IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal
IEEE EDUCATION SOCIETY SECTION

Received February 21, 2021, accepted March 7, 2021, date of publication March 10, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065248

Active Distance Learning of Embedded Systems

ABDULHADI SHOUFAN
Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, United Arab Emirates
e-mail: abdulhadi.shoufan@ku.ac.ae

ABSTRACT The move from face-to-face to distance learning poses a challenge for courses that rely on hands-on experience such as embedded systems. In this course, students need to work with hardware and software to achieve various learning objectives. For full advantage, the hands-on experience should be aligned with the acquisition of related concepts and procedural knowledge. The alignment of conceptual learning with hands-on experience is a big challenge, in general, and for distance learning, in particular. This article describes how different learning technologies can be integrated to achieve such alignment for embedded systems in a distance learning mode. A framework for active, lecture-free learning was established using a learning management system, YouTube, various web resources, a hardware kit, and a software development environment. The learning activities were implemented as ungraded quizzes on Moodle with different types of questions. These include review questions, conceptual questions, procedural questions, brainstorming questions, code analysis questions, and code creation questions. Our students used the provided hardware kit and the software development environment to complete the learning activities throughout the semester without listening to any live or recorded lecture from our end. This instructional design was evaluated by analyzing learning data generated by Moodle as well as self-report data. The results show high student engagement and positive perceptions of the course content and the learning method. We believe that the proposed pedagogical framework of this design is of general value and can be adopted in other engineering courses with similar requirements of hands-on experience in distance learning.

INDEX TERMS Distance learning, embedded systems, learning management systems, student engagement, lecture-free instruction, COVID-19.

I. INTRODUCTION

COVID-19 has caused a dramatic change in education and made distance learning one of the few options for educational institutions to continue their operations [1]–[3]. Many or most faculty and students were forced to rapidly adopt new technologies and ways of teaching and learning they were never used to before. Reports about the transition to distance learning highlight many challenges and obstacles both for faculty and students [4], [5]. These include personal, pedagogical, technical as well as financial and organizational obstacles [6].

The related work on this topic has emphasized the importance of student engagement for successful distance learning that is not about ‘‘moving lectures online and continuing business as usual’’, according to DeFranco and Roschelle [2]. Distance learning is associated with a low level of student engagement [7], [8]. Kaczmarek *et al.* surveyed students and faculty about their perceptions about distance learning in the pandemic [9]. The authors found out that students (i) perceive virtual lectures as less engaging than interactive case-based sessions, (ii) consider the learning of technical concepts through virtual classrooms as hard, and (iii) feel that long classes increase fatigue. Faculty reported difficulties in judging student engagement and understanding because they don't see students' faces. Also, the faculty stated that distance learning makes the explanation of technical concepts difficult [9].

To improve student engagement in distance learning several suggestions were made in the literature. The following recommendations are related to our work:

1) Focusing on active-learning techniques rather than passive methods [10].
2) Diversifying distance learning activities to stimulate student motivation [6].
3) Offering ungraded or anonymous quizzes to improve understanding and engagement [9].
4) Offering more interactive sessions to avoid fatigue [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Mark Lee.

5) Combining synchronous and asynchronous learning formats [9], [10].

Embedded systems are ubiquitous in modern life. Graduates with related skills and competencies are in high demand to develop and support this technology [11]. Universities worldwide have responded to this demand by offering different programs in this area [12]–[15]. The embedded systems' didactics was addressed thoroughly by Grimheden and Toerngren [16]. The authors highlighted that this subject has a thematic identity and functional legitimacy. Thematic identity means that the subject of embedded systems is developed within other disciplinary subjects (e.g. digital design and software engineering) and the specific areas of knowledge in embedded systems arise from the delimitations of these subjects. Functional legitimacy refers to the necessity to acquire functional, i.e., practical skills towards product development. These functional skills are usually not part of traditional lectures and cannot be acquired by reading textbooks [16]. Rather, practical competencies are developed during hands-on exercises, laboratory experiments, and projects which rely on a trial-and-error method for problem-solving [16]–[18]. Knowing this, it should be expected that the unprepared shift to distance learning may have a considerable impact on the quality of teaching embedded systems and practical courses in general [19]. One reason for this is that teaching laboratories are hardly accessible in the pandemic and many universities may not be ready for virtual laboratories. Remote labs have long been utilized to overcome some challenges of traditional hands-on labs such as costs, accessibility, and safety [20]–[22]. Virtual labs for embedded systems were also presented by some authors [23], [24]. Information technology-based solutions for connecting distributed remote labs are available to facilitate global sharing [25]. The pandemic of COVID-19 has attracted considerable attention to remote laboratories [26] with several opportunities and challenges [27], [28].

In this paper, we argue that today's technologies can be deployed to support both conceptual learning and hands-on experience at a distance. For this, we present the design of a course on the fundamentals of embedded systems that we offered at our university for computer engineering senior-level students in Fall 2020. The core aspect of this instructional design is the replacement of lectures with learning activities. These activities combine conceptual knowledge with hands-on exercises using hardware and software. Our students performed these activities at home using a hardware kit that we provided at the beginning of the term and the freely available Arduino IDE. Each activity consists of multiple pages with a question on each page. Using this ungraded quiz format is essential to segment the topic and sequence its parts properly with immediate feedback and scaffolding.

Lecture-free teaching has been addressed in the context of blended learning and flipped-classroom models [29], [30]. Also, complete courses that replace lectures with learning activities were described in the literature [31], [32]. However, the classroom in these contributions still plays a central role. It is not clear whether or how the proposed models would work for full distance learning. Furthermore, none of the related designs combines conceptual learning with hands-on exercises thoroughly in the way presented in this article. The relevance of our contribution comes from a unique combination of the following features:

1) Distance learning using a lecture-free, fully active learning mode.
2) The learning activities integrate conceptual learning with hands-on experience using purposeful hardware/software experiments.
3) The learning activities are implemented using an ungraded multi-page quiz format on Moodle to support relevant pedagogical and cognitive principles such as segmentation, sequencing, and immediate feedback.
4) The learning activities are diversified in terms of question types, hands-on experiments, and used information sources.
5) The learning activities can be completed synchronously or asynchronously.

As can be seen, these features are in line with the above-mentioned recommendations for improving student engagement in distance learning. Our analysis of completion reports, student performance in the learning activities, student perceptions of individual activities, as well as self-report data confirms that the proposed design is associated with high student engagement.

The rest of paper has five sections. Section II describes the instructional design with focus on the creation of learning activities. Section III describes the involved technologies. Section IV outlines the project. Section V presents evaluation results. Section VI discusses the findings and Section VII concludes the paper.

II. INSTRUCTIONAL DESIGN

Embedded Systems at our university is a core course for computer engineering students at the senior level. The students must have taken a course on microprocessors as a prerequisite. For a curricular reason, the course has a lecture module with three credit points only, but no lab. However, the students are required to do a semester project using a microcontroller of their choice or by using embedded kits available in the department labs. The course has the following learning outcomes, which are aligned with the ACM/IEEE Computer Engineering Curriculum to some extent [33]:

1) Define an embedded system and name its main features and applications.
2) Explain the general architecture of an embedded system and analyze its main hardware components.
3) Analyze, develop, and create embedded software programs.
4) Explain the hardware/software interaction in embedded systems.
5) Classify embedded system interfaces and analyze their operation.

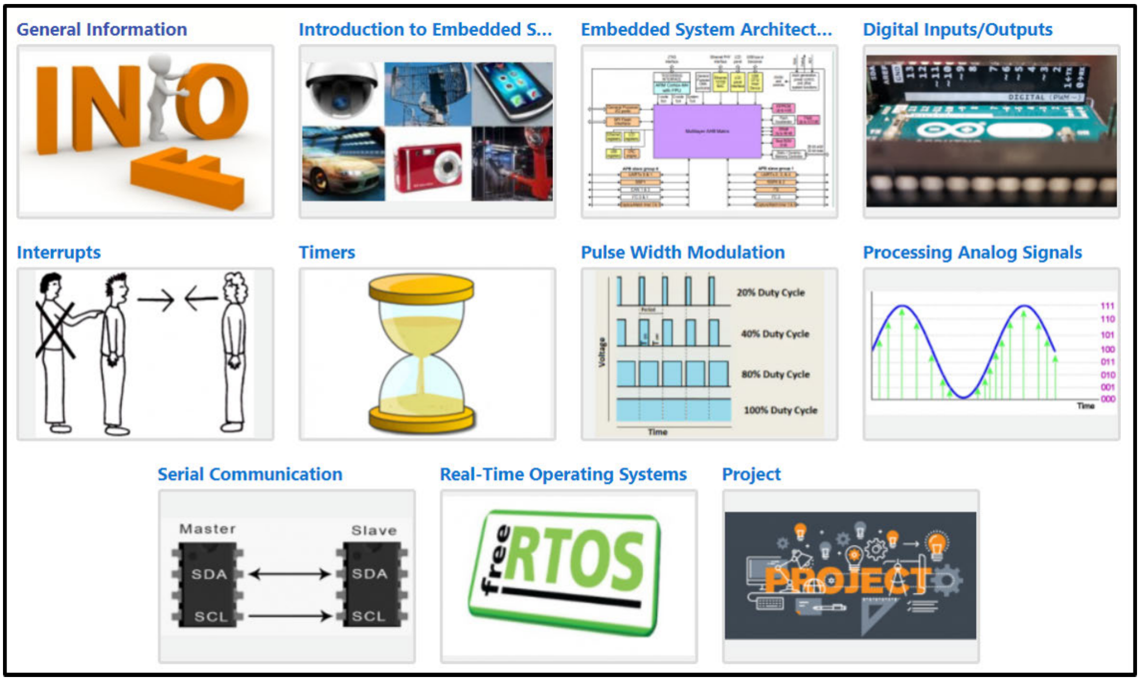


FIGURE 1. Course layout on Moodle.

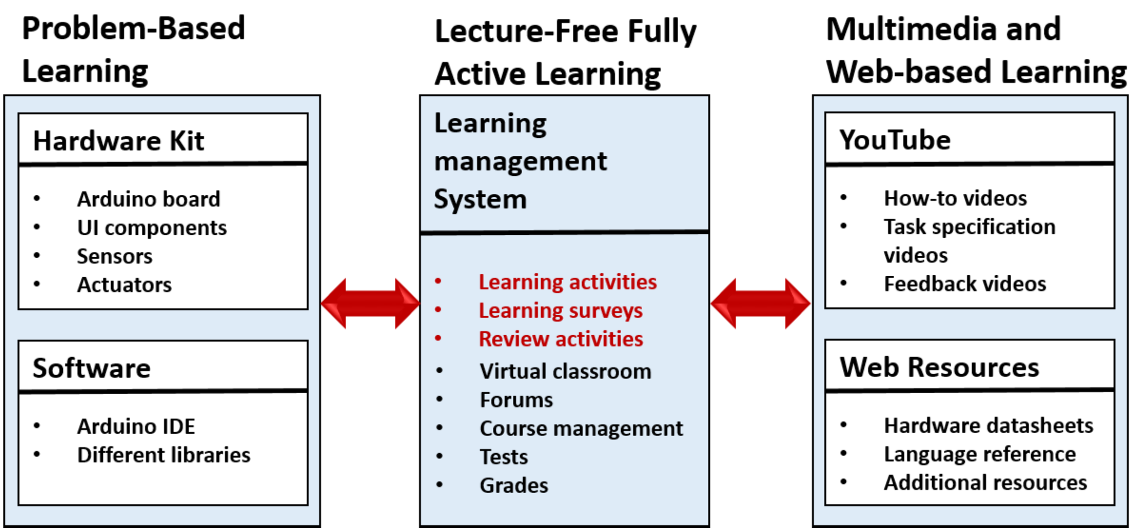


FIGURE 2. Pedagogical framework and supporting technologies.

6) Create embedded programs with a real-time operating system.
7) Design an embedded system for a particular application and assess its functionality.

The course is structured into nine chapters as illustrated in Fig. 1 that shows the course layout on Moodle. Note that the course has a general information chapter and a project chapter.

A. PEDAGOGICAL FRAMEWORK

The course design relies on different pedagogies, which are supported by different technologies as illustrated in Fig.2. The core pedagogy of the presented course is lecture-free active learning. The learning activities are provided through Moodle using the quiz format. Each learning activity (also referred to as *Learning Quiz*) focuses on one or a few learning objectives and replaces a lecture. It consists of multiple pages, where each page contains an introductory part followed by one or more embedded questions as shown in Fig. 3. To answer these questions, students rely on:

1) Knowledge from previous topics in the course.
2) New knowledge presented in the current activity.
3) Knowledge from external sources including YouTube videos, online datasheets, and software manuals.

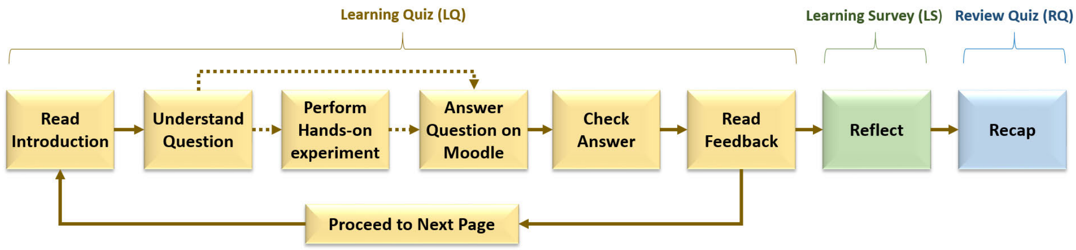


FIGURE 3. The structure of three activities which replace one class hour.

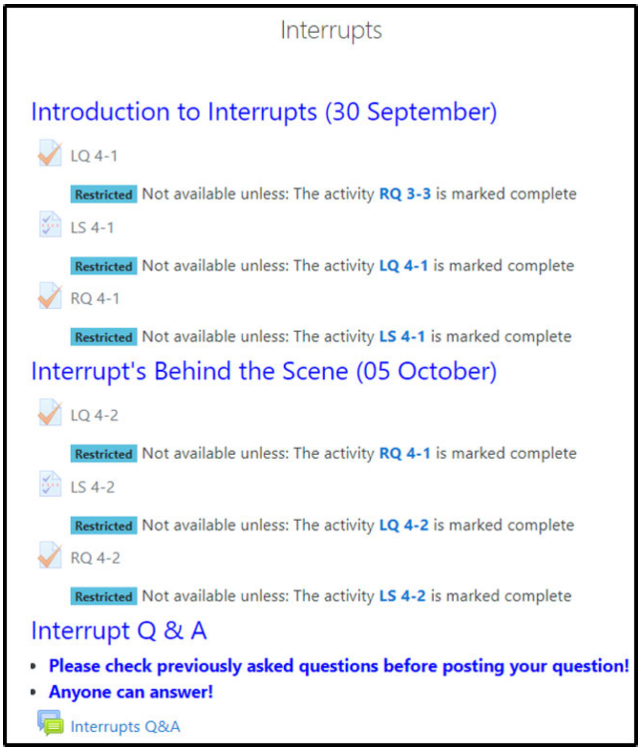


FIGURE 4. The layout of activities within one chapter.

4) Knowledge gained from running hands-on experiments using a provided hardware kit and a software development environment.
5) A combination of the previous four items.

Every learning activity is followed by a one-question survey (referred to as *Learning Survey*) to understand students' perceptions of the completed activity. The learning session is concluded with an ungraded *Review Quiz*, which recaps the learning objective of the current session. Fig. 3 illustrates the structure of these three components.

Fig. 4 shows an example for the layout of activities within one course chapter. Accordingly, the activities are separated by labels which give a title for the current activity and its date. As can be seen, we applied access control rules to make sure that the students have already completed previous activities before starting new ones. This method is highly important for keeping students on track and to be ready for the current activity.

B. LEARNING ACTIVITIES

The design of engaging learning activities is the main facet of this course. A learning activity has two aspects: a structural and a content-specific aspect. Fig. 3 illustrates the *structural aspect* of a learning activity featuring the following pedagogical elements:

1) Splitting the learning topic to multiple segments of knowledge and assigning these segments to successive pages. Segmentation and sequencing are believed to reduce the intrinsic cognitive load in the learner's working memory [34], [35].
2) The student is responsible for the generation of the basic knowledge segment on each page. This is achieved by using an ungraded quizzing approach that strongly relates to inquiry-based learning [36].
3) Providing the right amount of information (or links to information) on each page to avoid the split-attention effect that is believed to increase the extraneous cognitive load in working memory [37].
4) Providing immediate evaluation of students' solutions as well as immediate feedback. The feedback can also serve as scaffolding for the next question if needed. Both, immediate feedback and proper scaffolding are believed to enhance learning considerably [38], [39].

The *content-specific aspect* of a learning activity relies on the instructor's experience to make the activity purposeful, doable, and interesting. The interestingness of the learning activity is very important to maintain students' attention and keep them engaged. The inquiry-based approach (question and answer) provides an initial trigger for engagement. However, to stay engaged throughout the whole activity, students should have a positive experience when they move from one question to another in the session. Therefore it is mandatory to design the activities to be doable but still not trivial. In other words, the activity questions should keep some level of challenge. Even when a student fails to answer a question or solve a problem correctly, the built-in feedback should provide a sufficient explanation of the answer and possibly highlight related misconceptions that may lead to characteristic mistakes.

A typical learning activity in our course has some or all of the following content-related types of questions depending on the specifics of the learning topic at hand:

1) Review questions

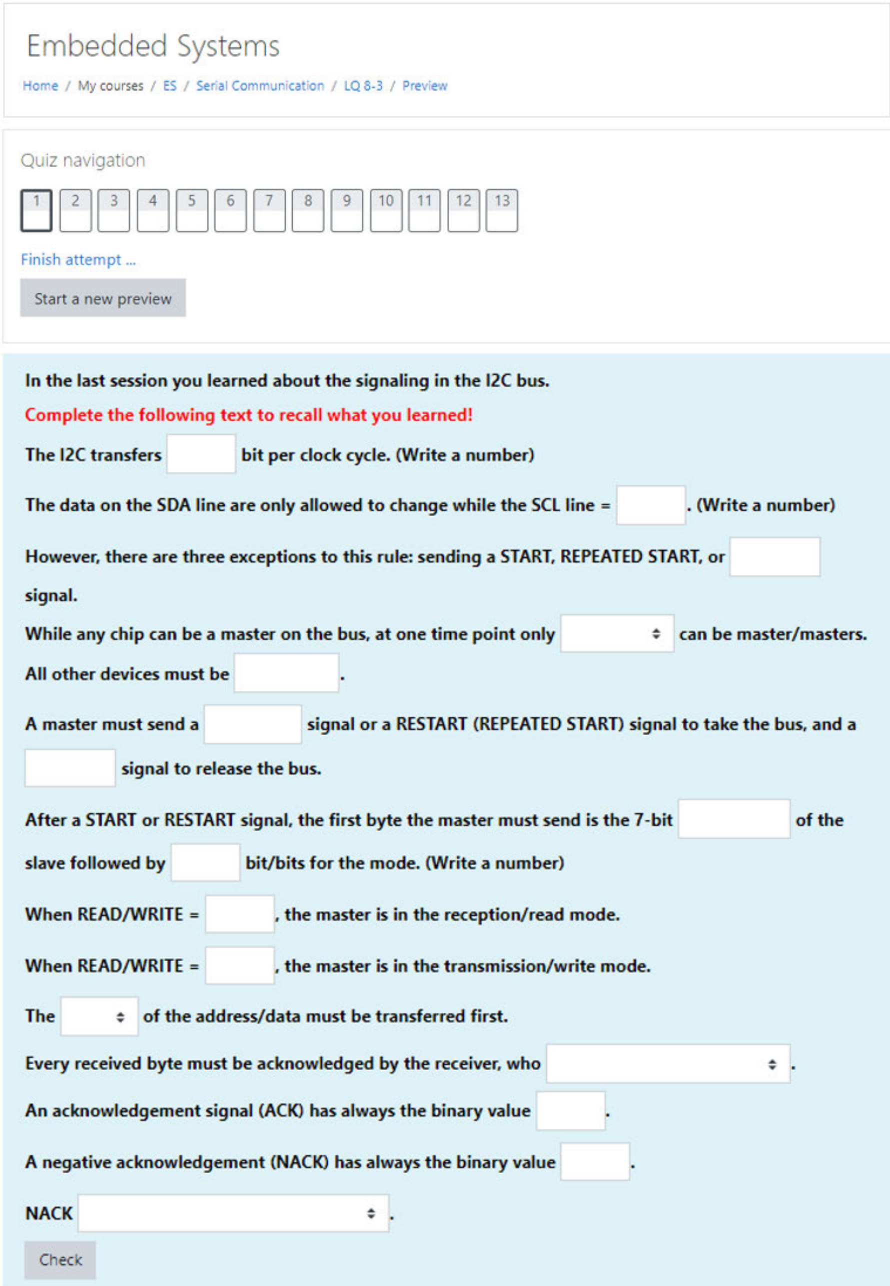


FIGURE 5. An example for a review question.

2) Conceptual questions
3) Procedural questions
4) Brainstorming questions
5) Code analysis questions
6) Coding questions

In the following paragraphs, we explain each of these question types using screenshot examples from the course on Moodle.

1) REVIEW QUESTIONS

Usually, a learning activity starts with a review question to recall what was learned in the previous session and create a link to the current activity. Fig. 5 shows the first question in the activity LQ 8-3 with the chapter about serial communication. Here we are using a question type on Moodle called *cloze*. This question type has a special syntax and allows embedding several short-answer questions, numerical questions, and multiple-choice questions in a single question. For an insight into the syntax of this question, see Appendix A, which presents the question text related to Fig. 5. Note that the abbreviations SA, NM, and MCS in the question text refer to short-answer, numerical, or shuffled multiple-choice questions, respectively.

TABLE 1. Code analysis and code creation questions.

	Software	Function	Property	Hands-on activity	Moodle question
Analysis	Given	Required	X	1. Connect HW if needed 2. Compile and upload SW 3. Analyze function	Multiple choice, numerical, short answer, matching, essay, cloze, etc.
	Given	X	Required	1. Connect HW if needed 2. Compile and upload SW 3. Extract property	Multiple choice, numerical, short answer, matching, essay, cloze, etc.
Design	Required	Given	X	1. Connect HW if needed 2. Complete SW 3. Compile and upload SW 4. Verify function 5. Debug if needed	Cloze essentially
	Required	X	Given	1. Connect HW if needed 2. Complete SW 3. Compile and upload SW 4. Validate property 5. Debug if needed	Cloze essentially

```
const int button_pin = 2;
const int led_pin = 5;

void setup() {
  pinMode(button_pin, INPUT_PULLUP);
  pinMode(led_pin, OUTPUT);
}

void loop() {
  int button = digitalRead(button_pin);

  if (button == LOW) {
    digitalWrite(led_pin, HIGH);
  }
  else {
    digitalWrite(led_pin, LOW);
  }
}
```

Change the mode of the button pin from INPUT_PULLUP to INPUT.

Re-compile and run the code.

What do you observe?

(Watch the linked video in the feedback!)

- ☐ The system behaves the same like the one with INPUT_PULLUP.
- ☐ The LED keeps going on and off arbitrarily when the button is not pressed.
- ☐ The LED goes off when the button is pressed and on when the button is released.

Check

FIGURE 6. An example for a conceptual question.

2) CONCEPTUAL QUESTIONS

A course on embedded systems is rich in concepts related to hardware, software, and hardware/software interfacing. Examples of these concepts include GPIOs, memory maps, multi-function pins, tri-state buffers, master-slave operation, PWM, UART, I2C, and CAN. In the proposed active learning approach, we avoid providing the definitions of new concepts. Instead, we guide the students towards the generation of these definitions. Generating rather than providing a definition is believed to be more effective for conceptual learning [40]. However, we should note that designing questions that help students generate concept definitions can be very time-consuming because it requires making a case for the new concept in the first place. Take the *pull-up resistor* concept as example. This concept is highly relevant for an embedded system to suppress the noise at an input pin or set a pre-defined value for the open-drain operation. Fig. 6 shows how this concept is approached experimentally in the presented course. Note that the feedback to this question includes a video that shows the behavior of the LED when the pull-up resistor is excluded.

3) PROCEDURAL QUESTIONS

Embedded systems require students to learn various procedures, e.g., to convert a high-level program to a register-level program, to replace a polling mechanism by interrupts, to turn a multi-function code to a multi-tasking program using a real-time operating system, to reduce RAM usage by moving data to the program flash or the embedded EEPROM, etc. Procedural learning in the proposed design is facilitated by the ability to segment the procedure into steps and to sequence these steps through multiple procedural questions to reduce complexity. Fig. 7 shows one procedural step towards replacing the call of a library function with a register-level code to control a hardware timer. Note the question how we are making the students aware of the procedure by saying *"In the next questions we will find the addresses and values of all registers which are relevant for transforming the blinker code to the register level"* and *"The value of this register is not yet final because we need to set the bits WGM13 and WGM12 later"*. The final step of this procedure is shown in Fig. 8. This question shows the complete table with the addresses and values of all relevant registers. The students are requested to complete the code with these register addresses and values.

4) BRAINSTORMING QUESTIONS

Sometimes it is helpful to ask students to analyze a problem and to think about a solution even if they don't have a sufficient background. We use this kind of brainstorming to prepare students for the following questions that provide concrete directions towards a working solution.

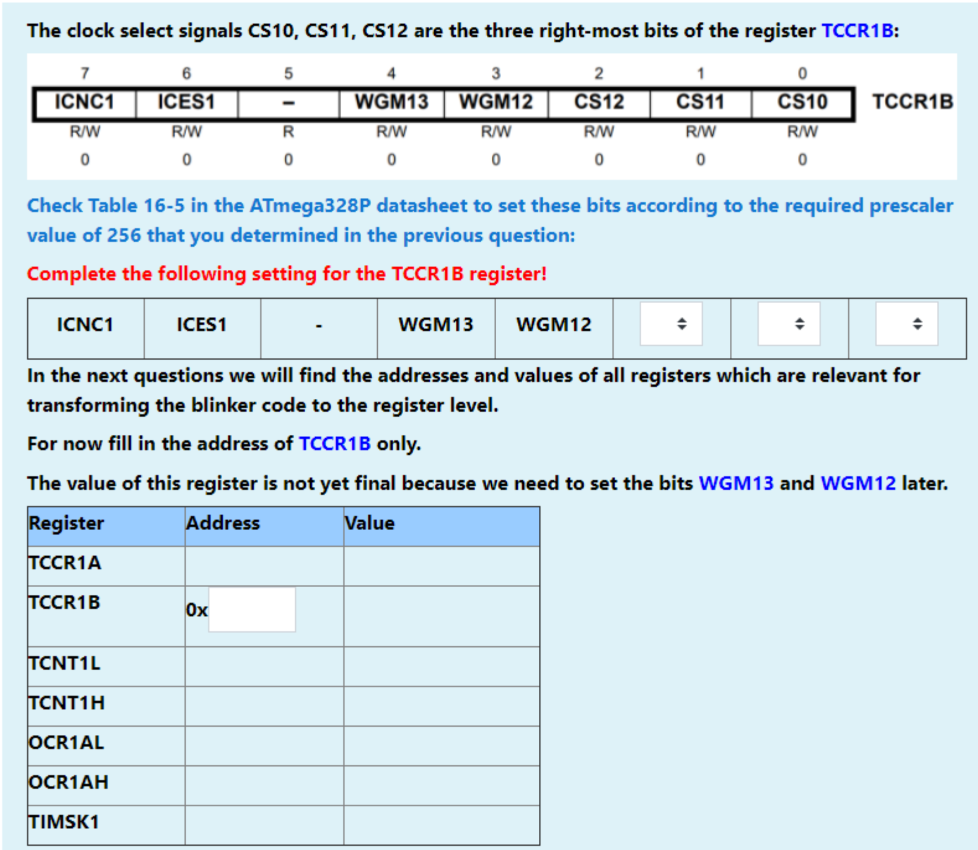


FIGURE 7. An example for a procedural question-A.

Brainstorming questions use the essay question type on Moodle as shown in the example of Fig. 9. Here we motivate the concept of multitasking in real-time operating systems.

5) CODE ANALYSIS QUESTIONS

Working with software is essential for embedded systems. Generally, students need to learn both the analysis of a given code and the creation of new. The learning activities support the development of both skills extensively. Table 1 outlines how this is accomplished in general. As can be seen, in code analysis questions we provide the code and ask the students either to extract the function or to determine some non-functional property such as memory usage. In most cases, we ask the students to support their answers experimentally by running the code on the microcontroller after connecting the required hardware. Fig. 10 shows two examples of code analysis questions. In the left-side question, the students are asked to find out the function of the given code (using an analog signal to control load power) and to identify/recall the control mechanism of pulse width modulation (changing the duty cycle). In the right-side example, the students need to determine the memory usage for the given code as an introduction for memory optimization.

6) CODING QUESTIONS

Design is the ultimate goal of learning embedded systems. Projects are important to enhance students' coding skills. However, to understand embedded systems' concepts, it is important to learn how to encode these concepts in software. Therefore, we attach great value to including code design questions to the learning activity as far as possible. For this, we use the question type cloze on Moodle and we ask the students to complete a code rather than writing it from scratch. Code completion has the following advantages over code creation from scratch:

1) It helps students focus on the language syntax and semantics relevant to the learning objective at hand.
2) It saves time because students don't need to rewrite code already-learned parts.
3) It allows for automatic grading and immediate feedback since Moodle cannot grade entire codes automatically.

Fig. 10 shows two short examples of coding questions. In the left-side question, the students are requested to study the syntax of the function analogReference on the Arduino's website and to complete the function call accordingly. In the right-side example, the students need to learn the syntax of the modifier PROGMEM, its related utility pgm_read_byte(),

FIGURE 8. An example for a procedural question-B.

and the macro F to optimize the SRAM usage by shifting data to the code Flash.

We should note that some related work on teaching programming has proposed using the cloze question type for assessment as an alternative to writing complete codes [41].

III. SUPPORTING TECHNOLOGIES

The pedagogical framework of the presented design is enabled by different technologies as illustrated in Fig. 2. In this section, we describe the role of each technology and how it is used in our course.

A. LEARNING MANAGEMENT SYSTEMS

Moodle was used to accommodate the learning activities. In the background of these activities is a question bank that is structured similarly to the course chapters and sections using question categories and subcategories. Moodle supports many built-in or plugged-in question types that can be used purposefully. In our course, we used ten question types: cloze, drop into text, drop onto image, essay, matching, multiple-choice, numerical, select missing words, short answer, and true/false. The examples described in the previous sections illustrate how to use some of these question types.

As illustrated in Fig. 3 and Fig. 4, each learning activity is followed by a learning survey. The learning survey has a *single* question as shown in Fig. 12 and aims to solicit students' perceptions of the completed activities for reflection and future improvement. In Section V we will present the results of these surveys.

After the learning survey, the students take a short ungraded review quiz in the same format as the learning activity. The review quiz is limited to one or two questions that summarize the essence of the learning activity.

In addition to learning activities, learning surveys, and review quizzes, each chapter includes a Q&A forum. Students can use this forum to post and answer questions related to the respective chapter, see Fig. 4.

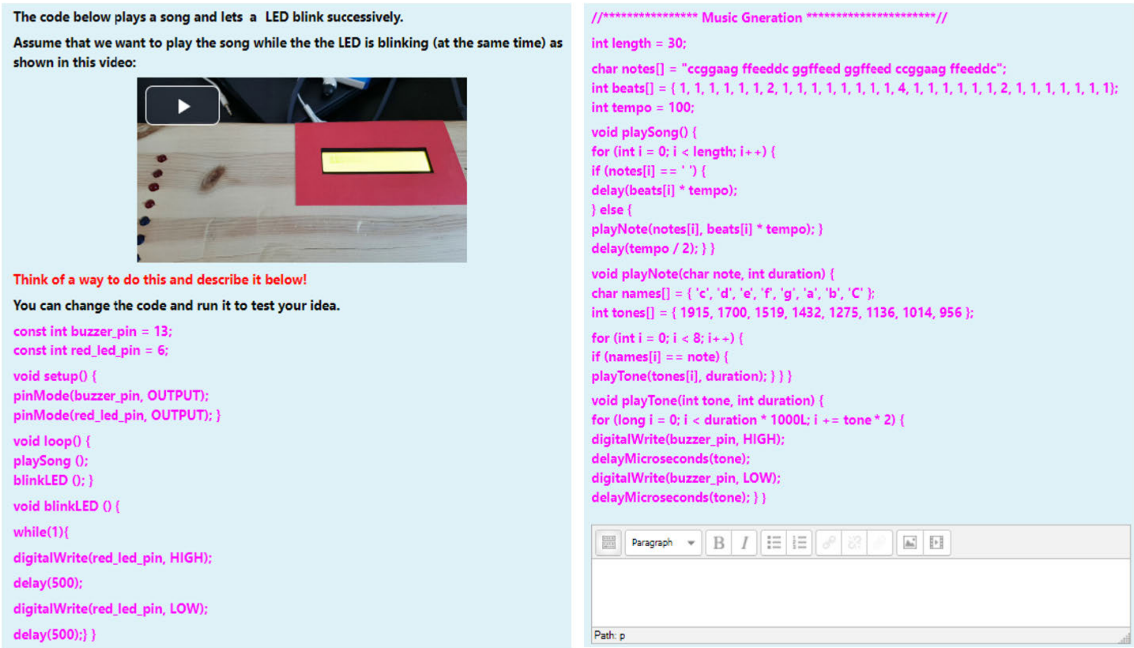


FIGURE 9. An example for brainstorming question.

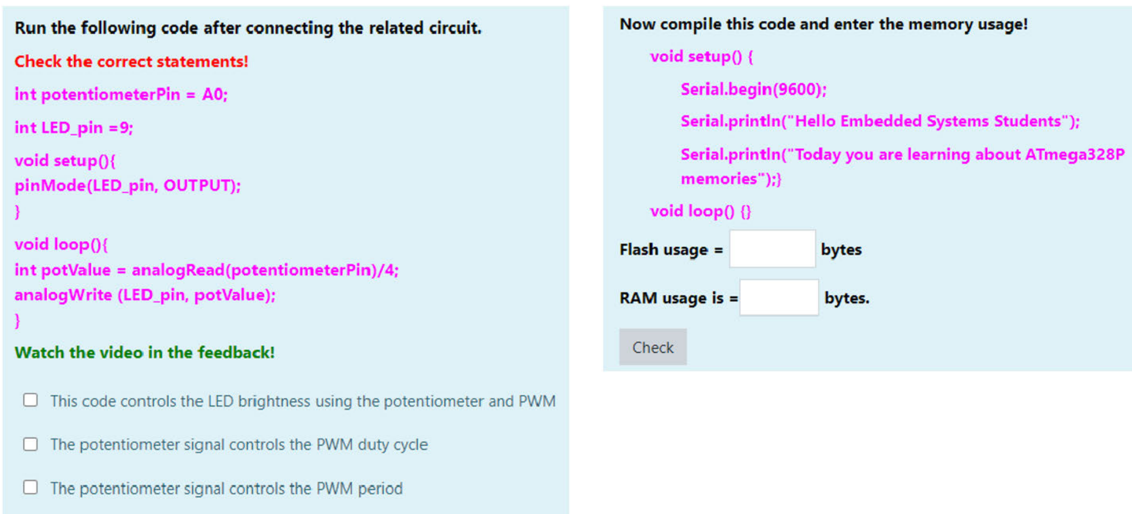


FIGURE 10. Two examples for code analysis questions.

For historical reasons, the author uses his own cloud-hosted Moodle for the previously described activities. The students access Moodle using a self-registration and self-enrollment method. On the other hand, the author uses a university-hosted Blackboard for the virtual classroom and to administer assessment quizzes and exams. The students and the instructor access the virtual classroom in the scheduled lecture time (75 minutes) twice per week. Instead of listening to a lecture, however, the students work on the learning activities and use the chat room to ask questions, if any. We use text chat rather than voice chat to avoid distracting the non-asking students who are busy completing the learning activities. Although students can complete the activities asynchronously, our experience (also with other courses) shows that synchronous learning sessions are very useful for keeping attendance discipline and because the students can get immediate answers to questions that arise during the activity completion.

All course quizzes and exams were computer-based using Blackboard. Blackboard offers a variety of question types that are sufficient to create tests similar to Moodle quizzes. However, Blackboard does not support the question type cloze. Instead, we use a question type called ''Fill in Multiple Blanks''. This question type allows students to enter

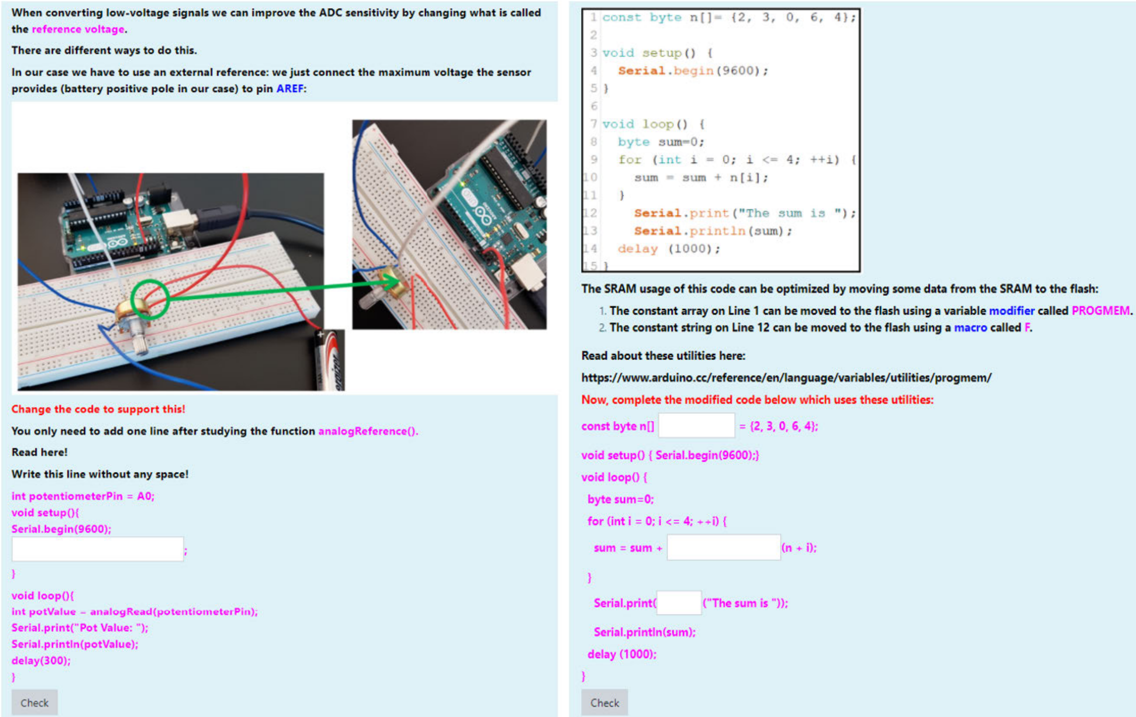


FIGURE 11. Two examples for code design questions.

numbers and short answers. Multiple-choice blanks can also be supported indirectly by giving the list of choices close to the blank. An example of this question type on Blackboard will be given later in Section V.

B. YouTube AND OTHER WEB RESOURCES

As shown in three previous examples, some questions have links to videos on YouTube. We created and uploaded these videos to support learning whenever we felt that such multimedia is useful for understanding the question at hand. Videos were used for three purposes:

1) Showing how to do something, e.g., how to download, install, and configure the Arduino IDE. Generally, videos are known to be effective for learning step-by-step procedures [42].
2) Describing an embedded system's function that should be implemented in coding questions. In this case, we provide a link to the video in the question text as shown in the example of Fig. 9.
3) Explaining the function of a code in the feedback section of code analysis questions as shown in the example of Fig. 6 and the left-side example of Fig. 10. Note how we refer the students to the feedback video explicitly.

In addition to YouTube videos, the learning activities frequently refer the students to other web resources to answer specific questions. Most used resources are hardware datasheets and software language references. The design of embedded systems often requires access to datasheets, e.g., to find the address values of specific registers or the configuration of various hardware units. Typically, datasheets are long documents[1] and reading through them is knowingly difficult even for experts. Therefore, when the learning activity requires information from the datasheet, we try to be specific by giving precise information about its location in the linked datasheet as was seen in the example of Fig. 7. Here, we give the number of the table from which the students should pick the right settings for the clock select bits. The microcontroller datasheet is the most used but not the only one.

Similarly, to facilitate the analysis and the creation of code, we insert links to relevant information in the software manual as was shown in the left-side example of Fig. 11. Here, the students need to learn the syntax of the function analogReference and apply it in the code to obtain the required function.

C. HARDWARE KIT AND DEVELOPMENT SOFTWARE

In the first week of the term, we arranged to provide every student with the hardware kit shown Fig. 13. The kit contains an Arduino UNO board, an LCD, a keypad, buttons, LEDs, resistors, potentiometers, different types of motors and sensors, a breadboard, a battery, and connecting wires. Not all parts were used in the learning activities but made available for the project. The students were supposed to return the kit at the end of the term.

[1]The datasheet of the ATmega328P microcontroller used on Arduino UNO board has 653 pages.

Selecting an appropriate platform is a known issue for teaching and learning embedded systems [43]. Our selection of Arduino was essentially motivated by the distance learning mode on the one hand, and by the simplicity, availability, affordability of this platform, on the other. The Arduino's integrated development environment (IDE) is available for free and easy to use. Arduino is increasingly used in education also in graduate courses [44]. We believe that this technology is sufficient for teaching the fundamentals of embedded systems since memory and speed limitations are not an issue in this case. Apart from this, Ardunio's microcontrollers, including Atmega328P used in UNO, have most of the fundamental capabilities of high-end microcontrollers.

IV. PROJECT

Learning embedded systems is best accomplished when the learned concepts and methods are reinforced through project design. Also, working on design projects is indispensable to the acquisition of practical skills that are hard to cover in lectures or learning activities on specific topics. Such skills include specifying the requirements for solving real-world problems, design space exploration, design decisions, implementation, test, and evaluation. Many authors addressed project-based learning (PBL) for embedded systems [45], [46]. For example, Kumar *et al.* used an FPGA platform to explore hardware and software issues associated with real-time and embedded systems [45]. In one course, the authors asked the students to develop a soccer system on multiple FPGA boards using embedded processors. In another course, the students used an embedded processor-based system to perform decryption of a block encrypted image, accelerated through a custom co-processor.

Also, the role of PBL in supporting self-guided learning was recently addressed by Larson *et al.* [47]. Through students' and instructors' interviews as well as informal classroom observations, the authors found out that project-based learning can facilitate self-directed learning in an embedded systems course. However, this study was conducted in face-to-face learning setting with students using campus facilities and interacting with instructors, teaching assistants, and teammates.

Performing embedded systems' projects at distance raises multiple challenges due to the limited access to lab facilities and the difficulty of face-to-face interaction with teaching staff and team members. Choosing a platform with a freely available development environment and libraries and providing students with hardware, in the way used in our course, can mitigate the necessity to access university labs. However, this approach still has some issues. The selection of the hardware is constrained by the available budget and the class size. Ambitious students with project ideas that require additional hardware may need to purchase this hardware on their own. On the other hand, opting for open platforms with tremendous community contributions make it hard to propose any project that is original and has no related code on the web. Jamieson highlighted this aspect as a major contention ''…the presence of open source and reusable designs makes it difficult to identify what a student is doing'' [48]. The last point makes the assessment of projects difficult because instructors may be unable to check the originality of students' contributions. A related point is that some students taking the course may be working on other projects such as senior design projects using the same platform. If the students are given the freedom to propose their project, some may suggest something related to their other projects. This can raise fairness questions because those students may obtain credits with less work than others.

To mitigate these issues we decided to specify the project topic rather than allowing students to choose a project topic freely. Also, we highlighted that the project will be assessed not only based on the features and functionality of the system but also a deep understanding of the delivered code. In the following, we describe the project topic, how the students worked on the project and how they were assessed.

A. PROJECT TOPIC

The students had to develop a math game, where the player is given number sentences to solve rapidly. The students were encouraged to make the game as interesting and useful as possible and to make comprehensive use of the provided kit components. Using additional hardware was also allowed. The students came up with many interesting features for the game including:

- Support of one, two, or multiple players.
- Support of different difficulty levels. The level was controlled by the type of operations (basic operations, modulo arithmetic, etc.), the range of the numbers, the representation of the numbers (decimal, binary, etc.), or the time-out for entering the answer.
- Using a WiFi shield to play the game using a web browser.
- Using a micro SD or the EEPROM to store player profile and record.
- Using unconnected analog inputs to generate pseudo-random numbers.
- Using a buzzer to play different melodies for winning, losing, etc.
- Using a motion sensor to start the game.
- Register-level programming.

B. PROJECT PROGRESS AND ASSESSMENT

The project assignment was posted on Moodle in Week 5 and the students had eight weeks to complete it. Although the students worked from home, they were encouraged to work in teams of two. Our university provides MS Teams for online meetings and conferences. The teaching assistant was available to address students' questions and give support using MS Teams, too. The students used their cameras to show the circuit if needed.

According to the course syllabus, the project has 20% of the course grade. The course has 3 credit hours corresponding to 135 hours for self-study. So, every student was expected to invest at least $0.2 \times 135 = 27$ hours on the project. The students had to submit a short report which should include a brief description of the game idea and how to play it, a list of the used hardware and libraries, a schematic diagram of the hardware connection, the main structure of the code, and a conclusion. Additionally, the students had to upload the code. The students used Moodle for submitting their projects.

For assessment, we met the students on MS Teams in the last week of the semester. Before each meeting, we reviewed the report and the code. In the meeting, the students first demonstrated their game using their camera. Then they explained their code in brief. At selected points, we asked concrete questions about the code to assess their level of understanding. Some questions were directed to selected team members to assess individual contributions. In general, we were very satisfied with the students' demonstrations and the level of preparation for the meetings. Although most projects were average in terms of game idea and supported features, some teams were very creative. Only one team presented a code that was not functional.

V. EVALUATION

In this section, we first evaluate the proposed design for its ability to engage students in distance learning. Then, we analyze the results of a survey that we conducted to have students' feedback regarding the content and the learning method. Finally, we present the results of item analysis performed on the final and the midterm exams to show the validity of the used quizzing method for computer-based examination.

The course was attended by 43 students who continued to the final exam without any drop. But we are aware that 11 students have opted for a Pass/Fail grade rather than a letter grade for this course. The university allowed the students to choose the grading scheme before a deadline that was two weeks before the final exam. The teaching period is 15 weeks including 30 lecture sessions. Three sessions were used for introducing the projects and related presentations and demonstrations. One session was used for the midterm exam. So, 26 sessions remained for 26 learning activities that will be included in the analyses below. The last learning activity had a different format and had no related learning survey or review quiz.

A. ENGAGEMENT

Moodle generates a completion report for all activities. We analyzed this report and found out that the total *completion rate* of the learning activities is 98.3%. We attribute this high completion rate to the access control mechanism we implemented to chain the activities so that the students cannot start any activity before completing previous related ones, see Fig. 4. We also analyzed the completion date to understand students' *punctuality* at completing the learning activities. Fig. 14 shows the percentage of students who completed the activity on the same day we posted it on Moodle. Accordingly, most of the activities were completed on the posting day by at least 80% of the students. There are three exceptions, which are marked by squares in the figure. These include the last activity and two other activities which were posted on two Saturdays. The reason for this was shortening the lecturing period by one week and extending the examination period by one week to cope with the required social distancing due to COVID-19. To make up for the missing lecture week the university decided to teach on selected Saturdays. Furthermore, we analyzed the completion time of the learning activities' first attempt and found it to vary between 44 and 70 minutes on average, depending on the learning activity.

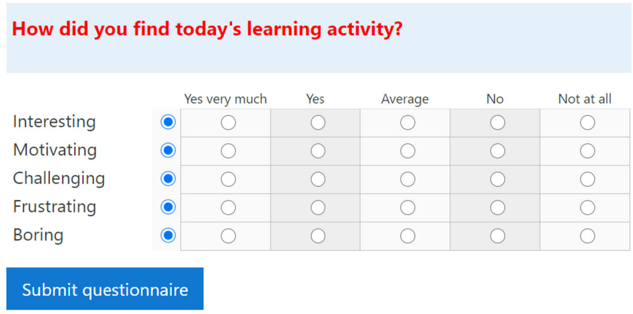


FIGURE 12. Learning survey.

While the completion report confirms the timely engagement with the learning content, it does not reflect the level of cognitive engagement. For this, we analyzed students' grades in the learning activities and their perceptions as well as the relationship between the grades and the perceptions. Recall that the learning activities are graded automatically for feedback –not for assessment– and that we surveyed students' perceptions using the single rating question shown Fig. 12.

Fig. 15 shows the *class average grade* in the first attempt of each learning activity. The statistics show that students' performance varied between 5.79 and 9 out of 10 and that the average grade was 7.84. This performance reflects a considerable level of cognitive engagement, knowing that these grades stem from the first attempt. We gain more insight by looking at the *grade distributions* in two learning activities as seen in Fig. 16. Here, we selected LQ 3-3 with low mean and LQ 8-4 with high mean. The diagram shows that students' grades in LQ 3-3 have a wider distribution over the grade range compared to LQ 8-4. Note how the grades in LQ 8-4 are concentrated in the upper half of the grade range. These grade distributions are in line with students' perceptions of the respective activities as shown in Fig. 17. Note how the students rated LQ 3-3 as clearly more challenging and frustrating than LQ 8-4. We performed a correlation analysis between the mean grades and the students' perceptions for all activities and found out that the grades have:

1) a weak positive correlation with the activity's interestingness and motivation ($r = 0.22$ and 0.38, respectively),

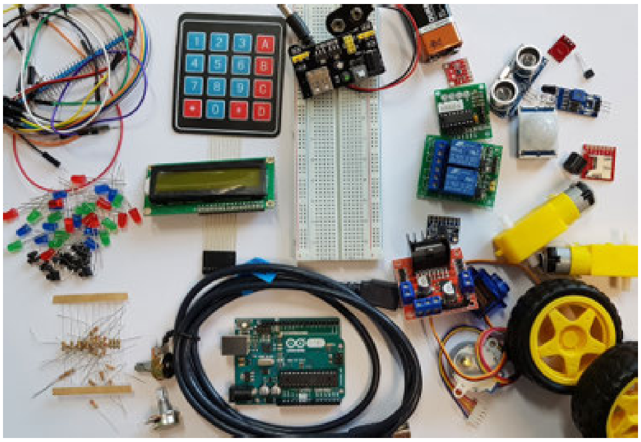


FIGURE 13. Used hardware kit.

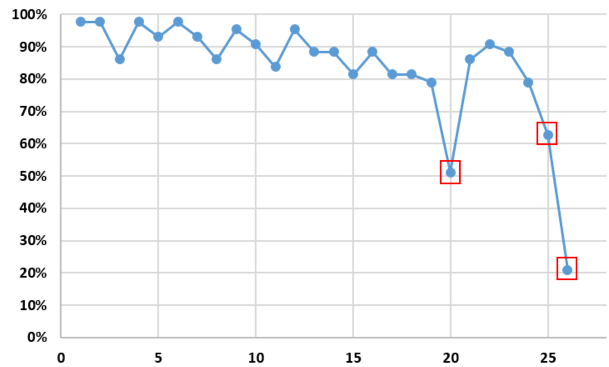


FIGURE 14. Punctuality: percentage of the students who complete the learning activity on the same day they were posted on Moodle.

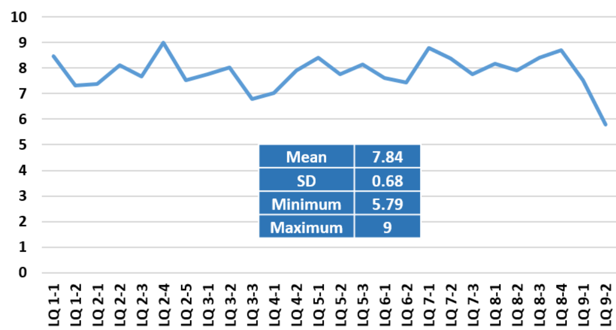


FIGURE 15. Class average grades in the first attempt of the learning activities.

2) a moderate negative correlation with the challenge and frustration ($r = -0.58$ and -0.56, respectively), and
3) a weak negative correlation with the boredom ($r = -0.36$).

We believe that the moderate negative correlation between the perception of the learning activity's challenge/frustration and the attained grade can be used as an indicator for cognitive engagement. On the other hand, the perception of interestingness is consistently high as can be seen in Fig. 18.

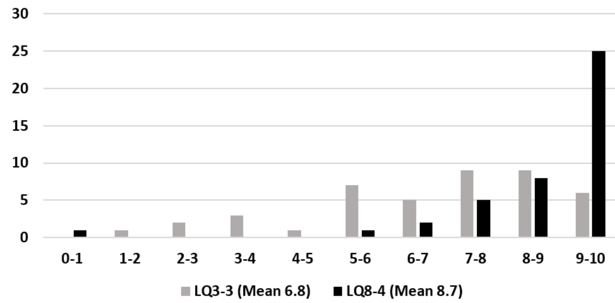


FIGURE 16. Grade frequency distributions for LQ3-3 and LQ8-4.

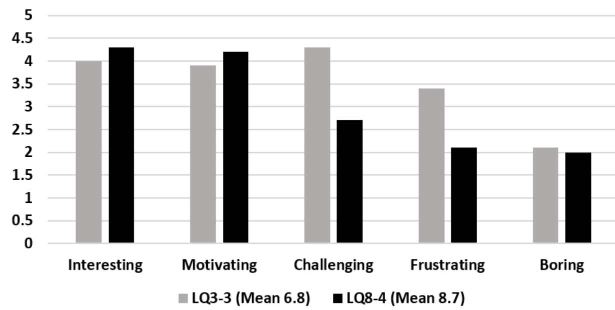


FIGURE 17. Student perceptions of LQ3-3 and LQ8-4.

Task interestingness is known to be highly associated with attention, i.e., with cognitive engagement [49].

B. STUDENTS ANONYMOUS SURVEY

In the second half of the term, we asked the students to comment about their experience with the course anonymously. We formulated our question as follows: *"Please describe this course regarding its content and teaching method! Please be as honest as possible and as detailed as possible! This survey is anonymous."*

18 students responded to this free-text survey and wrote in total 2915 words. We analyzed the responses qualitatively and could identify several aspects that provide insights into student experiences with the course content and teaching/learning method. In particular, most comments refer to one of the following aspects, which will be described below in details:

1) Missing lectures
2) Challenges of active distance learning
3) Usefulness of active distance learning
4) Content evaluation

1) MISSING LECTURES

Five students evaluated the teaching/learning method in consideration of the missing lecture in the course. They suggested that the learning activities should be complemented by lectures before or after doing the activities:

- *I would much rather watch pre-recorded lectures and answer the LQs to check if I understood from the lectures correctly.*

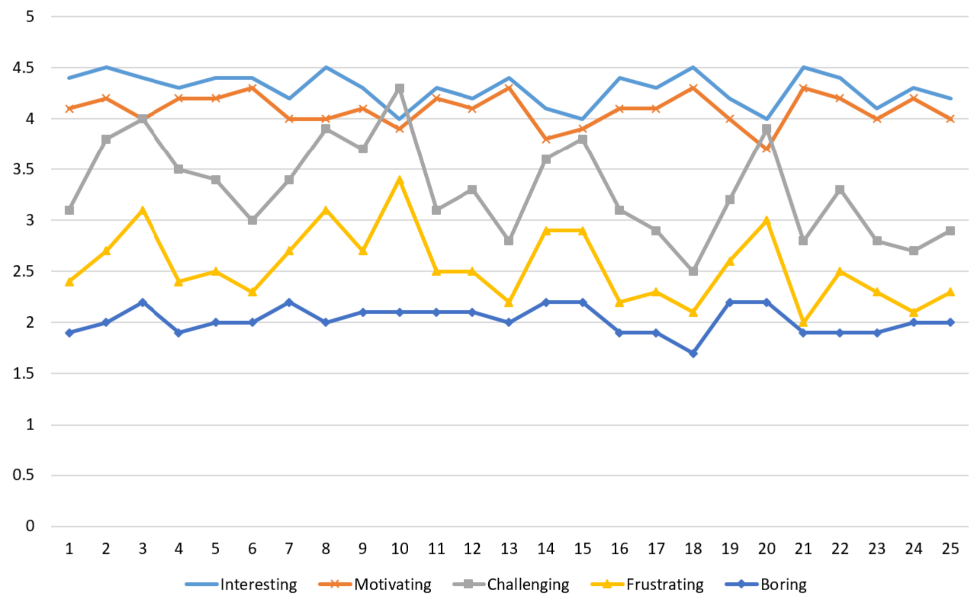


FIGURE 18. Student perceptions of all learning activities.

- *I think a combination of a short lecture explaining the subject before starting the exercises will make it even better. Nevertheless, this method allows us to engage with the material differently.*
- *I think LQ's are useful but not as a full time learning method. Instead what you can do is do lectures and after lectures we can solve the LQ's based on our understanding from the lectures. This makes much more sense.*
- *Students would be able to understand some of the material better in a lecture setting.*
- *While the online course is fun and motivating, I think that it would be preferable if the course included some lecturing as well.*

2) CHALLENGES OF ACTIVE DISTANCE LEARNING

Some students mentioned the challenges they faced while doing the learning activities. These include difficulties in connecting hardware, running and debugging code, doing online examinations, lack of teamwork as well as the lack of lecture notes:

- *There are no instructions on how to make most of the connections which I find very harsh as we have not really worked with Arduino in any course before. I understand that most of us are senior students and hence by now, should be able to analyze hardware parts and connections alone; but I don't think that this assumption should be taken and carried away in an online teaching environment.*
- *However, I was facing problems when running the code on Arduino. The code was not always working as expected which affected somehow my understanding to the content of the course.*
- *In my personal opinion, I like engaging with others to solve problems.*
- *This very time-limited examination environment does not work for everyone.*
- *The exercises along with the feedback do help yes and benefit us a lot and your efforts are indeed truly appreciated, but slides/pdf files can combine the material nicely and in a good order which gives us more freedom and easier access to the material.*

3) USEFULNESS OF ACTIVE DISTANCE LEARNING

Multiple students praised the teaching/learning method as *"creative"*, *"one of the best"*, *"the best way to teach courses"*, or *"very lovely"*. Several of them explicitly or implicitly identified the missing lecture as a feature and highlighted specific useful aspects of active distance learning:

- *The teaching method is really good especially since we are staying at home. In other courses I get distracted and bored after a short while because all we do is keep staring at the screen and listening. But in this course we get to learn on our own at our own pace.*
- *Now that all the exams are online and the lectures are online I find the teaching method very suitable for the situation. I don't think lectures will be as beneficial during these times as the active learning. Even though I might not understand everything when doing the activities but I certainly do when revising later on. Especially that we have the Arduino with us and we can debug our code and see where is the problem, it made me understand things better.*
- *Learn then apply the new knowledge on the Arduino UNO. It is like having a lecture and a lab at the same time.*

- *One of the things that i really appreciate about this teaching method is that there is no one to distract you. You have material that you have to learn yourself with certain activities without any interference from others so you can take it at your preferred pace.*
- *The interactive learning is a wonderful way to learn and i appreciate the effort that is put in making the activities. The knowledge gained stick with me when i learn through the activities.*
- *The Teaching method is perfect in terms of the explanations and way the questions are presented.*
- *Since this course's method does not rely on memorization, we do learn better.*
- *Immediate feedback, students can learn on their own pace, concentration is not a problem.*
- *Easy to understand. Straightforward explanations and feedback. Exercises built into the learning material are great.*

4) CONTENT EVALUATION

When it comes to the content, all the responding students seem to be satisfied. General terms that they used to describe the content include *"fine"*, *"interesting"*, *"very interesting"*, *"enjoyable"*, *"very enjoyable"*, *"most enjoyable"*, *"beneficial"* and *"excellent"*. Several students addressed the load and the difficulty level of the content but were divided about this:

- *"a bit heavy"*, *"overwhelming"*, and *"not for beginners"*
- *"the content difficulty level overall is moderate"*
- *"very understandable and consistent"* and *"can be understood very easily."*

Furthermore, some students highlighted some content-related aspects which they found especially useful. These include:

- Programming skills and register-level programming: *"I like c/c++ programming too so I am finding it easy"*, *"Furthermore, i was always using libraries programing and was not aware about the benefits of register programing and how much it is more enjoyable to do it despite being more difficult"*
- Understanding hardware-related aspects and using datasheets: *"Before i took the course i did not pay too much attention to details like memory usage or where memory is stored etc. However, this course made me learn the importance of that."*, *"I find those questions that ask to find things in the datasheet the best"*, *"The course content has so many details and valuable information. There were parts like timers and memory storage that I studied about in other courses but I didn't really understand the deep details. But now, I'm glad to say that this course was delivered in a way that matches my way of understanding and now I have a better understanding to the timers' and storage concept."*

TABLE 2. Summary of the item analysis results for two major exams [50].

	Midterm Exam	Final Exam
Number of questions	10	11
Average score	70.3%	75.1%
Average time (Min.)	53.5	67
Discrimination		
Good questions	9	9
Fair questions	0	2
Poor questions	1	0
Difficulty		
Easy questions	3	2
Medium questions	7	9
Hard questions	0	0

- The structure of the course and its relation to other courses: *"The topics are also complementary of the other courses and combines both computer architecture course and microprocessor"*, *"Regarding the content of the course, how the topics progressed is very nice. It makes sense and it is easy to understand. I appreciate the thought that went into making the content." "I love tying ideas together, making sense of the whole topics of the course and relating one lesson to another."*

C. ASSESSMENT

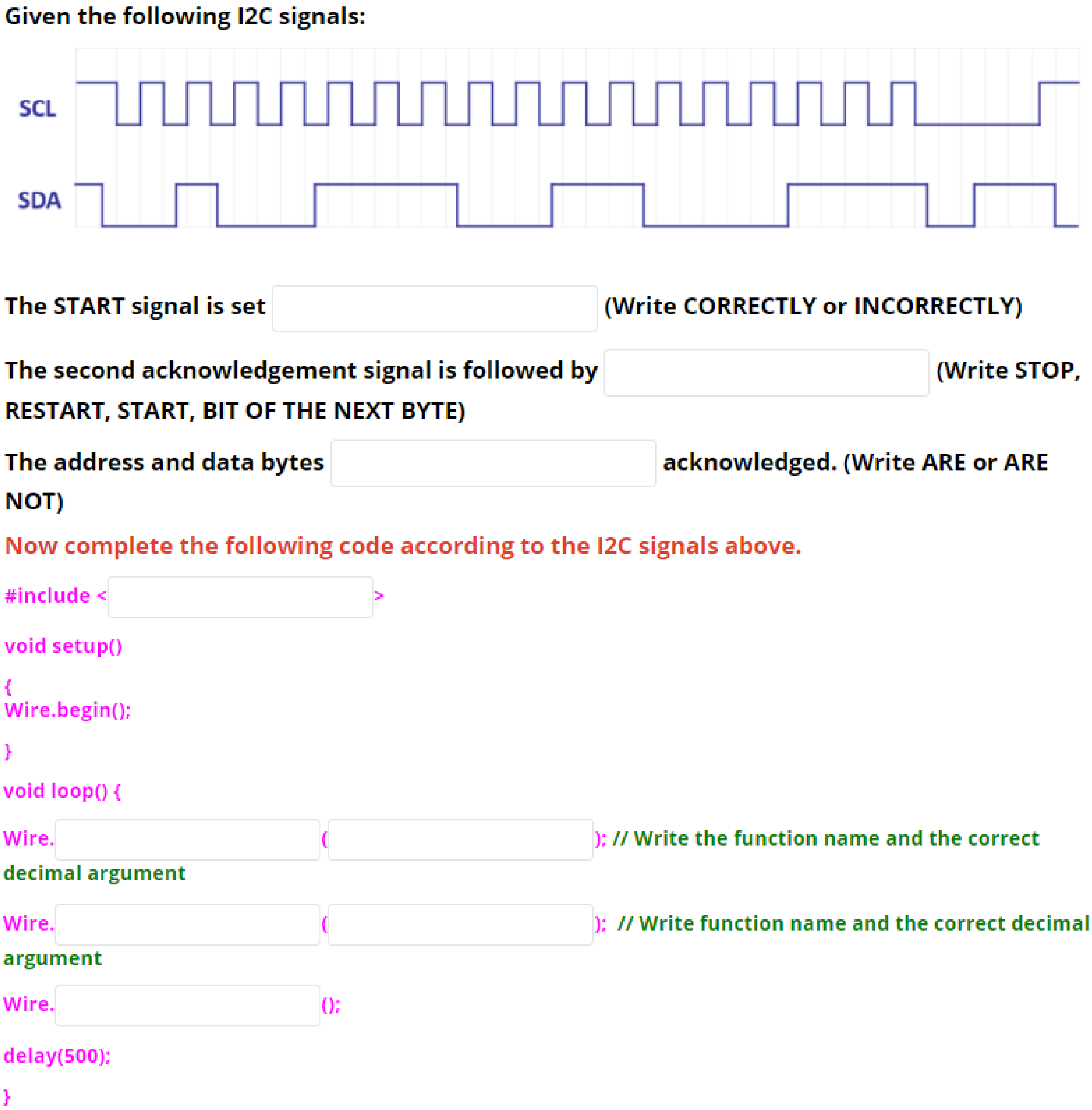
We assessed students' performance using two quizzes (5% each), a midterm exam (30%), a project (20%), and a final exam (40%). As mentioned before the quizzes and the exams were administered on computers using Blackboard. In this section, we just give an example for a test question and summarize the results of the item analyses for the midterm and the final exam to show the appropriateness of these computer-based tests for assessment.

Fig. 19 shows one question from the final exam as it appears on Blackboard. This question is of the type "Fill Multiple Blanks". Note how we provided the choices for the first three blanks. This is to simulate multiple-choice questions that are unsupported in this question type. As for the code, some of the blanks are numerical values. The students should extract these values from the timing diagram, i.e., 0×27 for the salve address and 199 for the sent data byte. Short-answer blanks are filled in with keywords or other labels. The students had access to a keyword sheet to avoid memorization. Some of the keywords required for this question are function names, e.g., beginTransmission.

The results of the item analyses of the midterm and final exams are summarized in Table 2. Accordingly, most of the questions are good in terms of discrimination power and of medium difficulty level.

VI. DISCUSSION

From the experience we gained during the design of learning activities, the interaction with the students in the virtual classroom, and the results of the data analyses shown in the previous section, we discuss some relevant aspects of the proposed design for active distance learning, highlighting



Given the following I2C signals:

SCL

SDA

The START signal is set ________ (Write CORRECTLY or INCORRECTLY)

The second acknowledgement signal is followed by ________ (Write STOP, RESTART, START, BIT OF THE NEXT BYTE)

The address and data bytes ________ acknowledged. (Write ARE or ARE NOT)

Now complete the following code according to the I2C signals above.

```
#include <          >
void setup()
{
Wire.begin();
}
void loop() {
Wire.          (          ); // Write the function name and the correct decimal argument
Wire.          (          ); // Write function name and the correct decimal argument
Wire.          ();
delay(500);
}
```

FIGURE 19. An exam question on blackboard.

some limitations of the approach and proposing some future improvements.

1) STUDENTS' ENGAGEMENT

The benefits of active learning for students' engagement and learning are well reported in the literature [51]. Also, students' engagement through lecture-free instruction was reported [31], [32]. The presented paper features the ability of instructional design to enhance hands-on experience in active distance learning and to make lectures unnecessary. We are not aware of any related work that showed such a constellation.

The results related to students' engagement are promising. Although we cannot prove that all students have performed the hardware/software tasks as expected, the performance in the learning activities as recorded in the Moodle grade book confirm a high level of cognitive engagement. Also, the students' self-reports about their perceptions, especially the rating of the interestingness of the activities, indicate strong engagement. As mentioned before, the literature associates task interestingness with higher attention, hence, with cognitive engagement [49].

Apart from this, some students missed the lecture component in the course. Fortunately, these students suggested that lectures or short lectures should complement the activities rather than replace them. We recognize that the presented approach is very different from what our students are used to. Not all students are on the same level of adaptability to different learning methods. Although we are confident that the presented method is more useful than pure lectures, we may need to test the impact of enhancing the course with some short lectures in the future.

2) MISSING LECTURE NOTES

The students indicated overall satisfaction with the course content. This shows the ability of the proposed design to accommodate comprehensive learning topics. Indeed we agree with students' comments regarding the lack of a compact learning resource for the course such as PDF files. Unfortunately, Moodle (like Blackboard) cannot generate printable versions of quizzes. We are aware that some students try this manually because they prefer to have quick and easy access to the content rather than scrolling through the activities every time they want to study. This is another point that requires our attention in the future.

3) CLASS SIZE

Our class had 43 students. We ran Moodle on a cloud server (IaaS) using a general-purpose Droplet Plan on DigitalOcean (2 virtual CPUs with 2 GB RAM each, 60 GB SSD, and 3 TB data transfer per month). This configuration allowed us to work without any issue in the whole term. Sticking to the scheduled class hours to complete the activities and joining the virtual classroom for answering students' questions were very useful. However, sometimes several students asked questions simultaneously in the chatbox and it was a little hectic to answer them one by one. This is especially true when there is an issue in some question. Sometimes we had to return to the question and fix it. We don't remember any significant issue with this but we can expect that a larger class would have caused more questions to come simultaneously. At any rate, this approach helped us fix most if not all issues in the learning activities so that we expect to get fewer questions in the future.

4) HARDWARE DEPENDENCY

The course and all learning activities strongly rely on Arduino and the hardware kit we provided. This is a limitation for the portability of the course. On the other hand, technology selection is a characteristic issue for teaching embedded systems. We support the view that focusing on one technology is more useful than using multiple technologies or making the course technology-independent. Having said this, the proposed design in terms of the pedagogical framework as illustrated in Fig. 2 is indeed portable and can be used with different hardware and software.

5) DESIGN OVERHEAD

We estimate that the design of one learning activity took the author approximately 15 hours on average. This time is spent on:

1) Finding an appropriate problem to base the learning activity on.
2) Hardware connection, coding, and debugging.
3) Creating learning activity on Moodle.

Note that the first two tasks should be accomplished with the third in mind because students should be able to complete the activity within the learning session (75 minutes). Through the repeated design of learning activities, we gained the experience that this task is artistic in a sense: In the beginning, it hard to tell how the learning activity will look like at the end. It is essential to be aware of relevant learning theories and to know the capabilities of the used technologies. However, the segmentation of the topic, the sequencing of the segments, the selection when to ask to code, when to upload a video, what question type, how much review, scaffolding, or feedback should be given, are all decisions that should be taken creatively during the design and cannot be planned easily in advance.

Indeed, the time overhead of this design can be a significant limitation. Still, we believe that this time investment is justified because it benefits students' learning. On the other hand, this overhead is more or less a one-time expense because the activities can be reused without or with small changes in the future. Furthermore, it should be highlighted that the computer-based assessment and automatic grading supported in this method provide considerable compensation by saving marking time.

6) OPEN ACCESS TO THE COURSE[2]

Learning management systems are usually used within institutions and only members of these institutions are granted access to content posted on these systems. In contrast, deploying our course on a cloud-based Moodle with self-registration and self-enrollment makes the course available to interested users outside our university.

7) PROJECT COMPLETION AT DISTANCE

The main focus of this paper was on the replacement of lectures by learning activities including hands-on experiments. The project component of the course was not investigated in depth. So, we did not collect data that can help in analyzing how the students worked in teams, which challenges they faced, and how they addressed their challenges. We believe that project-based learning in distance mode deserves a dedicated study in the future.

VII. CONCLUSION

The move to distance learning due to the COVID-19 pandemic poses several challenges for faculty and students. Student engagement is one of the most mentioned issues in this context. Active learning is often proposed to mitigate the situation. However, active learning can only make sense if students can receive timely scaffolding and feedback. This can be an issue for big classes. Another challenge in distance learning is the lack of access to physical laboratories and the difficulty of having hands-on experiments in practical courses such as embedded systems.

[2]The course is available at https://www.learn-smartly.com. Please contact the author to arrange access for your students and provide an enrolment key.

In this paper, we presented a pedagogical framework for active distance learning. The framework was applied to design a course on embedded systems with practical exercises. The results show strong student engagement and positive perceptions. Indeed, technology plays a central role in the current shift to distance learning. However, for full advantage, this technology should be deployed thoughtfully in the context of a solid pedagogical framework. This requires strong engagement from us and considerable time investment.

APPENDIX A
QUESTION TEXT RELATED TO FIG. 5

In the last session you learned about the signaling in the I2C bus.

Complete the following text to recall what you learned!

The I2C transfers {:NM: = 1} bit per clock cycle. (Write a number)

The data on the SDA line are only allowed to change while the SCL line = {:NM: = 0}. (Write a number)

However, there are three exceptions to this rule: sending a START, REPEATED START, or {:SA: = STOP} signal.

While any chip can be a master on the bus, at one time point only {:MCS: = one chip∼ two chips} can be master/masters. All other devices must be /:SA: = slaves/.

A master must send a {:SA: = START} signal or a RESTART (REPEATED START) signal to take the bus, and a :SA: = STOP signal to release the bus.

After a START or RESTART signal, the first byte the master must send is the 7-bit {:SA: = address} of the slave followed by {:NM: = 1} bit/bits for the mode. (Write a number)

When READ/WRITE = {:NM: = 1}, the master is in the reception/read mode.

When READ/WRITE = {:NM: = 0}, the master is in the transmission/write mode.

The {:MCS: = MSB∼ LSB} of the address/data must be transferred first.

Every received byte must be acknowledged by the receiver, who {:MCS: = can be the master or a slave∼ must be the master∼ must be a slave}.

An acknowledgement signal (ACK) has always the binary value {:NM: = 0}.

A negative acknowledgement (NACK) has always the binary value {:NM: = 1}.

NACK {:MCS: = can be "sent" by the master or a slave∼ must be "sent" by the master∼ must be "sent" by a slave}.

REFERENCES

[1] J. Arquilla and M. Guzdial, "Transitioning to distance learning and virtual conferencing," *Commun. ACM*, vol. 63, no. 7, pp. 10–11, Jun. 2020, doi: 10.1145/3398386.

[2] T. DeFranco and J. Roschelle, "Detecting/preventing infections, and moving instruction online," *Commun. ACM*, vol. 63, no. 6, pp. 8–9, May 2020, doi: 10.1145/3392370.

[3] S. L. Schneider and M. L. Council, "Distance learning in the era of COVID-19," *Arch. Dermatol. Res.*, pp. 1–2, May 2020, doi: 10.1007/s00403-020-02088-9.

[4] S. Unger and W. Meiran, "Student attitudes towards online education during the COVID-19 viral outbreak of 2020: Distance learning in a time of social distance," *Int. J. Technol. Educ. Sci.*, vol. 4, no. 4, pp. 256–266, Sep. 2020.

[5] M. Adnan and K. Anwar, "Online learning amid the COVID-19 pandemic: Students' perspectives," *Online Submission*, vol. 2, no. 1, pp. 45–51, 2020.

[6] Z. Lassoued, M. Alhendawi, and R. Bashitialshaaer, "An exploratory study of the obstacles for achieving quality in distance learning during the COVID-19 pandemic," *Educ. Sci.*, vol. 10, no. 9, p. 232, Sep. 2020.

[7] E. Chen, K. Kaczmarek, and H. Ohyama, "Student perceptions of distance learning strategies during COVID-19," *J. Dental Educ.*, pp. 1–2, Jul. 2020.

[8] D. Roache, D. Rowe-Holder, and R. Muschette, "Transitioning to online distance learning in the COVID-19 era: A call for skilled leadership in higher education institutions (HEIs)," in *Proc. CCEAM*, 2020, p. 103.

[9] K. Kaczmarek, E. Chen, and H. Ohyama, "Distance learning in the COVID-19 era: Comparison of student and faculty perceptions," *J. Dental Educ.*, pp. 1–3, Oct. 2020.

[10] A. K. Brady and D. Pradhan, "Learning without borders: Asynchronous and distance learning in the age of COVID-19 and beyond," *ATS Scholar*, vol. 1, no. 3, pp. 233–242, Sep. 2020.

[11] M. Sami, M. Malek, U. Bondi, and F. Regazzoni, "Embedded systems education: Job market expectations," *ACM SIGBED Rev.*, vol. 14, no. 1, pp. 22–28, Jan. 2017.

[12] P. Caspi, A. Sangiovanni-Vincentelli, L. Almeida, A. Benveniste, B. Bouyssounouse, G. Buttazzo, I. Crnkovic, W. Damm, J. Engblom, G. Folher, and M. García-Valls, "Guidelines for a graduate curriculum on embedded software and systems," *ACM Trans. Embedded Comput. Syst.*, vol. 4, no. 3, pp. 587–611, 2005.

[13] R. E. Seviora, "A curriculum for embedded system engineering," *ACM Trans. Embedded Comput. Syst.*, vol. 4, no. 3, pp. 569–586, Aug. 2005.

[14] A. L. Sangiovanni-Vincentelli and A. Pinto, "An overview of embedded system design education at berkeley," *ACM Trans. Embedded Comput. Syst.*, vol. 4, no. 3, pp. 472–499, Aug. 2005.

[15] A. Minaie and R. Sanati-Mehrizy, "A comparison of embedded systems education in the united states, European, and Far Eastern countries," in *Proc. Annu. Conf. Expo.*, 2007, pp. 2–6.

[16] M. Grimheden and M. Törngren, "What is embedded systems and how should it be taught?—Results from a didactic analysis," *ACM Trans. Embedded Comput. Syst.*, vol. 4, no. 3, pp. 633–651, Aug. 2005.

[17] J. O. Hamblen and G. M. E. van Bekkum, "An embedded systems laboratory to support rapid prototyping of robotics and the Internet of Things," *IEEE Trans. Educ.*, vol. 56, no. 1, pp. 121–128, Feb. 2013.

[18] M. Moallem, "A laboratory testbed for embedded computer control," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 340–347, Aug. 2004.

[19] I. A. Elhaty, T. Elhadary, R. Elgamil, and H. Kilic, "Teaching university practical courses online during COVID-19 crisis: A challenge for elearning," *J. Crit. Rev.*, vol. 7, no. 8, pp. 1–10, 2020.

[20] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literature review," *ACM Comput. Surv.*, vol. 38, no. 3, p. 7, Sep. 2006.

[21] E. K. Faulconer and A. B. Gruss, "A review to weigh the pros and cons of online, remote, and distance science laboratory experiences," *Int. Rev. Res. Open Distrib. Learn.*, vol. 19, no. 2, pp. 1–15, May 2018.

[22] M. E. Auer, A. K. Azad, A. Edwards, and T. De Jong, *Cyber-Physical Laboratories in Engineering and Science Education*. Cham, Switzerland: Springer, 2018.

[23] I. Angulo, L. Rodriguez-Gil, and J. Garcia-Zubia, "Scaling up the lab: An adaptable and scalable architecture for embedded systems remote labs," *IEEE Access*, vol. 6, pp. 16887–16900, 2018.

[24] P. Anzhelika, G. Olga, E. Ivanov, A. Sokolyanskii, and S. Kurson, "Development and application of remote laboratory for embedded systems design," in *Proc. 12th Int. Conf. Remote Eng. Virtual Instrum. (REV)*, Feb. 2015, pp. 69–73.

[25] P. Orduña, L. Rodriguez-Gil, J. Garcia-Zubia, I. Angulo, U. Hernandez, and E. Azcuenaga, "Increasing the value of remote laboratory federations through an open sharing platform: Labsland," in *Online Engineering & Internet of Things*. Cham, Switzerland: Springer, 2018, pp. 859–873.

[26] J. Glassey and F. D. Magalhães, "Virtual labs–love them or hate them, they are likely to be used more in the future," *Educ. Chem. Eng.*, vol. 33, pp. 76–77, Oct. 2020.

[27] R. Li, J. R. Morelock, and D. May, "A comparative study of an online lab using Labsland and zoom during COVID-19," *Adv. Eng. Educ.*, vol. 8, pp. 1–10, Dec. 2020.

[28] S. Caño de las Heras, B. Kensington-Miller, B. Young, V. Gonzalez, U. Krühne, S. S. Mansouri, and S. Baroutian, ''Benefits and challenges of a virtual laboratory in chemical and biochemical engineering: Students experiences in fermentation,'' *J. Chem. Educ.*, vol. 98, no. 3, pp. 866–875, Jan. 2021.
[29] C. R. Graham, ''Blended learning systems,'' *The Handbook of Blended Learning: Global Perspectives, Local Designs*. San Francisco, CA, USA: Wiley, 2006, pp. 3–21.
[30] J. L. Bishop and M. A. Verleger, ''The flipped classroom: A survey of the research,'' in *Proc. ASEE Nat. Conf.*, Atlanta, GA, USA, vol. 30, no. 9, Jun. 2013, pp. 1–18.
[31] A. Shoufan, ''Lecture-free classroom: Fully active learning on moodle,'' *IEEE Trans. Educ.*, vol. 63, no. 4, pp. 314–321, Nov. 2020.
[32] V. Minderhout and J. Loertscher, ''Lecture-free biochemistry: A process oriented guided inquiry approach,'' *Biochem. Mol. Biol. Educ.*, vol. 35, no. 3, pp. 172–180, 2007.
[33] ACM/IEEE. (2016). *Computer Engineering Curricula*. [Online]. Available: https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf
[34] J. J. G. van Merrienboer, P. A. Kirschner, and L. Kester, ''Taking the load off a Learner's mind: Instructional design for complex learning,'' *Educ. Psychol.*, vol. 38, no. 1, pp. 5–13, Mar. 2003.
[35] R. Moreno, ''Optimising learning from animations by minimising cognitive load: Cognitive and affective consequences of signalling and segmentation methods,'' *Appl. Cognit. Psychol.*, vol. 21, no. 6, pp. 765–781, 2007.
[36] M. Pedaste, M. Mäeots, L. A. Siiman, T. de Jong, S. A. N. van Riesen, E. T. Kamp, C. C. Manoli, Z. C. Zacharia, and E. Tsourlidaki, ''Phases of inquiry-based learning: Definitions and the inquiry cycle,'' *Educ. Res. Rev.*, vol. 14, pp. 47–61, Feb. 2015.
[37] P. Chandler and J. Sweller, ''The split-attention effect as a factor in the design of instruction,'' *Brit. J. Educ. Psychol.*, vol. 62, no. 2, pp. 233–246, Jun. 1992.
[38] M. L. Epstein, A. D. Lazarus, T. B. Calvano, K. A. Matthews, R. A. Hendel, B. B. Epstein, and G. M. Brosvic, ''Immediate feedback assessment technique promotes learning and corrects inaccurate first responses,'' *Psychol. Rec.*, vol. 52, no. 2, pp. 187–201, Apr. 2002.
[39] J. Maybin, N. Mercer, and B. Stierer, ''Scaffolding learning in the classroom,'' *Thinking Voices: The Work of the National Oracy Project*. London, U.K.: Hodder, 1992, pp. 186–195.
[40] B. Rittle-Johnson and M. Schneider, ''Developing conceptual and procedural knowledge of mathematics,'' *Oxford Handbook of Numerical Cognition*. Oxford, U.K.: Oxford Univ. Press, 2015, pp. 1118–1134.
[41] W. E. Hall and S. H. Zweben, ''The cloze procedure and software comprehensibility measurement,'' *IEEE Trans. Softw. Eng.*, vol. SE-12, no. 5, pp. 608–623, May 1986.
[42] D. Y. Lee and M. R. Lehto, ''User acceptance of YouTube for procedural learning: An extension of the technology acceptance model,'' *Comput. Educ.*, vol. 61, pp. 193–208, Feb. 2013.
[43] I. Kastelan, N. Teslic, and M. Temerinac, ''Challenges in embedded engineering education,'' in *Embedded Engineering Education*. Cham, Switzerland: Springer, 2016, pp. 1–27.
[44] M. El-Abd, ''A review of embedded systems education in the Arduino age: Lessons learned and future directions,'' *Int. J. Eng. Pedagogy*, vol. 7, no. 2, pp. 79–93, 2017.
[45] A. Kumar, S. Fernando, and R. C. Panicker, ''Project-based learning in embedded systems education using an FPGA platform,'' *IEEE Trans. Educ.*, vol. 56, no. 4, pp. 407–415, Nov. 2013.
[46] R. C. Hsu and W.-C. Liu, ''Project based learning as a pedagogical tool for embedded system education,'' in *Proc. 3rd Int. Conf. Inf. Technol., Res. Educ. (ITRE)*, Jun. 2005, pp. 362–366.
[47] J. Larson, S. S. Jordan, M. Lande, and S. Weiner, ''Supporting self-directed learning in a project-based embedded systems design course,'' *IEEE Trans. Educ.*, vol. 63, no. 2, pp. 88–97, May 2020.
[48] P. Jamieson, ''Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?'' in *Proc. Int. Conf. Frontiers Educ., Comput. Sci. Comput. Eng. (FECS)*, 2011, p. 1.
[49] S. Hidi and K. A. Renninger, ''The four-phase model of interest development,'' *Educ. Psychol.*, vol. 41, no. 2, pp. 111–127, Jun. 2006.
[50] Blackboard. *Item Analysis*. Accessed: Feb. 20, 2021. [Online]. Available: https://help.blackboard.com/Learn/Instructor/Tests_Pools_Surveys/Item_Analysis
[51] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, ''Active learning increases student performance in science, engineering, and mathematics,'' *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 23, pp. 8410–8415, Jun. 2014.

ABDULHADI SHOUFAN received the Dr.-Ing. degree from Technische Universität Darmstadt, Germany, in 2007. He is currently an Associate Professor of electrical engineering and computer science with Khalifa University, Abu Dhabi. His research interests include drone security and safe operation, embedded security, cryptography hardware, learning analytics, and engineering education.