kong Polytechnic University. Major components of the course include the introduction to models of computation for representing behaviors for em-bedded applications, organizations and architectures, implementation of embedded systems with hardware/software codesign, and programming languages for embedded systems. The course also covers simulations, testing and verification of embedded systems. Advanced topics, such as assembly language, closed loop embedded systems, and multiprocessor communication. The course is taken by full-time as well as part-time students, primarily from the department of computing, but there is no restriction.

Formal teaching is subdivided into two parts: lectures with the focus on core concepts and hands-on tutorials focusing on practical aspects of the course. To fully engage the participants, a group project is assigned to students (up to three members) to make a robot. At the end of the semester, these robots contest in a mini-robot competition to stimulate students' intrinsic motivation of excellence. There is a progress presentation midway through the semester and a final presentation, just before the end-of-semester robot competition.

Student Heterogeneity

Since this course is open to full-time as well as part-time students, the class is generally comprised of a heterogeneous group of people from different backgrounds. Over last two years (fall 2015 and fall 2016), students from backgrounds in computer science, electronics, biomedical and mathematics have joined this course. Further, some students are enrolled in a program that requires thesis while others are pursuing purely coursework-based degrees. Since the course is offered in the evening, every year software engineers enroll in this course with the goal of expanding their expertise to capitalize on new job markets. Overall, the majority of students have no experience of working on an embedded systems before taking this course. In fall 2015, out of a class of 19 students, only one student had prior experience of working on any embedded system.

Prerequisites

Owing to diverse backgrounds of students, there is no official prerequisite for this course; any interested student (eligible to take advanced courses) can enroll. This implies that a certain balance should be maintained between breadth of topics covered and the technical depth and level of difficulty. For instructors, a key challenge is to make the course suitable for the majority of participants while galvanizing interest of students. That is why challenge based learning and an interesting robotic project have been adopted where students can physically see the results of their efforts.

In the first week, students are apprised about the course in detail: topics to be covered, required outcomes, project, assessment methodology, and expected learning outcomes. This allows students to judge whether this course matches their desired outcomes and learning goals and decide the suitability of course for them.

Learning Outcomes

After successfully completing the course, students are expected to achieve the specific learning outcomes: students should be able to do the following:

Have an understanding of definitions, scope and common properties of embedded systems from a variety of embedded applications in different industrial domains.

Possess the knowledge of basic organization and architecture of embedded systems.

Have an understanding of basic design flows for implementing embedded systems with hardware/software co-design.

Able to design and implement embedded software for application specific systems by utilizing specialized application software development platforms.

Have the expertise to design and implement the solution on hardware, including testing and verification techniques for embedded systems.

Project Objectives

Major intended objectives of the project, which helped us finalize the details, are summarized below.

1. Brainstorming Solutions: After learning the formal concepts of embedded systems and programming, it is of vital importance to practice. The most fundamental part of embedded solutions is to brainstorm the overall solution for the specific problem, including the hardware, software and interfacing parts [8], [9], [10]. For small projects, expertise in a single domain is sufficient. But, the increasingly competent world demands the professionals to work in the team while maintaining individual performance excellence. That is the main motivation of forming the groups to conduct projects: to familiarize students with brainstorming and critically analyzing solutions.
2. Practical Considerations: A practical project offers a diverse set of advantages, but the limitations must be accounted for while designing the task. The first and most restricting limitation is time: students only have one semester which includes exams and first few weeks are needed for the introduction. The workload from other courses and delays in shipment of hardware components further add to the constraint of time. Even though the cost of open-source components is not very high, still it can go high if a large number of sophisticated components is needed. Finally, in accordance with diverse backgrounds of students, necessary prerequisites, required to successfully conduct the project, must be delivered to students in a timely fashion.
3. Learning Open-Source Tools: We decided that a central goal of the project should be learning open-source hardware and programming tools based on three reasons. First, as embedded hardware (and programming tools) evolve, more and more easy and user-friendly tools are emerging in the industry, leading to the formation of many large communities and users. It is one of the most prized assets to master the skill of utilizing and contributing to such open-source ecosystems. Secondly, experience with such hardware and software help in job hunting, especially in emerging and technological markets. Thirdly, the utilization of such tools conforms with our goals (of practical hands-on learning) while adhering to the constraints of limited time and financial resources.
4. Holistic Solution Development: Software of embedded platforms are more demanding than those of conventional computers. They differ primarily because the computational and interfacing resources are limited and software is generally dependent on the capabilities of the hardware. System design around the particular embedded hardware, including component selection and interface design, is the necessary process which must be accomplished before programming. Practically, the challenges are not restricted to just programming, rather most

constraints come from the limitations of the embedded system and other connected devices. Considering these facts, the requirement of a holistic learning approach is imperative.

5. Develop Expertise of Physical System: Wearable electronics, smart phones and electronic gadgets have already contributed to the rise of new billion dollar industries which are expected to gain momentum in future. The added complexity of simultaneously dealing with various devices, with different interfacing and temporal requirements, makes such systems more challenging. To enable students to successfully learn the expertise to dive into the modern fields of smart electronics which interact with the environment through various sensing systems, expertise on physical hardware with actual sensors and actuators should be developed.
6. Complex System Realization: Last but not least, complete system realization by actual hardware implementation is an important step to test and validate the solution. Testing on actual embedded hardware is an additional step which is not required in software that run on standard computers. Computational as well as interfacing limitations of embedded systems, make it essential to test the solution on hardware and verify the feasibility and reliability of the whole design encompassing hardware and the embedded software.

Market Needs

The global market for embedded system is over 1.6 trillion dollars and compound annual growth rate is 10% [11]. Major application domains of embedded systems include communica-tions, computing, consumer electronics, energy, healthcare, and transportation [11]. The growth of embedded system industry, higher than that of traditional IT field, is 1.5 times faster than semiconductor sector [11]. Among all microprocessors in the world, embedded processors have a 98% share, thus leading computing power in information technology by some margin [12]. In 2014, the North American and Asia-Pacific markets had combined share of 65% of the global market [13]. In Asia Pacific region, market leaders include China, Taiwan and South Korea, having well-developed embedded system markets. Hong Kong aims to cement its place in the technical and innovative leaders of the world. The goal of “Digital 21 Strategy” is to become a leading information and technology hubs in the world [14]. Information technology and communication sector (ITC) employs nearly 78000 individuals which are around 2% of the overall workforce in Hong Kong [14]. Hong Kong government’s Emphasis on technology in general and ICT, in particular, means the technical and programming jobs will steadily continue to increase in future. The large and stable job market of Hong Kong is already enticing many graduates of closely-related fields to seek employment in programming jobs. That is why we have designed this course so as the students without rigorous background can also be fully engaged in the learning situation. Considering the growth in this area, it is desirable to fully equip the graduate students with future needs of the market.

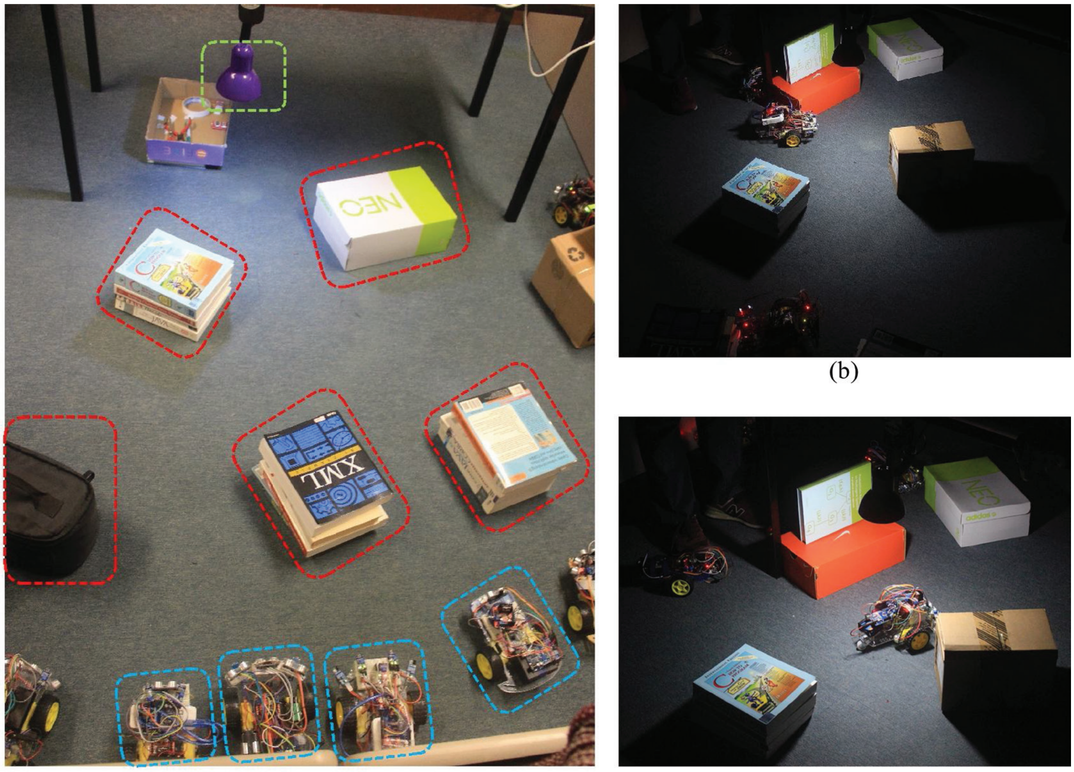
ROBOT COMPETITION: REACH THE LIGHT SOURCE

This is the time when finally, the student's effort will pay off. Although the core purpose of this exercise is learning based. But no learning is useful if it is not put to the test or passed through some healthy competition, this is also a way for the instructor to know how heartedly students have participated in the exercise and what they have learned throughout the course. This competition is not to make someone feel bad it is just a general assessment to see how useful the layout of the course was and it also develops the sense of group effort among the students. Based on the project goals, and conforming to constraints, we designed a robot competition in which robots programmed by the students contest. It is an inherent requirement for the task to be interesting but still achievable for students. All the equipments including, structure, sensors, wires, batteries etc are available on different on-line stores including, amazon, alibaba, taobao etc or one can also order the whole robotic-toolkit from robotshop (on-line store). The robot competition is held among robots in a dark room with a single bright light source. In the arena, there are some static obstacles, though other robots can be treated as dynamic obstacles. The scenario of robot competition is shown in Fig. 1.

A detailed video of actual robot competition can be seen at: https://www.youtube.com/watch?v=uEOBjOfq75Q.

All robots start at the same time, with the goal of reaching the light source before other robots, without hitting obstacles. Detailed rules of the competition are listed here:

1. The arena will be a closed area having multiple, solid obstacles (such as books and blocks).
2. There will be a single light source in the arena provided by a reading lamp. The lamp mounted at a height will be pointing downward creating a lit-up area of at least 30 cm radius.
3. All robots are uniformly placed and obstacles are evenly distributed by the instructors and student helpers.
4. To ensure fairness and remove the factor of luck, there will be a total of three runs of the competition. Starting position of robots and obstacles will be changed in every trial to fairly judge all robots.
5. Before the start of the competition, students will be given time to calibrate their sensors and conduct test runs of their robots.
6. A robot is said to complete its task if it reaches and stays in the light zone completely autonomously, without any human intervention. The robot must stop under the brightest region (beneath the lamp) for at least 10 seconds.
7. Robots are ranked based on the order of completing the task i.e. first robot to complete the task is the winner and so on. Once a robot has completed the task, it will be taken out of the arena.



(a) (b) (c)

Figure 1. Setup and execution of the robot competition: seek the light source. (a) The scenario of the arena just before start of the contest. Light is on complete the setting and to demonstrate the scene. Red boxes show the objects that are serving as obstacles and the green box shows the lamp which is illuminating a bright spot. Robots are marked with blue rectangles. (b) An instant during the competition. In accordance with rules of competition, placement of obstacles and initial positions of robots are randomly changed in every run. In this instant, we can see a robot is about to reach the brightest spot. (c) In this frame, we can see a robot has reached the light source, while another robot is nearby. A descriptive video of the competition is available at: https://www.youtube.com/watch?v=uEOBjOfq75Q.

8. There is no penalty for hitting the obstacle. However, a robot that cannot avoid obstacle may not be able to reach the light source.
9. If a robot is stuck in a corner or near an obstacle, the student may request instructor to restart the run. In this case, the robot will be put back in the starting position.

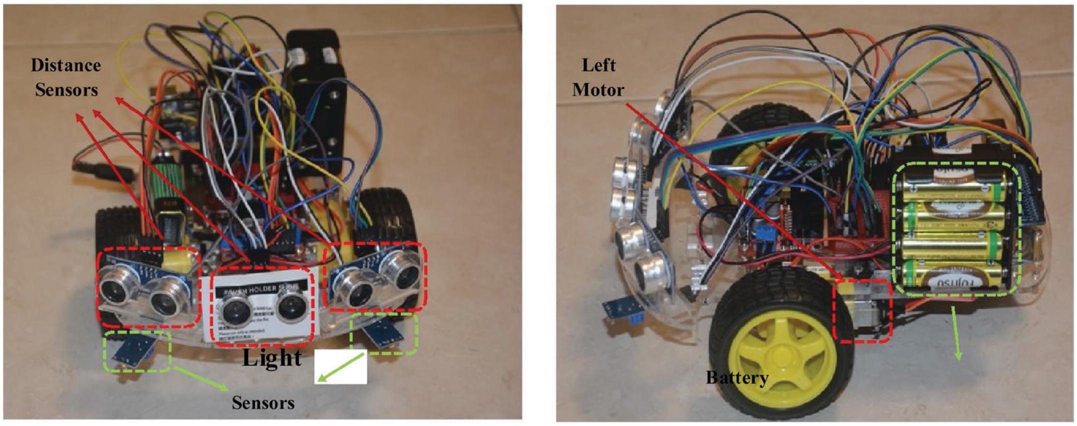


Figure 2. A typical robot developed by students in COMP 5228 course in Fall 2016. The read-made chassis is equipped with batteries, light sensors, distance sensors and micro-controller board. The chassis has two driving wheels and a supporting free wheel. A video (made by student) showing basic functions of the robot is available online at: https://www.youtube.com/watch?v=opnJ1zlHvSA.

CONCEPTUAL DESIGN OF LIGHT SEEKING ROBOT

In this section, we discuss the design and pragmatic considerations of the robotic platform. Picture of a typical mobile robot for this course is shown in Fig. 2. Traditional industrial robots with fixed base (also referred as manipulators) have been successfully used for decades [15], [16], [17]. Mobile robots, that can move around, offer unique solutions, challenges and commercial possibilities [18], [19]. An example robot video (uploaded by a student) of a robot can be seen at: https://www.youtube.com/watch?v=opnJ1zlHvSA.

Embedded Controller

Arduino is a prototyping platform which is used in the development of the project or the robotic kit. Arduino was originally introduced and designed in 2005 in Italy and since then it is treated as open-source which offers hardware acquisition and implementation. Arduino UNO is supported on Windows, Macintosh, Linux making it multi-platform based on IDE environment and the development is based upon the C language. For our project, we recommend Arduino UNO (or any of its variant) since it fulfills the computational and interface requirements. Genuino is the official version of Arduino UNO for markets outside the USA and there are many variants provided by other suppliers since this is an open-source platform. Availability of ready-made boards, that do not require the external device for programming, make their utilization easier without requiring low-level hardware expertise.

Many features like reliability, ease-of-use, open-source libraries and processing power play a vital role in the selection of Arduino boards. The Arduino UNO consists of ATmega328p, 8-bit ATMEL microcontroller, with internal, permanent memory called EEP-ROM. This memory is two-fold which is used for coding for creating applications and for storing temporary data variables for decision making. Lower power consumption is another major feature of the Arduino UNO board where it can be brought to life and function by providing a power from a 9 volts battery. The operation and performance of the UNO model have shown a great improvement and versatility over coming years, making it suitable for embedded applications like the robotic car.

Robot Chassis

Since the emphasis of this course is not on hardware, we advise students to buy ready-made robotic bases consisting of three or four wheels. These light-weight platforms are driven by the differential drive with a motor on the left and other on right with additional one or two supporting wheels. Both driving motors (left and right) are DC servo motors that are equipped with digital optical encoders that provide feedback of actual position and speed of the motor shaft. Availability of wheel motion information allows the precise feedback control of the individual motor, which can be used to develop accurate robot motion.

It is important to note that since the course project does not require very precise control, it is possible to operate the motors in an open-loop fashion by ignoring encoder data. On an abstract level, a robot might seek information from light sensors and choose to move towards the brightest spot and iteratively correct its direction solely based on light sensors. Indeed it was observed that many students implemented open-loop feedforward control of motors.

Light Sensor

The goal of robot competition is to program the motor actions responding to which light sensor receives the highest light source energy. Students have the option of using multiple light sensors and estimate the direction of the light source based on their relative outputs and line-of-sight. Alternatively, it is allowed to mount a light sensor on a servo motor and this way, the robot can 'search' for light source by rotating the motor. A light sensor module is comprised of a light-dependent resistor (LDR), also referred as photo-resistor and a variable resistor (potentiometer). The sensor module can be calibrated by adjusting its resistance to have maximum output difference under extreme cases: light and dark. The increased sensitivity of sensor allows for better sensing and overall improvement in control. A tutorial is dedicated to interfacing, calibrating and interpreting the reading from light sensors. Even though the sensor itself is analog (which is good because it provides more information than digital light-detectors), its interface is straightforward because internal Analog-to-Digital Converter (ADC) in Arduino boards remove the requirement of installing external ADC. A key observation noted is that students quickly find out that instead of looking up for the light source, it is easier to point sensors down and inspect the floor for light intensity, since there is the only single light source in the arena. This simple example shows the requirement of lateral thinking and innovative solutions adopted by students.

Distance Sensor

For obstacle detection, students have the choice to use either (or both) Infra-red (IR) or Ultrasonic sonar sensors. Typical IR sensor, such as GP2D12 [20] and GP2Y0A21 [21] by Sharp Microelectronics, have high accuracy but smaller range (usually less than 80 centimeters). Typical distance sensors provide analog voltage, corresponding to the distance of the object in front of the sensor. Again, the internal ADC of Arduino board can be used to read the value, which can be converted to distance (say in centimeter) based on a simple formula. On the other hand, ultrasonic sensors do not offer highly accurate result, but their range is generally longer. For example SRF05 by Devantech [22] can sense up to 4 meters. Typically these ultrasonic range sensors provide timing information only. Based on known speed on sound, the microcontroller has to calculate the distance by measuring the total time taken for an emitted sound wave (typically at 40 kilo-Hertz) to hit the nearest obstacle and reach back.

EMBEDDED SOFTWARE

The challenge posed by the project requires programming of different behaviors or procedures for the robot. First, we discuss the development of individual control modes, such as light seeking and collision avoidance. Then we present some approaches that deal with complex problems arising due to multiple requirements of the robot. The temporal interdependence of these apparently independent routines cannot be guaranteed. For example, the robot has to move towards light source but at the same time, it must deal with the obstacle in front. A figure showing behavior and procedures of a typical robot is shown in Fig. 3.

Collision Avoidance

Collision avoidance with obstacles and other robots is the fundamental ability for every robot. If a robot is unable to circumnavigate an obstacle in its path, its high-level functions (such as

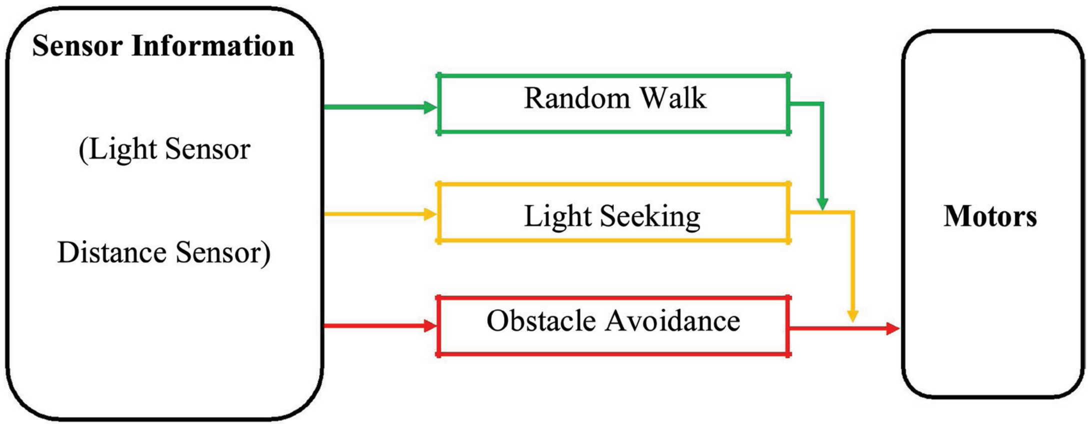


Figure 3. Behaviors and procedures of a light seeking robot which can avoid collision with obstacles.

light-seeking) are of no use. That is why, in Fig. 3, we can see that obstacle avoidance behavior is the most basic one (and shown in red color). From the implementation perspective, students might use either or both of IR and ultrasonic distance sensors to detect and avoid obstacles (and other robots).

Basic collision avoidance algorithms, such as bug algorithms, are introduced in this course. The moth-like behavior generated by bug-algorithms [23], [24], [25] is easy to implement. A simple functionality of wall-following can be extended to have different variants of bug algorithms, with the guarantee of finding the goal if it exists. In our settings, obstacles are convex and bounded, and practically there is always a way to reach the goal (no dead ends, except the boundaries of the arena). This makes it easier for robots to reply on simple bug algorithms for motion control to avoid the obstacle.

Other than low-level feedback control system, high-level motion planning concept such as artificial potential field [26], [27], [28] are also introduced. In potential field method, an arbitrary field is assumed where obstacles produce a repulsive force and the goal generates an attractive force.

Light Seeking

There are two basic ways of implementing the basic functionality of light seeking. Either a robot can have multiple light sensors at fixed positions and orientations on the robot base. Alternatively, a single light sensor can be mounted on a small servo motor which can sweep around. Turning angle of a typical servo motors varies from 180 to 360 degrees. As discussed earlier, calibration of individual light sensors is a vital step prior to sensor utilization, requiring adjustment of the on-board potentiometer to increase the sensor sensitivity. Since practically output ranges of sensors may be slightly different, for ease of programming, a simple solution can be implemented in software wherein data from all sensors is normalized to a uniform scale.

Various control algorithms are covered in theoretical content which can be used to develop a light-seeking behavior. In control system algorithms, perhaps the simplest method might be the implementation of bang-bang control [29]. In this approach, the reference may be the difference between the output of light sensors at left and right: in order to move towards the light source, the equally-partitioned left and right sensors should sense the same amount of light. Basics of Proportional-Integral-Derivate (PID) control [30], [31] is also introduced to move towards brightest place. Some students indeed opted to implement this method or its simplified form of Proportional-Integral or Proportional controller.

The artificial potential field can be used for light-seeking behavior. The robot can use a gradient-descent method in this artificial field to reach the goal position. IN our setting, the light source acts as the goal which attracts the robot. It is important to consider the relevant model for attractive force, especially considering the limited computational resources in our case. Attractive force that is linearly dependent on the distance between robot and goal is the simplest method, while another easy way is to use a quadratic function of distance-to-goal for calculation of attractive force. For motion planning, the aforementioned bug algorithm can also be used to reach the light source, by iteratively treating the brightest spot as the goal.

Subsumption Architecture

In behavior-based robotics [32] offers a rich theoretical basis for robots that have to deal with multiple, complex and often interconnected behaviors. In this domain, subsumption architecture [33], [34], [35] is a famous method that can formally design the inhibition and activation of different behaviors, depending on the situation. The subsumption architecture organizes the control layers according to their priority. For the light-seeking robot, the subsumption architecture of various control layers is shown in Fig. 3. We can see that the most basic and critical one is at the lowest level: obstacle avoidance (shown in red). If this behavior is activated (implying there is an obstacle in front of the robot), this behavior will inhibit high-level behaviors (light-seeking and random walking). If this behavior is inactive, the robot can continue to seek light, and as long as the robot is moving towards a light source, the random walk behavior will be suppressed. If both low-level behaviors (obstacle avoidance and light seeking) are inactive, only then the robot can randomly explore its surroundings to find a light source.

Multi-tasking

In a strictly theoretical sense, low-end Arduino boards are not capable of multi-threading or parallel processing. However, there are some possibilities which can suffice our original requirements of dealing with multiple inputs at the same time. Interrupts of Arduino can be used to monitor multiple sensors at the same time and some software libraries are available which provide the functionality close to multi-threading, referred as proto-threading.

Interrupts are a powerful tool for any embedded computer. Any enabled interrupt can cause the code to jump to its Interrupt Service Routine (ISR) and resume the code once the ISR is finished. In contrast to polling (a method where the input is monitored all the time manually), interrupts do not consume all the time of CPU waiting (or servicing) a particular input. The normal code continues running and when an interrupt has already occurred, its corresponding ISR is executed by temporarily suspending the normal code execution. In principal, the code execution can only deal with one device, however utilizing interrupts enables to look for multiple conditions at the same time. In our robot, a simple way could be to have the normal code for light seeking behavior and put the obstacle detection on interrupt. If there is any object in front of the robot, ISR of obstacle avoidance will begin, temporarily putting the light seeking behavior on hold.

An alternative solution is CPU time-division among multiple processes (code functions) in a round-robin fashion. The execution time of different segments of codes (typically in milliseconds) can be specified, which enables designers to allocate more computation time to the tasks of higher complexity and importance. The simple time-division capability, referred as proto-threading, provides an easy and neat way to organize multiple distinct routines in the code.

TEACHING METHODOLOGY

In order to fully engage students in the challenge-based learning, and considering heterogeneous backgrounds, special care is taken to design and plan the lectures and tutorials. Assess-ments are also stretched throughout the semester to motivate students and keep them engaged in the project. An overview of the tutorials and assessments with respect to their timing is shown in Table I.

Grading Criteria

In grading, it is required to keep a balance between theoretical contents and the grand challenge of programming the robot. Exams of the course contributed a total of 55% of the overall grading.

Table I. Plan of tutorials and assessments about the grand challenge.

Week	Module	Description
1	Introductory lecture	Students are briefed about the course, the grand challenge and schedule of assessments.
2	Tutorial 1	Basics of Arduino controllers and programming environment is introduced.
3	Tutorial 2	Arduino Programmin inclusing Input/Output, Serial communication and arith-metic operations are explained.
4	Tutorial 3	An overview of electric circuits and interfacing with light sensor.
5	Tutorial 4	Interfacing and visualizing data from Ultrasonic distance sensor.
6	Tutorial 5	Process of making Android Apps using AppInventor2 is explained.
7	Progress Assessment	All students are asked to present progress of their projects including overall design, preliminary results, problems encountered and future plan.
8	Tutorial 6	Interfacing Embedded controller with Android device using Bluetooth communication.
9	Tutorial 7	An example Android app using location sensor and displaying information on Google Maps.
10	Tutorial 8	Interfacing servo motor with embedded controller using digital optical en-coders.
11	Tutorial 9	Wireless communication using low-cost components and open-source li-braries.
12	Tutorial 10	A compilation of interesting projects which students can pursue on their own.
13	Robot Competition	All robots compete in the final contest. Students also present their final design and improvements they made since progress presentation.

Midterm progress presentation accounts for 15% of the marks. Final project presentation and robot competition contribute 30% in the final grade. This grading scheme meets the institutional guidelines of having at least 55% contribution of theoretical examinations.

PILOT TESTING AND EVALUATION RESULTS

Pedagogical Consideration

Embedded programming is a multi-disciplinary field which requires the expertise in hardware, software and the holistic design process. In this course, we opted a challenge-based learning method, instead of traditional knowledge or assessment-based strategies. Central to this theme is the project of programming a real robot. A major advantage of challenge-based learning is the increase in motivation by team building and technical collaboration, required to complete the competitive project. Indeed it was observed that students went on to voluntarily do the additional, ungraded tasks for sole purposes of learning and accomplishment. Because of working in groups, students brainstorm and generate ideas to accomplish the project. Theoretical contents covered in lectures are tailored to equip students with major aspects of the challenge and the key areas of knowledge. Students continue working on their designs and plans and revise them for improvement, based on practical examples and new tools introduced in the weekly hands-on tutorials. Students' ideas are put to the test in lab modules and incremental milestones for the project. To ensure the continuation of research and brainstorming cycle, students are required to present their progress midway through the semester and they are provided individual feedback on their progress and directions for further improvement. In addition to mastery of subject contents, brainstorming and collaboration with peers develops innovative skills in students. Group work also leads to the team building and collaborative work among students which adds to the overall learning experience in the course. It was observed that some groups extended help towards others, despite the competitiveness of problem, showing the conducive learning environment and interactive exposure.

The problems faced in dealing with multiple hardware and software modules (and their interfacing) during project require quick interventions by students and often they have to do some study to find the solutions, which are generally simple. This leads to adaptive learning which encompasses knowledge as well as innovative thinking in new contexts and problems. In summary, challenge-based approach assists in delivering real case studies in cutting-edge practical education and promotes adaptive learning, knowledge acquisition and innovation.

Evaluation Methods

Let us critically analyze the survey results of this course, based on feedback provided from students who took this course in Fall 2015 and Fall 2016. A questionnaire was provided to all students before the end of the semester and they were requested to anonymously fill it online, using Google Forms. Out of 35 students that completed this course in two semesters, 32 responses were received. Questions were grouped into four major parts: teaching content, course impact, project support and grading criteria. All questions are shown in Table II. The response type of all questions is shown in Table III.

Results

Results of answers of questions 1 to 6 are shown in Fig. 4. Results of remaining questions (7 to 11) about teaching are presented in Fig. 5. We can see that for all questions (with the exception of one) the average response is above 4, meaning the majority agrees with the suitability of the content. Only question number 10 had a mean response of 3.77, where 3 refers to partially agree and 4 means the agreement of student with the content. Question 10 inquired students if they had the required knowledge to complete the project without additional study outside the course. While we have been trying to include all the relevant content in the lectures and tutorials, some students without relevant technical background have to spend time on their own.

Table II. Questionnaire provided to students before the end of semester.

	Teaching Content and Pedagogy
1)	The lecture project motivated me towards the subject of Embedded Software Engineering.
2)	I found the project an interesting application of embedded software programming.
3)	The lecture materials helped me understand Embedded Software Engineering.
4)	Tutorials were helpful in understanding and completing my project.
5)	The course helped me to learn theoretical concepts of Embedded Software engineering.
6)	The project familiarized me with practical issues of embedded programming.
7)	I am confident about interfacing and programming many devices (sensors etc) with embedded controllers.
8)	I am confident about interfacing and programming android devices with embedded controllers.
9)	Overall, the project improved my learning experience.
10)	I had the knowledge and skill to complete this project without additional study beyond the lecture.
11)	The project was easy to carry out and did not take me too much time.
	Course Impact
12)	My understanding of the Embedded Software Engineering BEFORE I took the course.
13)	My understanding of the Embedded Software Engineering AFTER taking the course.
	Project Support
14)	Lectures and tutorials provided me with sufficient background knowledge to understand and plan my project.
15)	I was provided satisfactory assistance to finalize components of my project.
16)	My practical problems in embedded system integration and programming were answered by instructor.
17)	I am satisfied with the total spending on this project.
	Grading Creiteria
18)	I am satisfied with evaluation based on embedded programming project (progress presentation + final demo).
19)	Evaluation components reflect the material emphasized in the course.
20)	Difficulty level in the evaluation components is commensurate with what is taught in the class.
21)	Evaluation is fair and transparent.

Table III. Types of responses of questions in the survey form.

Questions	Response Types
1–11	5 = strongly agree, 4 = agree, 3 = partially agree, 2 = disagree, 1 = strongly disagree.
12–13	5 = highest level of understanding, . . . 1 = lowest level of understanding.
14–21	5 = strongly agree, 4 = agree, 3 = partially agree, 2 = disagree, 1 = strongly disagree.

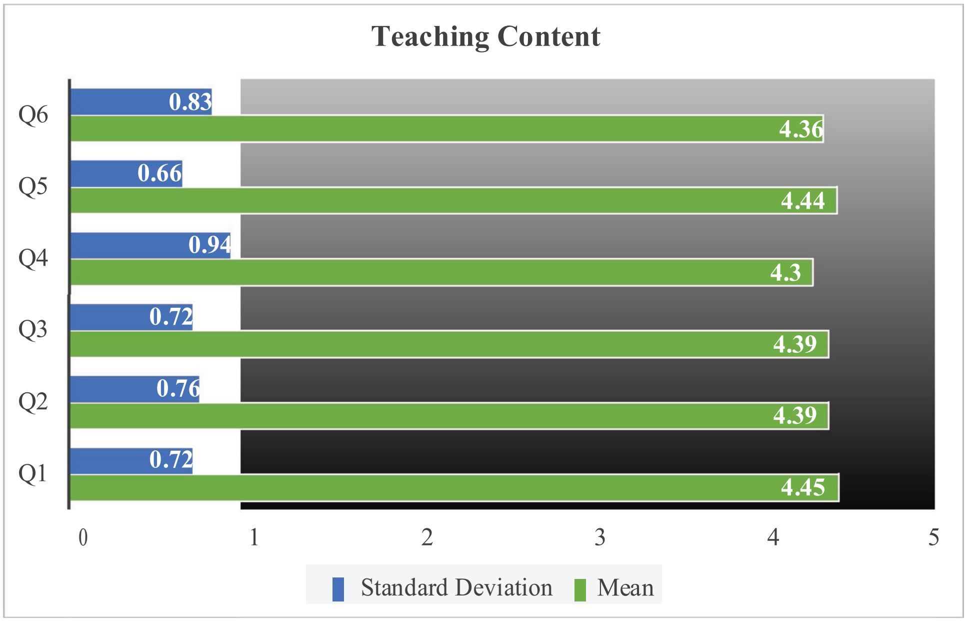


Figure 4. Survey results about teaching content and project ranging from question 1 to question 6 of the questionnaire. Questions can be seen in Table II. 5: strongly agree, 4: agree, 3: partially agree, 2: disagree, 1: strongly disagree.

Since the idea of challenge based learning is developed around the practical project, we included specialized questions about support and difficulties encountered by students in the project. Results of questions pertaining project-related issues (question 14–17) are presented in Fig. 6. It can be observed that despite the introduction of a challenging project, students agreed that enough support and guidance was provided to them through lectures and tutorials.

Students' opinion about grading and assessment of the whole course, with project-related evaluations forming a major part, are inquired in questions 18–21 of the survey. The results of student responses about the grading of this course are presented in Fig. 7. With all average scores higher than 4, it can be noted that students are satisfied with the grading criteria of the course.

Finally, and most importantly, let us examine the response of students about the overall impact of the course and its significance in learning of embedded software engineering. Students were asked about their understanding before and after taking the course (question 12–13), on a scale of 1 (lowest level of understanding) to 5 (highest level of understanding). As we can see in Fig. 8, average perceived understanding of students has improved from 2:92 to 4:71 which shows the significant

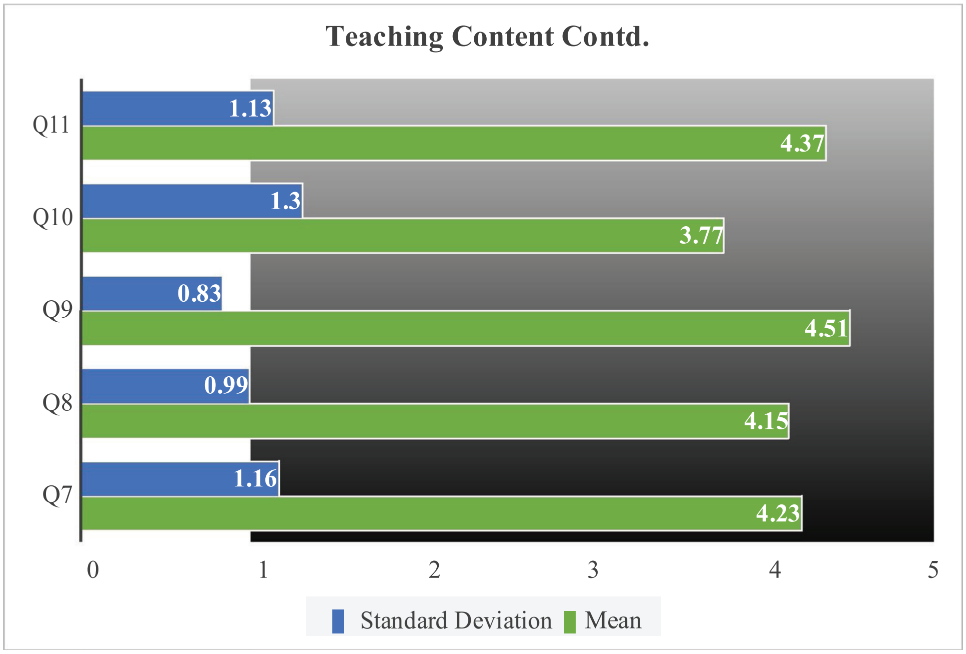


Figure 5. Survey results about teaching content, question 7 to 11, of the questionnaire. Except question 10, average response of all questions is higher than 4.1. Questions can be seen in Table II. 5: strongly agree, 4: agree, 3: partially agree, 2: disagree, 1: strongly disagree.

impact of the challenge-based teaching adopted in this course. Other than direct feedback from the students, observations of instructors also confirm the positive impact of challenge based approach in the overall learning experience of the students.

Lesson Learns

Since we have carried the pilot testing for two semesters (fall 2015 and fall 2016), we have identified and addressed the major issues and areas of weaknesses for students. Most importantly, students who lack relevant prior expertise need the assistance and confidence to perform well. We found out that personalized help is useful, but more importantly (and interestingly), we observed that small tutorials and even short handouts gave them more confidence. For example, the handout we made to explain them about the wiring and battery calculation was appreciated by most of the students. Similarly, based on past experience, in 2016, we incorporated an easy-to-use purchasing list concept: all students were provided a list of all required components which was sufficient to complete the project. However, to harness the innovation, students were given freedom to buy extra components of their choice. The component list provided by every group was reviewed by

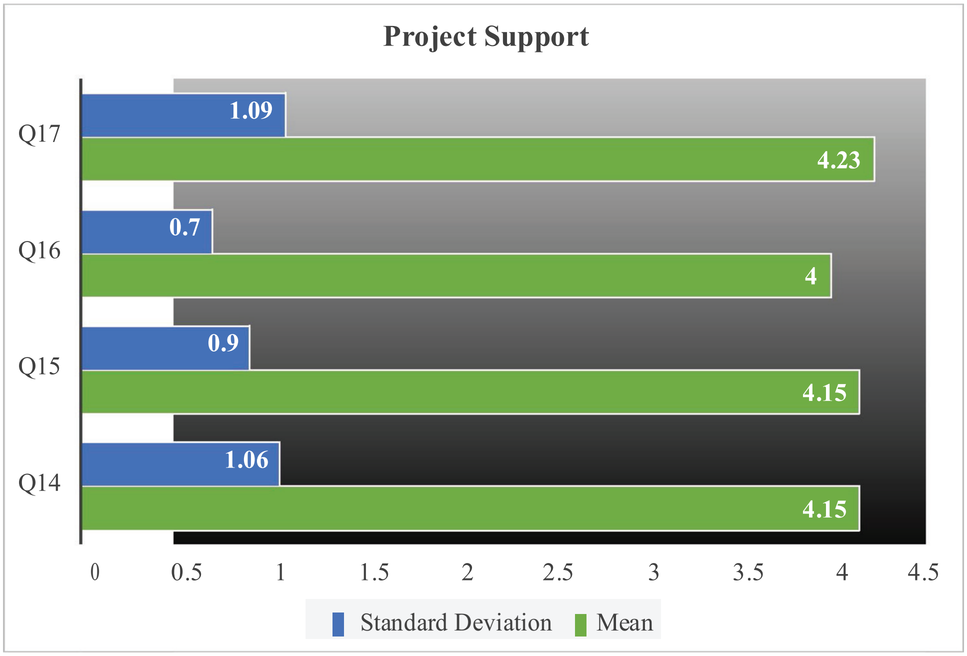


Figure 6. Students' response about support and guidance about project, question 7 to 11, of the questionnaire. We can see that all responses have an average of 4 or higher which shows satisfaction of the students. Questions can be seen in Table II. 5: strongly agree, 4: agree, 3: partially agree, 2: disagree, 1: strongly disagree.

instructors and every group was informed about any missing component before the end of the third week of the semester. This vital step required an investment of instructors' time in the beginning, but it helped the students by standardizing hardware purchase and reducing their waiting time for shipment of components ordered late.

Adoption of Arduino was widely appreciated by our students and plethora of helpful and interesting materials on the internet further motivated students to explore things on their own. Similarly, android interfacing with Arduino and development of android applications received positive feedback from students. Ideas of future work of students and additional hardware and software applications they developed (which were not covered under course grading) showed that students consider these valuable and interesting tools which they would pursue in the future as well.

Course contents and hands-on tutorials were specially designed and timed to assist students in completing their projects in a timely fashion. In fall 2016, we observed that robots of all the groups accomplished both the tasks: light seeking and obstacle avoidance.

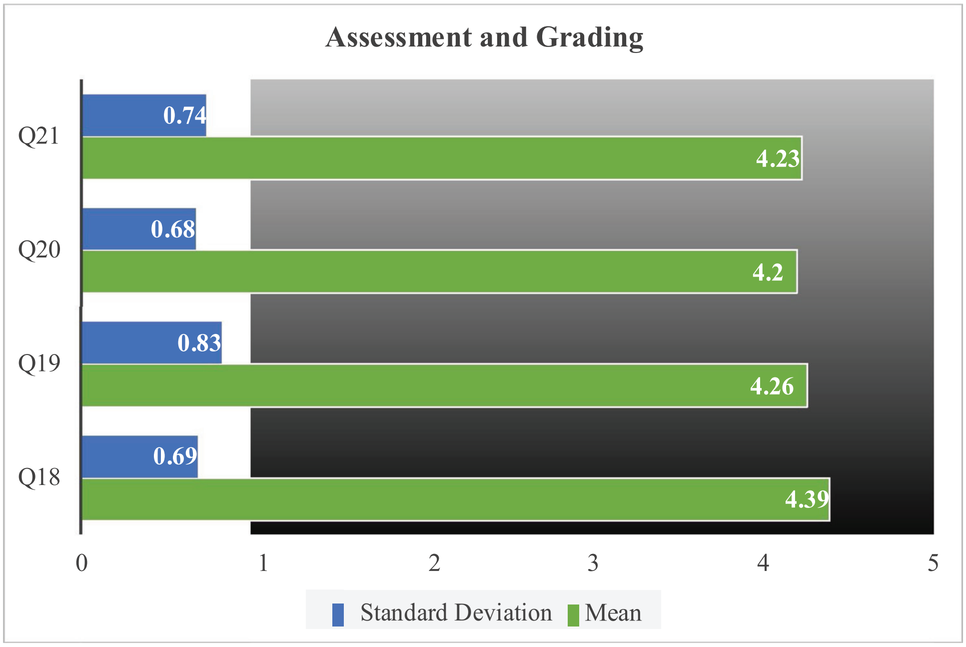


Figure 7. Survey results about grading and assessment, question 18–21, of the questionnaire. Questions can be seen in Table II. 5: strongly agree, 4: agree, 3: partially agree, 2: disagree, 1: strongly disagree.

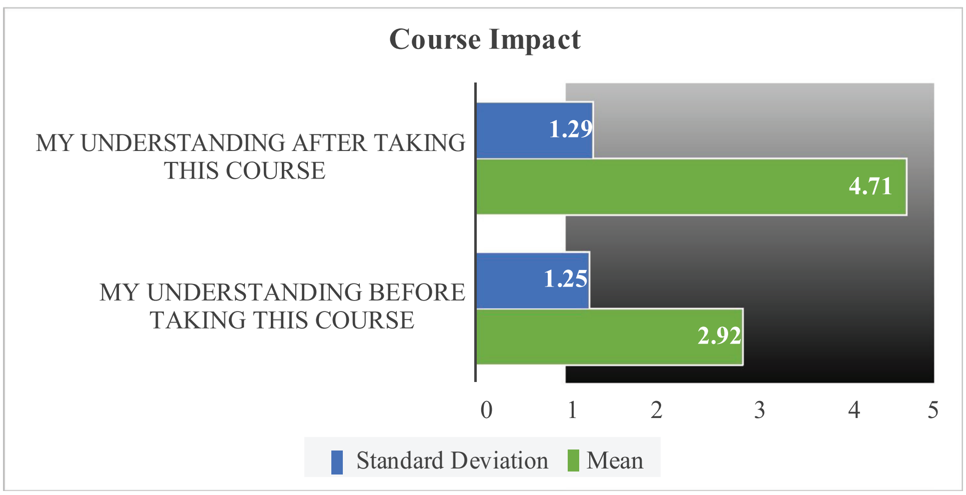


Figure 8. Student opinion about course impact, question 12–13. Questions can be seen in Table II. 5: highest level of understanding, 1: lowest level of understanding.

By design we reduced the grading proportion of robot competition on overall grade, mainly to let students focus on their own strengths (and weaknesses) and to create a collaborative and supportive environment. In the end, a healthy competition among students was observed where victory was celebrated in good spirit.

Challenges and Further Improvement

We identify that despite improvement in the second year, perhaps there is still room for improvement in the area of providing background knowledge. The question that asked if stu-dents had the required background knowledge to complete the project without additional study (question 10 in the questionnaire, Table II) received the least average score of 3.77 (on a scale of 1 to 5). Since people of different backgrounds join this course, we understand that perhaps another written step-by-step tutorial about the project will be useful to build the basic knowledge and familiarity of students, in addition to existing introductory lecture and interactive tutorial sessions.

The challenges we faced in organizing and while carrying out the course are listed below:

- Trade-off of coursework complexity with time requirements.
- Objective was to use the bottom- up approach for learning which is complex and requires lot of effort and time.
- Embedded systems are complicate and difficult to apprehend fully in a limited time espe-cially when you are focusing on the basis of the course as well.
- Although different open-source tools were used while teaching but some complex tools required more time for the students to fully grasp their working.

The course is designed in a way that the students with no background in electronics can also participate in it. This exercise of learning treats every student equally, while learning we will go from bottom to the top, exploring each aspect of the hardware and the software necessary.

The purpose of this exercise was to introduce the students with new way of learning that is why we chose the subject which involves both hardware and software. But this course can also accommodate purely software and hardware-based subjects as well. For example, programming, software architectures, computer hardware?s architectures, circuit analysis etc.

CONCLUSIONS

In this paper, we discuss the methodology and case study of employing challenge-based learning in embedded programming. We explain the technical challenges and key areas of knowledge to engage students in such projects. Additionally, we also provide a detailed account of the robot competition which can be adopted with minimal hardware effort, thanks to recent progress in modular kits. Imparting state-of-the-art education and cutting edge technologies is challenging because students with heterogeneous backgrounds register for this course. Therefore, open-source development tools such as Arduino, with interfacing and practical considerations, are covered in this course. These tools encompass the increasing job market including consumer electronics, such as smart electronic gadgets. To further motivate students, a robot contest is designed in which every group of students assembles and program their own robot to reach the light source while avoiding obstacles in the path. Pilot testing of the course was conducted for two semesters. Based on anonymous surveys, results of this course are promising and highlight increased understanding and motivation among students. We believe that with careful design and capitalization of easy-to-use tools, students without rigorous background can be trained for professional excellence.

REFERENCES

[1] I. Russell, K. H. Jin, and M. Sabin, "Make and learn: A cs principles course based on the arduino platform," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. ACM, 2016, pp. 366–366. [Online]. Available: https://dl.acm.org/citation.cfm?id=2925490

[2] T. Warren, "Arduino: The ultimate guide to arduino for beginners including arduino basics, tips & tricks, projects, and more!" 2015. [Online]. Available: https://www.amazon.com/Arduino-Ultimate-Beginners-Including-Projects-ebook/dp/B00WSICAB4

[3] L. F. Johnson, R. S. Smith, J. T. Smythe, and R. K. Varon, "Challenge-based learning: An approach for our time." New Media Consortium, 2009. [Online]. Available: https://eric.ed.gov/?id=ED505102

[4] M. Jou, C.-K. Hung, and S.-H. Lai, "Application of challenge based learning approaches in robotics education," International Journal of Technology and Engineering Education 2010, Vol. 7 No. 2, p. 17, 2010. [Online]. Available: http://ijtee.org/ijtee/system/db/pdf/72.pdf#page=21

[5] S. Li, L. Jin, and M. Aquil, Kinematic Control of Redundant Robot Arms Using Neural Networks. Wiley-IEEE Press, 2019. [Online]. Available: https://www.amazon.com/Kinematic-Control-Redundant-Neural-Networks/dp/1119556961

[6] S. D. Wurdinger, Time for action: Stop teaching to the test and start teaching skills. R&L Education, 2012. [Online]. Available: https://eric.ed.gov/?id=ED528556

[7] W. B. Gaskins, J. Johnson, C. Maltbie, and A. Kukreti, "Changing the learning environment in the college of engineering and applied science using challenge based learning." International Journal of Engineering Pedagogy, vol. 5, no. 1, 2015. [Online]. Available: https://online-journals.org/index.php/i-jep/article/view/4138

[8] M. Hagen, A. Bernard, and E. Grube, "Do it all wrong! using reverse-brainstorming to generate ideas, improve discussions, and move students to action," Management Teaching Review, p. 2379298116634738, 2016. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/2379298116634738?journalCode=mtra

[9] C. Boddy, "The nominal group technique: An aid to brainstorming ideas in research," Qualitative Market Research: An International Journal, vol. 15, no. 1, pp. 6–18, 2012. [Online]. Available: https://www.joe.org/joe/1984march/iw2.php

[10] Z. Rao, "Training in brainstorming and developing writing skills," ELT journal, vol. 61, no. 2, pp. 100–106, 2007. [Online]. Available: https://academic.oup.com/eltj/article-abstract/61/2/100/365857?redirectedFrom=fulltext

[11] M. Surti and C. Panchal, "A study: Identifying the transformation of the embedded systems industry," International Journal of Research and Engineering (IJRE), vol. 2, no. 1, pp. 28–33, 2015. [Online]. Available: https://digital.ijre.org/index.php/int_j_res_eng/article/view/41

[12] C. Ebert and C. Jones, "Embedded software: Facts, figures, and future," Computer, vol. 42, no. 4, pp. 0042–52, 2009. [Online]. Available: https://www.semanticscholar.org/paper/Embedded-Software%3A-Facts%2C-Figures% 2C-and-Future-Ebert-Jones/cdc7ac99f16729c84f01eea8e5d538c500f44eeb

[13] "Embedded system market - global industry analysis, size, share, growth, trends and forecast 2015–2021." [Online]. Available: http://www.transparencymarketresearch.com/embedded-system.html

[14] D. C. Donald, G. Mok, and A. Fong, "A people's market of hong kong," Finance, Rule of Law and Development in Asia: Perspectives from Singapore, Hong Kong and Mainland China, p. 220, 2016. [Online]. Available: https://brill.com/view/title/31615

[15] S. Li, L. Jin, and Y. Zhang, Kinematic Control of Redundant Manipulators Using Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 2017, vol. 10. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7499812

[16] T. Braunl,¨ Embedded robotics: mobile robot design and applications with embedded systems. Springer Science & Business Media, 2008. [Online]. Available: https://www.amazon.com/Embedded-Robotics-Mobile-Applications-Systems/dp/3540705333

[17] S. Li, H. Wang, and U. Rqfique, A Novel Recurrent Neural Network for Manipulator Control With Improved Noise Tolerance. IEEE Transactions on Neural Networks and Learning Systems, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7896650

[18] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to autonomous mobile robots. MIT press, 2011. [Online]. Available: https://mitpress.mit.edu/books/introduction-autonomous-mobile-robots-second-edition

[19] Y. Guo, B. Bran, and S. Li, Multi-robot Cooperative Control for Monitoring and Tracking Dynamic Plumes. IEEE International Conference on Robotics and Automation (ICRA), 2014. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6906591

[20] "Gp2d12 data sheet - sharp microelectronics of the americas." [Online]. Available: http://www.sharpsma.com/webfm_send/1203

[21] "Gp2y0a21yk data sheet - sharp microelectronics of the americas." [Online]. Available: http://www.sharpsma.com/webfm_send/1208

[22] "Devantech srf05 sonar range finder." [Online]. Available: https://acroname.com/products/devantech-srf05-sonar-ranging-module?sku=r271-srf05

[23] J. Ng and T. Braunl,¨ "Performance comparison of bug navigation algorithms," Journal of Intelligent and Robotic Systems, vol. 50, no. 1, pp. 73–84, 2007. [Online]. Available: https://link.springer.com/article/10.1007/s10846-007-9157-6

[24] C. H. Chiang, J.-S. Liu, and Y.-S. Chou, "Comparing path length by boundary following fast matching method and bug algorithms for path planning," in Opportunities and Challenges for Next-Generation Applied Intelligence. Springer, 2009, pp. 303–309. [Online]. Available: https://www.iis.sinica.edu.tw/papers/liu/8401-F.pdf

[25] F. Meddah and L. Dib, "E-bug: New bug path-planning algorithm for autonomous robot in unknown environment," in Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication. ACM, 2015, p. 69. [Online]. Available: https://dl.acm.org/citation.cfm?id=2816864&dl=ACM&coll=DL

[26] G. Li, S. Tong, G. Lv, R. Xiao, F. Cong, Z. Tong, A. Yamashita, and H. Asama, "An improved artificial potential field-based simultaneous forward search (improved apf-based sifors) method for robot path planning," in Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on. IEEE, 2015, pp. 330–335. [Online]. Available: https://ieeexplore.ieee.org/document/7405075

[27] H. M. Choset, Principles of robot motion: theory, algorithms, and implementation. MIT press, 2005. [Online]. Available: http://biorobotics.ri.cmu.edu/book/booboo book.pdf

[28] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on, vol. IEEE, 2001, pp. 1530–1535. [Online]. Available: https://ieeexplore.ieee.org/document/931933

[29] H. Maurer and N. P. Osmolovskii, "Second order sufficient conditions for time-optimal bang-bang control," SIAM journal on control and optimization, vol. 42, no. 6, pp. 2239–2263, 2004. [Online]. Available: https://epubs.siam.org/doi/10.1137/S0363012902402578

[30] B. Friedland, Control system design: an introduction to state-space methods. Courier Corporation, 2012. [Online]. Available: http://store.doverpublications.com/0486442780.html

[31] S. Li and Y. Guo, Distributed source seeking by cooperative robots: All-to-all and limited communications. IEEE International Conference on Robotics and Automation, 2013. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6224713

[32] R. C. Arkin, Behavior-based robotics. MIT press, 1998. [Online]. Available: https://mitpress.mit.edu/books/behavior-based-robotics

[33] R. A. Brooks, "Elephants don't play chess," Robotics and autonomous systems, vol. 6, no. 1, pp. 3–15, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889005800259

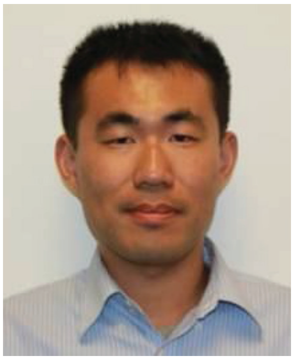
[34] S. G. Tzafestas, "Intelligent control system architectures," in Sociorobot World. Springer, 2016, pp. 25–39. [Online]. Available: https://www.springer.com/la/book/9783319214214

[35] A. M. Axelrod, H. A. Kingravi, and G. V. Chowdhary, "Gaussian process based subsumption of a parasitic control component," in 2015 American Control Conference (ACC). IEEE, 2015, pp. 2888–2893. [Online]. Available: https://ieeexplore.ieee.org/document/7171173

AUTHORS

Muhammad Usman Rafique received the bachelor's and master's degrees in mechatronics engineering from the National University of Sciences and Technology, Karachi, Pakistan, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He was with the National Space Agency of Pakistan (SUPARCO), Karachi, from 2007 to 2008. From 2010 to 2015, he was a Lecturer of Mechatronics Engineering with Air University, Islamabad, Pakistan. His current research interests include mobile robots, manipulation, motion planning, machine learning, computer vision, and control systems.

Mohammed Aquil-Mirza received the B.E. (first class honors) in Information Technology from Osmania University, India (2007). He received M.S. by Research in VLSI and Embedded Systems (with distinction) from International Institute of Technology-Hyderabad, India (2009) and M.S. in Electrical Engineering (first class honours) from King Abdullah University of Science and Technology, Saudi Arabia (2012). He is currently pursuing the Ph.D. in Computing at The Hong Kong Polytechnic University (PolyU) under the supervision of Dr. Shuai Li. His current research involves robotics, dynamic systems and control, recurrent neural networks, distributed control and optimization.



Shuai Li received the B.E. degree in precision mechanical engineering from Hefei University of Technology, China in 2005, the M.E. degree in automatic control engineering from University of Science and Technology of China, China in 2008, and the Ph.D. in electrical and computer engineering from Stevens Institute of Technology, USA in 2014.

Dr. Seifedine Kadry has a Bachelor degree in applied mathematics in 1999 from Lebanese University, MS degree in computation in 2002 from Reims University (France) and EPFL (Lausanne), PhD in applied statistics in 2007 from Blaise Pascal University (France), HDR degree in 2017 from Rouen University. At present his research focuses on education using technology, system prognostics, stochastic systems, and probability and reliability analysis. He is ABET program evaluator.

Ameer Tamoor Khan did my bachelor's in Electrical Engineering from Pakistan Institute of Engineering and Applied Sciences, Pakistan. I worked as a Research Assistant at Hong Kong Polytechnic University. Now I am pursuing my Ph.D. in computing from Hong Kong Polytechnic University, Hong Kong under the supervision of Dr. Shuai Li. My research area includes; optimization and path planning of soft robotic systems.