Designing instructional videos and classwork activities: teaching internet of things via flipped classroom

Chi-Un Lei*

Technology-Enriched Learning Initiative,
The University of Hong Kong,
Pokfulam Road, Hong Kong
Email: culei@hku.hk
*Corresponding author

Cheuk-Wang Yau, K-S. Lui and
Vincent Tam

Department of Electrical and Electronic Engineering,
The University of Hong Kong,
Pokfulam Road, Hong Kong
Email: cwyau@eee.hku.hk
Email: kslui@eee.hku.hk
Email: vtam@eee.hku.hk

Allan H-K. Yuen

Faculty of Education,
The University of Hong Kong,
Pokfulam Road, Hong Kong
Email: hkyuen@hku.hk

Edmund Y. Lam

Department of Electrical and Electronic Engineering,
The University of Hong Kong,
Pokfulam Road, Hong Kong
Email: elam@eee.hku.hk

Abstract: Internet of Things (IoT) applications demonstrates great potential to improve our quality of life and efficiency of business operations. Traditionally, students are taught engineering concepts in lectures. However, this may not be the most effective approach for teaching students IoT, since it can hardly allow students to gain exposure to relevant concepts, programming practices, and system-level design of a practical application. In order to help students get hands-on experience, an undergraduate course focusing on IoT application development was flipped recently: Classroom-based lectures were replaced by a series of student-paced online video lectures, and most weekly meetings were transformed into laboratory sessions facilitated by the course team. In this paper, we (1) discuss the rationale for designing the course structure and learning activities, (2) evaluate the course effectiveness through analysing students' video viewing behaviour and their project deliverables, and (3) propose future pedagogical development for scaling up students' learning.

Keywords: IoT; internet of things; flipped classroom; classwork design; instructional videos; video analytics; blended learning; design project.

Reference to this paper should be made as follows: Lei, C-U., Yau, C-W., Lui, K-S., Tam, V., Yuen, A.H-K. and Lam, E.Y. (2019) 'Designing instructional videos and classwork activities: teaching internet of things via flipped classroom', *Int. J. Mobile Learning and Organisation*, Vol. 13, No. 4, pp.392–411.

Biographical notes: Chi-Un Lei is currently an eLearning Technologist in Technology-Enriched Learning Initiative, The University of Hong Kong. He is also a Honorary Assistant Professor in the Department of Electrical and Electronic Engineering, The University of Hong Kong.

Cheuk-Wang Yau received the BEng and MPhil degrees from the Department of Electrical and Electronic Engineering, The University of Hong Kong. He is currently a Software Engineer in Technology-Enriched Learning Initiative, The University of Hong Kong.

K-S. Lui is currently an Associate Professor in the Department of Electrical and Electronic Engineering, The University of Hong Kong.

Vincent Tam is currently a Principal Lecturer in the Department of Electrical and Electronic Engineering, The University of Hong Kong.

Allan H-K. Yuen is currently an Associate Professor in Information and Technology Studies and Deputy Director of the Centre for Information Technology in Education, Faculty of Education, The University of Hong Kong.

Edmund Y. Lam is currently a Professor in the Department of Electrical and Electronic Engineering, The University of Hong Kong.

This paper is a revised and expanded version of a paper entitled 'Teaching Internet of Things: Enhancing Learning Efficiency via Full-Semester Flipped Programming' presented at the 'IEEE International Conference on Teaching, Assessment, and Learning for Engineering', Hong Kong, 12–14 December 2017.

1 Introduction

The concept of Internet of Things (IoT) evolved from connecting pervasive cyber-physical systems to the internet. IoT devices, such as sensor nodes, wearable devices, and "smart" everyday objects, are utilised to acquire a tremendous volume of different types of data from the environment (Atzori et al., 2010; Gubbi et al., 2013), in order to create better working and learning spaces in a mobile environment. It is anticipated that analysing and visualising the collected data can help generate meaningful insights and enable us to better understand our surroundings, such that governments, corporates, and individuals can make better-informed decisions (Yau et al., 2018). Despite the great potential of IoT to improve the quality of life and efficiency of business operations in numerous sectors, developing IoT applications is not without challenges. Compared to conventional software engineering, embedded systems and mobile applications development, designing IoT applications involves higher levels of technical complexity and stronger multidisciplinarity, as it requires the developers to take into considerations interactions with the physical world, network constraints, volume of data, system scalability, etc. (Gubbi et al., 2013). When teaching students IoT application development, it is necessary for teachers to ensure that they have adequate exposure to these engineering concepts, skills, and more importantly, system-level design and integration of such components into a practical application. In traditional lectures, students hardly gain exposure to these complicated concepts. Therefore, it might not be an effective approach to teach IoT application development.

Previous studies have shown that various blended learning and flipped classroom approaches had been widely adopted in the teaching of computer programming and other courses (DeLozier and Rhodes, 2017; Giannakos et al., 2014; Tanguma et al., 2008). These approaches create new opportunities for students to actively solve problems and apply knowledge with greater flexibility. Existing flipped classroom approaches have been used in teaching programming in both introductory (Jonsson, 2015) and advanced courses (Garcia and Cano, 2014; Maher et al., 2015). However, limited in-depth studies have been conducted to evaluate the considerations, process and effectiveness of implementing flipped classroom and blended learning in programming.

Recently, a full-semester undergraduate-level course on IoT and programming practices had been flipped (Lei et al., 2017a, December). Classroom-based lectures were replaced by a series of student-paced online video lectures. Meanwhile, most regular face-to-face sessions were conducted as laboratory classes facilitated by the course team, which consisted of an instructor and a Teaching Assistant (TA), in order to provide students with hands-on experience in programming with IoT devices. Through analysing online courseware usage and conducting interviews, the flipped mode was proven to be effective in (1) generating and retaining interactions among students and instructors; and (2) improving students' engagement with the content. However, the engagement of instructional videos, various aspects of implementations and the course project as a whole was not investigated thoroughly in our previous work. In particular, the effectiveness of multimedia instructional materials and course project was not comprehensively examined in our previous work.

In this paper, we hope to further investigate our previous (Lei et al., 2017b, December) discussion, including:

- Explaining the design rationale of the course structure, instructional videos, and learning activities.
- Analysing students' video viewing behaviour and projects developed by students.
- Proposing future pedagogical development for scaling up students' learning (e.g., adoption of mobile learning).

The development of the course is first described in Section 2. The design of videos and learning activities is then discussed in Section 3. Course evaluation, including analysis of video usage and performance of student projects, is presented in Section 4. Finally, proposals for future development can be found in Section 5.

2 Course development

2.1 Learning outcomes and course development

The 13-week undergraduate course discussed in this study aims at introducing principles of software development in constrained IoT devices (e.g., sensor nodes, portable and wearable devices), with a particular focus on the common issues and solutions in developing a practical IoT application (in contrast to desktop or mobile applications taught in typical software engineering courses). Through the course, students can also explore new opportunities generated by these devices, such as location-aware services and remote sensing, in developing new IoT applications.

The learning outcomes of the course are as follows (ranked according to the levels of complexity and specificity mentioned in Bloom's taxonomy (Bloom's, 1965)):

- Identify issues such as processing power and memory constraints, security, coordination and synchronisation, etc. in application development.
- Comprehend the concept of IoT.
- Apply programming principles and techniques in software development for portable or constrained devices.
- Develop applications on a portable or constrained device such as Raspberry Pi.
- Develop network applications with multiple machines.
- Develop applications that incorporate sensors or other measurement devices.

The course was offered to students who had taken at least two programming courses, such that students were assumed to be (1) familiar with writing basic programs; and (2) comfortable with enhancing sample programs. However, students might have limited exposure to more advanced programming topics such as space and running time analysis, network programming, program parallelisation, network security, data collection, distributed algorithms, etc. Most of them did not have programming experience on portable or constrained devices, such as Raspberry Pi.

The class schedule of the course is shown in Appendix. Apart from 149 minutes (2.48 hours) of online video lecture content, there were in total 43 hours of face-to-face contact hours, which include 26 hours of laboratory activities, and 17 hours of other in-class activities. The laboratory activities, which will be discussed in Sub-section 3.3, were designed to allow students to gain hands-on experience with hardware assembly and programming. On the other hand, the 17-hour in-class activities include guest lectures, tutorials taught based on students' feedback on the video lectures, project consultation and presentation sessions, and written examinations. It is believed that students spent similar hours for studying the course, since original lecturing hours had been used for both laboratory sessions and supplementary in-class learning activities. Discussions of learner-content interactions and learner-instructor interactions can be founded in Lei et al. (2017a).

Final score of the course is contributed by the following activities: i) final exam: 40%, ii) individual course project: 30%, iii) laboratory exercises: 15%; iv) in-class mid-term paper test: 10%, and v) questions related to lecture videos: 5%.

2.2 Introducing authentic design experience via flipped classroom

A well-designed programming course should help students develop the ability to (1) write programs in a particular language paradigm; (2) acquire new knowledge through self-learning; and (3) identify opportunities in new technologies via programming.

However, in traditional programming curricula, regular lectures mainly focus on illustrating concepts and demonstrating sample programs. In tutorials, due to time constraints, very often teachers can only briefly explain the programming exercises, and provide limited coaching on programming. As a result, students often learn passively with limited opportunities for in-depth reflections in class.

The traditional lecture model also often fails to cater for individual students' learning needs. For example, students with more substantial prior knowledge may only need to learn a portion of a lecture, while weaker students may need more time to digest a topic before progressing to the next one. As a result, stronger students may lose their interests, while weaker students may not receive adequate support in traditional lectures. Weaker students could only manage to understand the code examples and apply them to very similar situations. Furthermore, weaker students may not be aware of common mistakes and may not be skilful enough in debugging, as they may not be able to fully comprehend error messages, isolate problematic code fragments, and analyse program output.

The flipped classroom approach allows students to watch video lectures anytime, anywhere, at their own pace. Although it requires students to spend time on watching videos before class, this approach caters for students with different learning needs. It also allows teachers to conduct intensive, interactive and engaging in-class activities, such as hands-on programming exercises, and act as facilitators. Through these activities, students can thoroughly and actively examine their understanding, discuss with teachers and peer on challenges encountered in the exercises, and sharpen their high-order thinking skills (e.g., evaluation, synthesis, and analysis).

3 Instructional videos and learning activities

Since the course was flipped, students were expected to watch videos before class to learn basic concepts on IoT and related topics. Meanwhile, Raspberry Pi and Sense HAT were used to teach IoT devices in weekly face-to-face laboratory sessions, with supervision from the course team. By the end of the semester, each student had to propose an IoT application and develop a prototype, as the course project.

3.1 Instructional videos with assessments

Instructional video should be effective for digesting, such that students can learn through videos. In other words, videos should be not difficult to comprehend. Ineffective videos may demotivate students from watching the entire video or a series of instructional videos (Lei et al., 2017b).

The video series covered advanced programming topics, with each video focusing on a particular topic. Examples of topics include computational power and memory constraints of IoT devices, as well as common issues like localisation and time synchronisation. The content originally delivered in lectures were presented in a series of short videos (around eight minutes for each video) to keep students engaged. Usually, one video was used for illustrating concepts, supplemented by two videos focusing on clarifying misconceptions and explaining examples. Students had to watch at most four videos each week, at their own pace. In order to ensure effective visual presentation of information in the videos, we have adopted research-based principles for designing multimedia instruction, which are based on the cognitive theory of multimedia learning (Clark and Mayer, 2016) as shown in Table 1.

Table 1 Principles for designing multimedia instruction

Reduce extraneous processing
Coherence: Delete extraneous materials
Signalling: Highlight essential materials
Redundancy: Do not add on-screen captions to narrated graphics
Spatial contiguity: Place printed words near corresponding graphics
Temporal contiguity: Present spoken words simultaneously with corresponding graphics
Managing essential processing
Segmenting: Break lessons into learner-paced parts
Pre-training: Present characteristics of key concepts before lessons
Modality: Use spoken words rather than printed words
Fostering generative processing
Personalisation: Put words in conversational style rather than formal style
Voice: Put words in human voice rather than machine voice
Embodiment: Have on-screen agent use human-like gestures and movements
Image: Do not necessarily put static image of agent on the screen

3.2 Individual readiness assessment test

Students were encouraged to get prepared for the laboratory sessions by watching the videos. In order to motivate them to do so, quizzes associated with each topic was prepared as a form of extrinsic motivation. The questions required students to run sample programs, write short code fragments, and identify misconceptions related to the topic. These questions were designed to assess students' higher-order thinking skills. The quiz scores accounted for 5% of the final grade. Examples of questions are shown in Table 2.

Table 2 Examples of questions for individual readiness assessment test

Topic	*Question*	*Thinking skill involved*
Merge sort	Consider the following codes. According to the implementation in Slide 5, how many “first” and “second” arrays will be created and what are their sizes? (*Applying*) int[] list = {7, 4, 8, 1}; mergeSort(list);	Applying
Socket programming	Suppose you want to develop a sorting server to help clients to sort large arrays. Suggest a feasible port number for your server.	Analysing
Socket programming	As more and more clients are requesting your service, you want to setup one more server, that is, there are two machines running the same sorting server program. Can you use the same port number for the two servers? Explain your answer.	Evaluating
TCP and UDP	In each of the following cases, suggest whether TCP or UDP should be used. Explain your answers. 1 Request the current time from a time server. 2 Remote login a machine. 3 Send an urgent message that requires immediate attention (such as reporting a fire).	Evaluating
Java threading	Is it necessary to use critical section (synchronised block) for ThreadInterleave.java?	Remembering
Quicksort	Given an integer array (int list[]). Write the Java code fragment that reverses the list in-place.	Creating
HTTP	HTTP/2 allows a server to push contents to a client before a client requests them. Give a scenario that the client would be benefited by this.	Creating

Students’ performance in the quizzes gave the course team an idea about whether they had mastered the concept(s). If many students failed to answer a certain question correctly, the course team would provide supplementary materials to further explain the topic(s) concerned. Meanwhile, through assessments, weaker students could identify their limitations and review the relevant videos on their own. A mid-term test and a final examination were also administered to motivate students to consolidate their subject knowledge. In summary, different stages of assessments had been used to gauge students’ learning and understanding of contents.

3.3 Laboratory programming exercises

Laboratory exercises were designed to serve as the building blocks of developing a complete IoT application. By completing these exercises, students could:

- Experience the functionalities and constraints of IoT devices through hands-on exercises. For example, in topics covering the difference of computational power and memory limitations between IoT devices and PC, the exercises involved explaining actual measurements students made. This was designed to enable students to understand concepts illustrated in videos in a more tangible way.

- Learn the roles of different protocols and software components in practical IoT solutions. For instance, some laboratory sessions involved discussing the implications of using transport-layer network protocols of TCP and UDP, as well as comparing between application-layer protocols of HTTP and MQTT. These sessions supplemented the theoretical discussion in videos through introducing the industry-standard implementation utilising available software libraries. These exercises also helped students create some building blocks that might be useful for their projects.
- Develop an idea of how to architect and document software projects, as well as using Git version control system.

Programming exercises were also student-paced: In the early stage, some sample codes had been provided as reference. In later stages, students needed to self-learn using Application Programming Interface (API) through reading online documentation. They had to submit their completed work, including codes and written explanations they wrote, a week after a laboratory session. Stronger students could look for external codes to enrich the functionality. Meanwhile, weaker students could work on the laboratory exercises and learn higher-order thinking skills under the guidance of the course team, and through discussion with their peers. Students could hardly learn these skills efficiently if they only attend lectures and read textbooks by themselves.

3.4 IoT design projects

In the project, each student was required to identify an IoT application that involves three (or more) Raspberry Pi's with sensing add-ons, and deliver a prototype at the end of the semester. The project was divided into several assessment phases, as shown in Table 3. Requirements of the project are as follows:

- The purpose of the application could address a social or environmental problem, or create business values with the IoT paradigm. The proposed solution should analyse or visualise acquired data, generate insights, and eventually provide benefits to individuals, businesses or the society.
- Autonomous sensors or human input (e.g., buttons or GUI) should be used as the IoT system input. The input data collected should be forwarded to a server for storage and analysis. An actuator network should be used to directly manipulate or adjust the environment. Meanwhile, a web interface or GUI should be used to provide human-perceivable insights and information.
- The designed system architecture of the proposed IoT application should clearly specify how components and devices interconnect and interact with each other. Students have to apply theories about IP networks illustrated in hands-on laboratory exercises into the system design.
- The system prototype could be developed through Raspberry Pi, Sense HAT, ThingSpeak (a data visualisation service), and other hardware and software components if needed. For example, even though Arduino, various sensors, Python GUI, and Telegram bot were not covered in the course, students were supported to self-learn and integrate these components into their projects.

Table 3 Assessment for the course project

Week	*Assessment items*	*Grading*
9	Project proposal	10%
11	System architecture	30%
14	Project proposal presentation	15%
18	Project demonstration	20%
18	Final report	25%

Students were required to submit a one-page project proposal describing the IoT application they wanted to develop in Week 9. Two regular face-to-face project consultation sessions had been arranged for students to discuss their proposals with the course team individually. Students could further arrange consultation sessions with the instructor and the TA during office hours by appointment. During these sessions, students asked the TA specific questions regarding the technical implementation of the projects. The students then had to design the system architecture of the application in two weeks. This allowed the course team to understand whether each student was on the right track in developing the application. Students could also request for extra equipment according to the system architecture.

In order to help students gain entrepreneurship and product pitching experience, a mock "fund-raising" exercise was arranged at the end of the course. Students had to (1) present their project proposals and pitch their products as if they were founders of start-up companies; and (2) judge their peers' projects as "venture capital investors." Each student was given a certain amount of virtual fund to invest on projects by other students, as a form of peer evaluation for the project and the presentation. After the students finished their examinations, they also had to demonstrate the prototype of the application, and answer questions regarding system design. They then worked on the final report based on the feedback received during the demonstration.

4 Course evaluation

The following evaluation measures had been discussed in Lei et al. (2017a) for illustrating the impact and effectiveness of flipping the course:

- students' feedback on the course from the student survey,
- students' behaviours from the student survey,
- students' performance on quiz and examinations, and
- usage analysis of courseware.

In this section, complementary discussions had been included in the manuscript for guiding readers understanding how the course can be further evaluated:

- student perceptions from the student survey (Sub-section 4.2);
- students' performance on the project (Sub-section 4.1); and
- usage analysis of instructional videos (Sub-section 4.3).

4.1 Students' performance on the projects

In general, students performed well in the projects. First, they were able to propose diverse applications that were beneficial to different groups of users and ultimately our society. This demonstrated that students could identify the potential of the skills they learned. Second, most students were able to develop a prototype that meets the requirements mentioned in Sub-section 3.4. This implies that the compulsory and extensive laboratory exercises were successful in equipping students with the programming skills needed for their projects. Through the course project, we believe students had achieved the learning outcomes, as discussed in Sub-section 2.1.

Two examples of student projects are presented below:

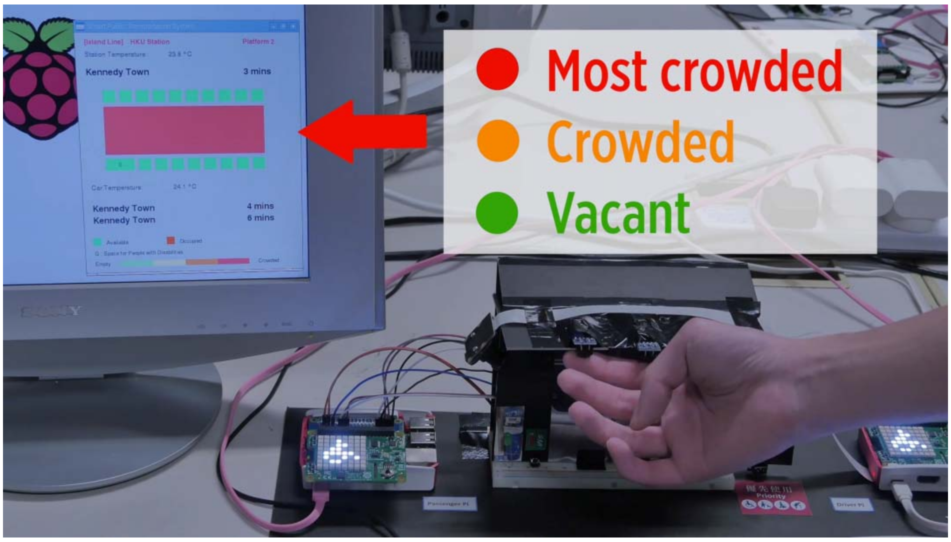
4.1.1 Smart transportation system

An intelligent public transport IoT solution shown in Figure 1 was designed by a student for providing passenger information and assistance (for individuals), crowd and scheduling management (for transport company), and emergency response (for society). Wirelessly networked sensors are installed in the train cabin.

Figure 1 Setup of one of the student projects: smart transportation system

- Infra-red sensors were used to detect seat availability. The vacancy information was then aggregated and shown on the workstation, as depicted in Figure 2.
- In case of emergency, for example if the smoke detector detects a fire, the security camera will be activated so that the staff members will be informed the real-time situation inside the car cabin through the workstation.
- The sensors in the train cabin are networked for notifying the train driver and station server over MQTT protocol. The station server will evaluate the situation and issue alerts if necessary. Python GUI and WhatsApp messaging bot for passenger information are used for reporting.

Figure 2 Interactions with infra-red sensors in the train cabin reflected in the occupancy information displayed on a networked workstation



The student had explored other existing projects in the design stage. In particular, he aimed to identify (1) what they were trying to achieve; (2) how they were designed to address a certain problem; and (3) why IoT systems can fail or why they are not feasible in the real-world environment. This process helped him avoid the potential problem in developing his own project and making the system functional.

4.1.2 Smart planting system

In the second project, the project aims to help people monitor their own crops at home with the help of sensors and actuators, as shown in Figures 3 and 4. The crop will be watered automatically when needed. The device is also connected to the internet, and interfaced with the Telegram instant messaging application. Therefore, users can enquire the status of a crop, search for harvested food, and monitor the planting process remotely through a Telegram messaging bot developed by the student. For example, information about crop growth, such as photos and time-lapse videos, can be displayed via the Telegram messaging application.

The student reported that with the flipped classroom approach, he could learn the course material on his own pace, giving him much more time in the laboratory to work with the devices (e.g., Raspberry Pi and Sense HAT). He also learned practical skills more proactively by developing his own codes.

4.2 Observations of student perceptions from the previous student survey

One observation in the survey is the confidence level of "developing network applications with multiple machines" dropped after students completed the course, compared to the entry survey. We believe that before students practiced networking the machines, they probably considered themselves competent in handling the communication among multiple machines given what they learned from TCP/IP, sockets, as well as the one-to-one client-server sample programs. However, a lot of unexpected and unpredictable situations that were never mentioned in textbooks might have appeared during implementation. This made the students to realise that network applications are not as simple as they perceived. In other words, through working on the laboratory programming exercises, students identified the gap between theory and practice, as well as the limitation of surface learning in the traditional teaching pedagogy.

Figure 3 Raspberry Pi with other hardware peripherals connected to support the smart planting system

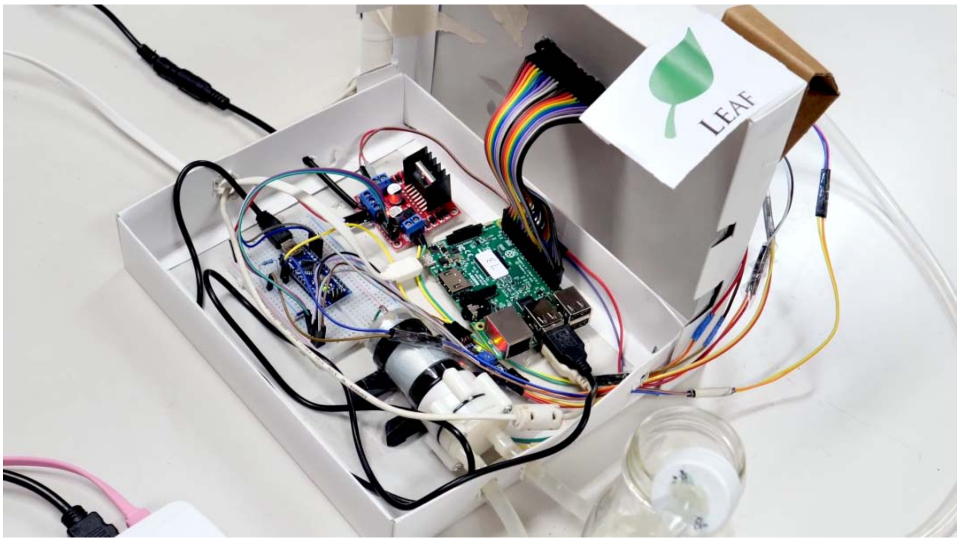
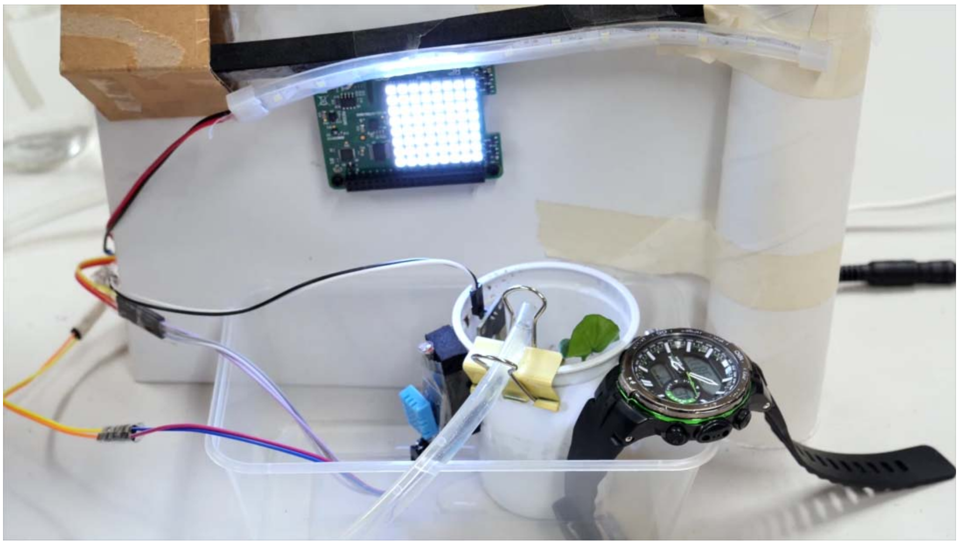


Figure 4 Close-up view of the sensors, lighting, and watering actuators installed on the plant



4.3 Learner browsing behaviour on instructional videos

In this section, we aim at analysing the video usage, i.e., how the courseware had been consumed by learners in terms of the breadth (e.g., frequency of visits) and depth (e.g., duration of each visit) of visits. In 13 teaching weeks, all 14 learners accessed the courseware and watched the videos. Several high-level educational insights on instructional videos have been obtained from the data.

4.3.1 Effectiveness of instructional video contents

27 videos had been watched 660 times by 14 students. Each video, with an average video length of 5.52 minutes, had been watched for 3.05 minutes on average. That means learners had watched 55.28% of all the videos' content. Details of learner browsing behaviour of selected videos can be found in Table 4. Educational insights can be obtained from data:

- Students tended to skip theoretical discussions (which could be found in the teaching slide sets for the course). On the other hand, they frequently watched (and re-watched) explanations of mechanisms in examples, which could not be explained easily through the slides, and could usually only be explained once in traditional lectures. This illustrates the benefits of providing instructional videos for self-paced learning.
- Learners were interested to learn more about the sensor networks, as shown by the high average percentage viewed of the video. It is because it is a new topic to students and is related to the course project. Meanwhile, there were a lower number of views and low average percentage viewed of the video on "Java UDP Server and Client" and "Basics of Unix Commands" – students might have learned these topics in previous courses.
- It is important to produce videos covering general concepts (which are more popular) and specific topics (which are less popular). Popular videos are usually those that provide background information for most students (which can be indicated by its high number of views and a low average percentage viewed of the video). Meanwhile, less popular videos usually have a high-retention rate, which indicates a specific group of students are interested in learning or further exploring certain topics. By producing videos covering both general concepts and specific topics, we can fully utilise students' learning time for personalised learning.

Table 4 Learner browsing behaviour of selected videos (14 students)

Video title	*Views*	*Length (minutes)*	*Average percentage viewed*
Multicast: Ordering	42	7.50	34.40
Hypertext Transfer Protocol	40	3.68	41.74
Sensor networks	29	4.83	87.40
Quicksort	23	5.38	79.06
Java UDP server and client	18	5.13	55.64
Basics of Unix commands	12	1.40	46.45
Compiling Java programs	11	5.00	76.44

4.3.2 Identifying ineffective moments of instructional videos

The consumption of every video had been analysed through video retention graphs (Lei et al., 2017b). The video retention indicator shows the number of views for a particular moment of a video as a percentage of the total number of video views. It indicates (1) how well the video engages its audience; and (2) the moments of unusual times of replay, i.e., the moments that learners have difficulties digesting the content. These graphs gave us insights for moments that need instructional revisions for the next cohort (Lei et al., 2017b).

After identifying unusual moments, instructional multimedia design principles described in Table 1 can be used for evaluating videos and generating insights for revisions. For example, moments with extraneous information to be processed and moments that require "Segmenting" had been identified. Problematic moments have been identified in 20 out of 27 videos. These observations give us hints of revising the videos for the next cohort. Examples of problematic moments are as follows:

- Figure 5 shows the moment with atypical retention rate. Learners re-watched the moment because the teacher had not clearly explained the three kinds of notations at the beginning of the example (i.e., concerns of "Pre-training"). Furthermore, the teacher used the same figure for illustrating two different scenarios (i.e., concerns of "Signalling" and "Coherence"). More visual aids can be used for clarification at the beginning.
- Figure 6 shows the moment of atypical browsing behaviour in another video when the teacher used a few words to explain the matrix mapping mechanism (i.e., concerns of "Pre-training"). As a result, students had to re-watch the moment to understand the mechanism.

Figure 5 A lecture video with atypical browsing behaviours: The retention rate of the video against time, and the moment illustrating the concept of ordering of nodes, in 4:00 (i.e., the moment indicated by a red vertical line)

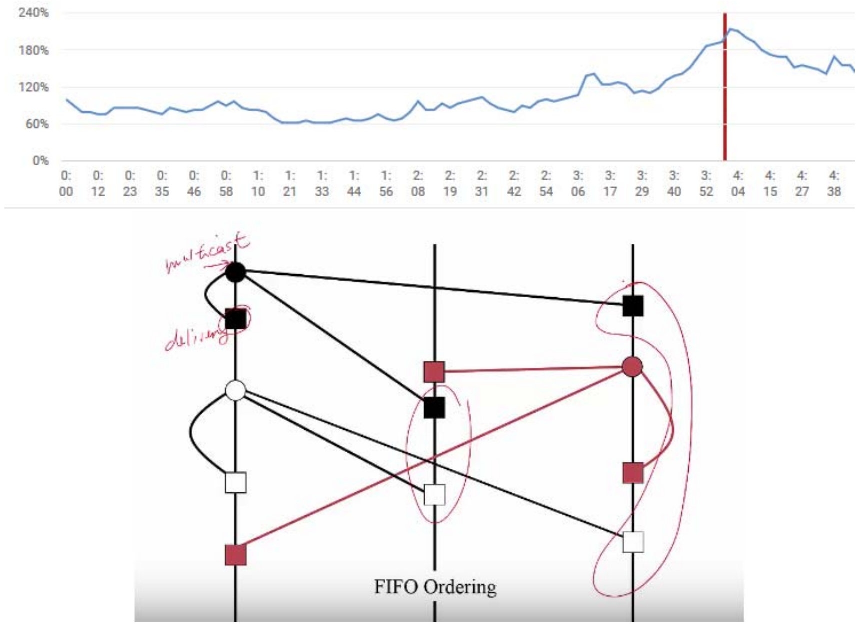
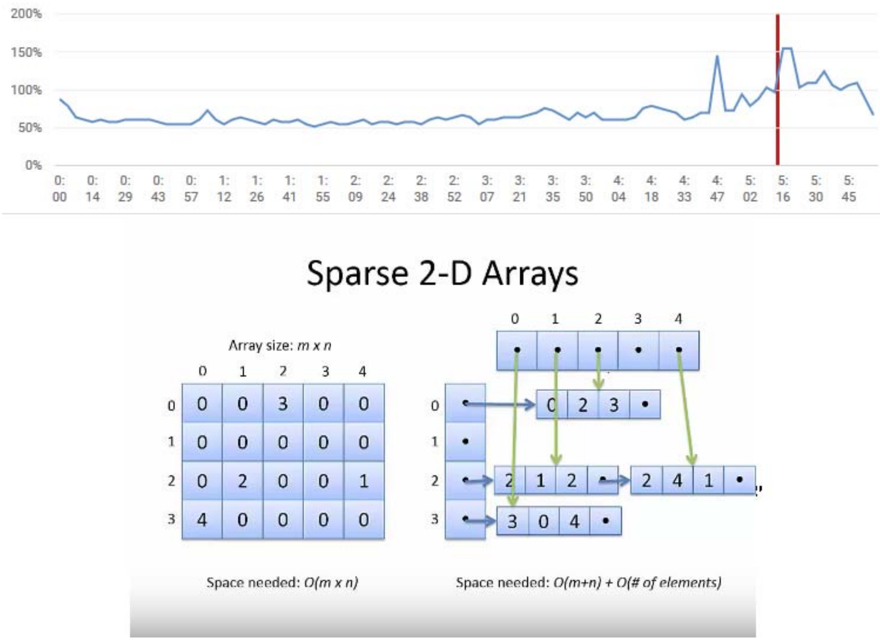


Figure 6 A lecture video with atypical browsing behaviours: The retention rate of the video against time, and the moment with illustrating mapping of the sparse 2-D arrays, in 5:15 (i.e., the moment indicated by a red vertical line)



5 Future development for scaling up learning

5.1 Seamless flipped learning experiences through mobile learning

Although online courseware and short lecture videos can be watched anytime and anywhere using mobile devices, we noticed the online learning module is not well connected with face-to-face activities. In other words, we have not fully explored how mobile devices can be used for learning both in the classroom and beyond the classroom. Researchers have investigated principles and strategies for K-12 learning (Hwang et al., 2015), but not higher education institutions. As our course targets on university students, students are more capable of handling challenging tasks. For example, the pitching presentation can be partially replaced by producing a pitching video, a product demonstration video and a "making of" portfolio video. Through making videos, students can acquire pitching and digital literacy skills. Students can also document and reflect their development cycle, as well as learn to evaluate the product from a user perspective, for consolidating their learning. Furthermore, we should also explore how applications and learning management systems should be designed for contents, such that learning can be done in-class or out-of-class (Boticki et al., 2012). Learning systems can also be further enhanced for personalised learning (Eap et al., 2009) or mobile learning in an higher education setting (Kranz et al., 2013).

5.2 Enhancing motivations to complete the course

Course grades had been used as an extrinsic motivator for students. In future cohorts, the overall course structure and in-class activities can be designed based on the Self-Determination Theory (Niemiec and Ryan, 2009; Gagné and Deci, 2005), so as to enhance students' intrinsic motivation to complete the course.

One way to build students' intrinsic motivation is to improve social interaction, by arranging a coding competition for them in laboratory sessions and inviting them to compete in completing exercises against each other in teams. To ensure even participation among all students, they have to take turns to solve problems. This will encourage peer interactions, which will generate stronger motivations in problem solving.

Multistage achievement and tangible product development in the course are also recommended to motivate students to persist until the end of the course. A sense of achievement can be easily created if students are given the opportunity to solve a problem, turn in an artefact or a product by the end of the course. Their competence can be further enhanced by introducing multistage challenges and a final win state. For example, game elements (e.g., badge systems (Lei et al., 2018), learning progress dashboard (Hu et al., 2017)) can be introduced to cultivate motivation in students' learning. These measures give students a feeling that they are making achievements progressively. These elements help teachers and students (self-) monitor and (self-) assess students' learning progress.

5.1 Holistic and consolidated assessments and feedback

One possible way to consolidate students' learning is to include lesson reflection or debriefing activities after each activity and a holistic post-project debriefing. Furthermore, interactive assessment tools (e.g., Mentimeter) can be introduced to facilitate quick in-class reflections. Meanwhile, code autograder and self- or peer-evaluation mechanisms can be introduced to provide prompt feedback on outcomes and behaviour, respectively. Learning activities, learning outcomes and assessments should also be aligned to ensure that feedback can be generated in a systematic manner.

5.4 Partial flipping for effective learning of higher-order thinking skills

In flipping the course, we noticed that an effective implementation of flipped classroom can provide extra classroom time and opportunities for students to consolidate their learning of practical skills and higher-order thinking skills (e.g., analysing, evaluating and synthesising) under the teacher's in-class supervision. This greatly assists students' learning since these skills are often difficult to self-master without adequate practices and teacher support. We also noticed that in this initial stage of transformation, it is more effective to partially flip the course rather than flipping the whole course. Based on the experience we gained in the study, classical content and theoretical concepts are less likely to change overtime, hence more suitable to be transformed into videos first. Furthermore, partially flipping the course minimises potential resistance from students as it may take time to convince some students take on more responsibility for their own learning (Talbert, 2012). Therefore, in future studies, we would like to facilitate more teachers in partially flipping their courses, with a focus on scaling up students' learning of hand-on skills and higher-order thinking skills, using pedagogical framework such as the Seven Effective Teaching Principles (Luo et al., 2017).

5.5 Analysing and designing instructional videos for engaging and effective learning experience

More and more teachers now show growing interest in producing instructional videos. However, they often encounter difficulties in producing videos by themselves. In particular, they do not know how to produce effective instructional videos. In this project, video analytic tools (Lei et al., 2017b) discussed in Sub-section 4.3.2 have been used to evaluate whether the videos were (not) designed according to research-proven multimedia instruction principles, as shown in Table I. We believe more cloud computational linguistic tools can be used for analysing the adoption of principles for video production. For example, the "Modality" and "Personalisation" concerns can be identified via computational linguistic analysis tools, such as IBM Watson Tone analyser or Google Cloud Platform Cloud Natural Language tool. Meanwhile, "Embodiment" and "Image" concerns can be identified via Google Cloud Platform Cloud Video Intelligence tool. In the future, we hope to propose an analytic-supported mechanism for producing instructional videos, which are essential for creating engaging and effective online or blended learning experiences.

6 Conclusions

This study covered the design of the instructional videos and classwork activities for a flipped engineering course. The design rationale of the course structure, instructional videos, and learning activities had been discussed. Students' video viewing behaviour and their design projects were analysed to evaluate the effectiveness of this flipped classroom pedagogy. Through this study, we found that compulsory and extensive laboratory exercises were effective in equipping students with the programming skills needed for prototyping practical applications. Meanwhile, we found that a specific group of students are interested in learning or further exploring certain topics. Therefore, we can fully utilise students' learning time for personalised learning, by producing videos for both general concepts and specialised topics. We noticed that in this initial stage of transformation, it is more effective to partially flip the course rather than flipping the whole course. For example, classical content and theoretical concepts are more suitable to be transformed into videos first. Furthermore, partially flipping the course minimises potential resistance from students as it may take time to convince some students take on more responsibility for their own learning.

Videos outlining this initiative can be found in https://hku.to/engineering_elearning.

Acknowledgement

The authors would like to thank Ms. Emily Leung for giving pedagogical and editorial advice during the manuscript development. This work is supported in part by the University of Hong Kong Teaching Development Grants Award (Project No. 16/602).

References

Atzori, L., Iera, A. and Morabito, G. (2010) 'The internet of things: a survey', *Computer Networks*, Vol. 54, No. 15, pp.2787–2805.

Bloom's, T.M.E. (1965) *Bloom's Taxonomy of Educational Objectives*, Longman.

Boticki, I., Wong, L.H. and Looi, C.K. (2012) 'Experiences in implementing and using a technological framework for mobile collaborative learning of mathematics and Chinese', *International Journal of Mobile Learning and Organisation*, Vol. 6, No. 1, pp.79–93.

Clark, R.C. and Mayer, R.E. (2016) *E-learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*, John Wiley & Sons.

DeLozier, S.J. and Rhodes, M.G. (2017) 'Flipped classrooms: a review of key ideas and recommendations for practice', *Educational Psychology Review*, Vol. 29, No. 1, pp.141–151.

Eap, T.M., Gasevic, D. and Lin, F. (2009) 'Personalised mobile learning content delivery: a learner centric approach', *International Journal of Mobile Learning and Organisation*, Vol. 3, No. 1, pp.84–101.

Gagné, M. and Deci, E.L. (2005) 'Self-determination theory and work motivation', *Journal of Organizational Behavior*, Vol. 26, No. 4, pp.331–362.

Garcia, I.A. and Cano, E.M. (2014) 'Designing and implementing a constructionist approach for improving the teaching–learning process in the embedded systems and wireless communications areas', *Computer Applications in Engineering Education*, Vol. 22, No. 3, pp.481–493.

Giannakos, M.N., Krogstie, J. and Chrisochoides, N. (2014) 'Reviewing the flipped classroom research: Reflections for computer science education', In *Proceedings of The Computer Science Education Research Conference*, ACM, pp.23–29.

Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013) 'Internet of things (IoT): a vision, architectural elements, and future directions', *Future Generation Computer Systems*, Vol. 29, No. 7, pp.1645–1660.

Hu, X., Hou, X., Lei, C.U., Yang, C. and Ng, J. (2017) 'An outcome-based dashboard for moodle and Open edX', *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, ACM, pp.604–605.

Hwang, G.J., Lai, C.L. and Wang, S.Y. (2015) 'Seamless flipped learning: a mobile technology-enhanced flipped classroom with effective learning strategies', *Journal of Computers in Education*, Vol. 2, No. 4, pp.449–473.

Jonsson, H. (2015) 'Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course', *Proceedings of the Frontiers in Education Conference (FIE)*, IEEE, pp.1–9.

Kranz, M., Möller, A., Diewald, S., Roalter, L., Beege, B., Meyer, B.E. and Hendrich, A. (2013) 'Mobile and contextual learning: a case study on mobile didactics in teaching and education', *International Journal of Mobile Learning and Organisation*, Vol. 7, No. 2, pp.113–139.

Lei, C.U., Gonda, D., Hou, X., Oh, E., Qi, X., Kwok, T.T., Au Yeung, Y.C. and Lau, R. (2017b) 'Data-assisted instructional video revision via course-level exploratory video retention analysis', *Proceedings of the Seventh International Learning Analytics and Knowledge Conference*, ACM, pp.554–555.

Lei, C.U., Hou, X., Gonda, D. and Hu, X. (2018) ''Evaluating the adoption of a badge system based on seven principles of effective teaching', *Proceedings of the 8th International Learning Analytics and Knowledge Conference*, pp.55–62.

Lei, C.U., Yau, C.W., Lui, K.S., Yum, P., Tam, V., Yuen, A.H.K. and Lam, E.Y. (2017a) 'Teaching internet of things: enhancing learning efficiency via full-semester flipped classroom', *Proceedings of the IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, IEEE, pp.56–60.

Luo, C., Hew, T.K., Lei, C. U. and Oh, E. (2017) 'Principle-guided flipped classroom implementation framework for teaching technological contents', *Proceedings of the IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, IEEE, pp.449–456.

Maher, M.L., Latulipe, C., Lipford, H. and Rorrer, A. (2015) 'Flipped classroom strategies for CS education', *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ACM, pp.218–223.

Niemiec, C.P. and Ryan, R.M. (2009) 'Autonomy, competence, and relatedness in the classroom: Applying self-determination theory to educational practice', *School Field*, Vol. 7, No. 2, pp.133–144.

Talbert, R. (2012) 'Inverted classroom', *Colleagues*, Vol. 9, No. 1, p.7.

Tanguma, J., Liu, L.C. and Koong, K.S. (2008) 'Hybrid learning and effects on student assessment outcomes', *International Journal of Mobile Learning and Organisation*, Vol. 2, No. 1, pp.36–47.

Yau, C.W., Kwok, T.T.O., Lei, C.U. and Kwok, Y.K. (2018) 'Energy harvesting in internet of things', *Internet of Everything*, Springer, Singapore, pp.35–79.

Appendix Schedule of class activities, laboratory activities and assessment

Teaching week	*Class activities (1 hour each)*	*Laboratory activities (1–2 hours each)*	*Assessment [Due date]*
1	Course overview	Raspberry Pi and GitHub	
		Merge sort	Merge sort
2		Threading in java	Processes and threads
			Threading in java
3		Threaded merge sort	
4		Quicksort	Quicksort
		Remote merge sort	Client-server model and socket programming
5		Sense HAT	Sensor networks
6		Sparse matrix	
7	Project consultation		
8	Project consultation		
9	Revision for mid-term test		Project proposal
10	Designing IoT applications	UDP	UDP
10	Mid-term test	HTTP	HTTP
		MQTT	MQTT
11		Network configuration	Project system architecture
			NAT
12	Entrepreneurship	Security	
13	Cloud computing and data analytics		
13	Final revision		
14*	Project presentation		
18^	Final examination (3 hours) Project demonstration (4 hours)		Final report

Note: *Teaching week 14: Revision week; ^ Teaching week 18: Assessment week