Teaching Real-Time Programming for Embedded Microcontrollers by Using Cloud-Based Simulator

Samuel Kosolapov, Braude Academic College of Engineering, Israel

Abstract

In "normal" times, students of electronic department enrolled in the course "Real-Time Digital Signals Processing" (RT-DSP) use real hardware – dedicated development boards containing microcontrollers and additional electronic components, and dedicated software (installed on the personal computer) to program those development boards. However, because of COVID 19 limitations, the distribution of development boards and dedicated software to the students became problematic. Even more, technical assistance in case of installation and/or hardware problems became practically impossible. Learning real-time algorithms and their practical implementation is an important part of Electronic Engineer's education; hence, it was decided to adapt the RT-DSP course to distant learning by using a cloud-based TinkerCad simulator. It was found that by using some tricks and simplifications, reasonably sophisticated electronic projects can be implemented. Additionally, it was found that the cloud-based TinkerCad simulator can run in parallel with the ZOOM session, while the Internet speed of the lecturer's computer is at least 50 Mbps – which means that providing frontal lectures and exercises (containing plain PowerPoint presentations and live simulations demonstrations) is feasible, at least for the groups of 30 students. After evaluating and grading homework and micro-projects created by students in the frames of this course, we may conclude that the professional level of the designed systems is good enough to continue this course in the frames of distant learning, in case "normal" options are not available.

Keywords: Real-Tine, DSP, TinkerCad, STEM, Cloud-Based Education

Introduction

Classical course RT-DSP (RT stands for Real-Time, whereas DSP stands for Digital Signal Processing) is "*a hard to learn course*" for advanced electronics engineers. It is provided after students of the electronics department learn basic math and physics (including complex numbers and Fourier Transform), and, after they have learned basic analog and digital electronics and basic programming (preferably C language). RT-DSP mainly is about the following items: how to make the acquisition of analog voltage signals from different sensors (for example, from microphone), how to use ADC (Analog to Digital Converter) to convert analog voltage signal to digital signal (sequence of numbers that can be stored in the electronic memory), how to use microcontrollers to do an acquisition, how to process digital signals practically (for example clear human speech from the street noises), and how to do all above fast enough by using specially designed microcontrollers. "Fast enough" in the content of RT-DSP means: in accordance with exact timing criteria described in technical terms for the specific application.

Providing the RT-DSP course is an obvious "pedagogic challenge. Some of the goals of this course are: to fortify the theoretical basis learned in the prerequisite courses, explain a set of classical DSP algorithms (like filtration by convolution and by using FFT (Fast Fourier Transform)), and explain using of advanced RT concepts (like timers, hardware interrupts, ring buffers, dual-buffer techniques, Q-Numbers). However, learning a "theory only" is not enough for the modern electronic engineer. Hence, additional goals here are: to equip the students with "tools" needed to acquire, at least, minimal practical experience: how to implement classical DSP algorithms by using advanced RT concepts while using real electronic components and development boards, and by using modern software development and simulation tools.

In the routine work, an electronic engineer typically uses specially designed "development boards". Those boards have some set of electronic components, which set is supposed to be adequate to design a DSP system in accordance with the customer' requirements. These days those development boards can be programmed by a connected to them PC. Obviously, professional "development boards" are too sophisticated and too expensive for most universities and colleges. Hence, simplified "development boards" are to be used in the frames of RT-DSP courses.

Literature Review

Considering that the DSP course is a classical course for the electronic engineer, many articles and reports on the scientific conferences were dedicated to the details of teaching this course. Many educators strongly believe, that dealing with real electronic components (like amplifiers, microcontrollers, ADDC, DAC) is a must for the course of that kind (Nikolic, 2015). Some educators use only mathematic tools to demonstrate usage of the typical DSP algorithms: "A substantial amount of effort from the teacher is required to deliver mathematical and algorithmic concepts", hence "specially designed Windows Store App can be used as a teaching aid for an introductory undergraduate DSP course" like "linear convolution, circular convolution, radix-2 Fast Fourier Transform (FFT), and Finite Impulse Response (FIR) filter design" (Diya, 2017). The above authors use non-electronic software emulation of a "hypothetical 16-bit floating-point digital signal processor, with a simple instruction set". Considering that software-based electronic simulators our days arrived to high fidelity, many educators started to use dedicated electronic simulators and software tools

(like LabView) for electronic courses (Yi, 2005). Some educators promote the "remote laboratory" concept, according to which, students remotely operate specific electronic equipment positioned in the "normal" electronic laboratory. (Auer, 2000). Taking into account that the DSP course contains a lot of sophisticated math, and, thus, is not simple for many students, many educators promote "project-based" learning (Hoffbeck, 2012).

Providing RT-DSP course without dedicated in-campus laboratory

If the lecturer decides to use nearly professional "development boards", then dedicated for this course class with proper professional and expensive equipment must be organized. In the Braude Academic College of Engineering, there is a deficit of non-occupied rooms, so that organizing a dedicated laboratory for the RT-DSP course was not an option. Hence, it was decided to prepare kits containing inexpensive "development boards" and additional electronic components. At the beginning of the semester, students are organized in pairs, and, every pair get the kit up to the end of the semester. Then, by using those kits, students implement the assignments in their homes. This decision about kit distribution eliminates the need to reserve the full-fledged RT-DSP laboratory.

Figure 1 demonstrates the components of the kit. The kit is stored in the durable box of the popular EasyPic Fusion V7 development board (see callouts 1a and 1b). When the box arrived from the factory, it contained many documentation leaflets and electronic components. As for the documentation – it is available on the Internet, so that there is no need to give "paper-based" documentation to the students. Most of the electronic components that are part of the box are not needed for this specific course, hence students did not get them. The main element of the kit is EasyPIC Fusion v7 development board (Callout 2 in Figure 1). "Easy" in the name of the board means that using of this board is not as complicated as using professional development boards. Additionally, "easy" in that case means that configuration for the different projects can be set by jumpers on the board and by software. Callout 2a points to the color 320x240 TFT screen. Usage of this screen enables to ask students to prepare applications that present graphically as input signals, as signals processed by a specific DSP algorithm.
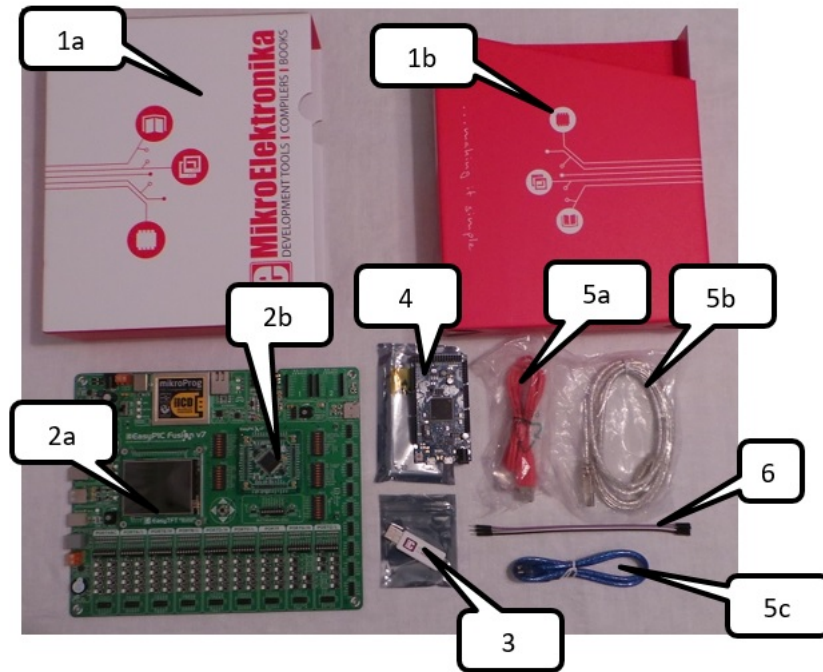
Figure 1 Components of the RT-DSP kit

Callout 2b points to the sophisticated DSP processor (PIC32MX795F512L MCU board). Additionally, the kit contains a "USB Dongle" (Callout 3). In order to program this board, it must be connected to the PC by using the standard USB cable (5a or 5b). Then special software ("Easy PIC compiler") must be downloaded and installed on the PC. This compiler can work without the board and without the dongle, but only in debug mode. Surprisingly, this feature was found extremely useful in the frames of this RT-DSP course. A student in the specific pair, which is currently without the board, can still write a code of some function and check how much time is needed for every line of the code. Then different algorithms can be compared relative to their time efficiency. Additionally, it can be decided whether the selected algorithm can be considered as a real-time algorithm by taking into account data type and size and the algorithm selected. By inserting simple screenshots of the compiler ' windows into assignment' report, students can prove that the algorithm they selected and implemented as C-code can be considered as real-time code in this specific situation. Additionally, students may be asked to find "the limits" of the application created. Actually, this usage of the compiler can be considered as a simulation with special features, and, as such, will be used in situations when the development board cannot be given to the students. The development board can be programmed without the USB Dongle, but in that case, only simple applications can be created. The normal situation is when the board is connected by USB cable to one USB port of the PPC, whereas the USB dongle is inserted into another USP port of the same PC.

While using the compiler in the "debug" mode, the input signal can be created by some software procedure. For example, a pure sinusoidal signal (or any other signal) can be created as an array filled by proper values calculated inside the "for loop". Then, this "synthetic" signal can be used as an input to the RT-DSP algorithm under the test. The advantage of this approach is that the input signal is "clean" and has no noise. In the "classical" electronic laboratory, an input signal is typically created by the dedicated Signal Generator. Despite the fact that a simple signal generator can be assembled from a small number of electronic components, it was decided to use the "Arduino DUE" development board as a "software-

defined signal generator". This usage is possible because Arduino DUE has two DACs, so that students can write simple code that creates the input signal as values in an array in Arduino DUE memory and then connects DAC pins of Arduino DUE with ADC pins of the EasyPIC development board. Figure 2 demonstrates the simple connection of the Arduino DUE Development Board (on the right) to Easy Pic Development Board (on the left) for this purpose. To connect both boards, students can use wires marked as 6 in Figure 1. USB Cable (marked as 5c in Figure 1) is used to program Arduino UNO and/or power it when needed.
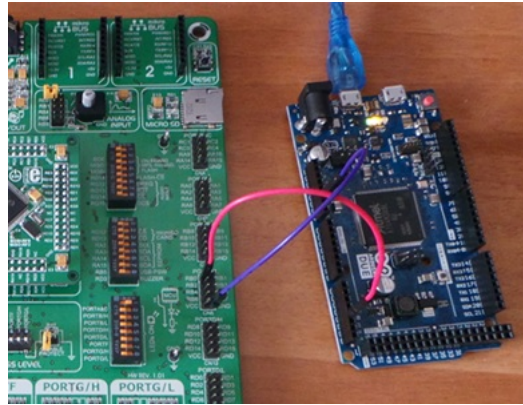

Figure 2. DAC0 pins of the Arduino UNO Development Board (on the right) connected to ADC pins of the Easy PIC Development Board (on the left)

Figure 3 demonstrates the result of classical RT-DSP Algorithm operation – filtration of 50 Hz noise by using a convolution filter. As input, as output signals are shown on the TFT Screen of the EasyPIC Development Board. Upper signal - rectangle wave with added 50 Hz sinusoidal noise created as "synthetic signal" by the code, uploaded to the Arduino DUE Development Board. Bottom signal: rectangle wave cleaned from sinusoidal noise by using a digital notch filter.

The above example demonstrates how RT-DSP assignments can be implemented without a dedicated electronic laboratory. Basically, there is no need to bring the kits to the campus. However, there are some situations when something goes wrong. This can be as software as hardware problem – for example, faulty cables. In a "normal" semester, students have an option to arrive at any available standard electronic laboratory and provide additional tests by using the equipment of a standard electronic laboratory – like oscilloscopes, signal generators, etc. Figure 4 demonstrates how a pair of students arrived with their kit at the standard electronic laboratory and validated signals by using the oscilloscope.
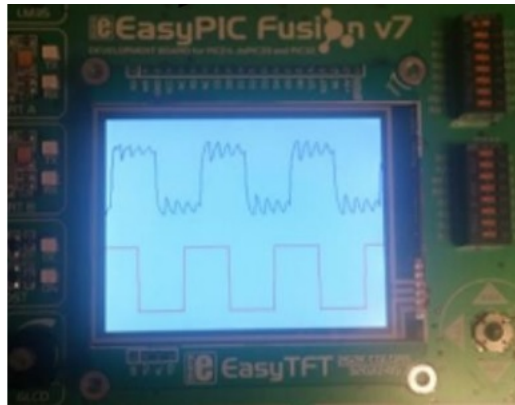
Figure 3. Example of RT-DSP operation. Upper signal: rectangle wave with added 50 Hz sinusoidal noise. Bottom signal: rectangle cleaned from sinusoidal noise by using a digital notch filter.

Figure 4. Pair of students validate signals by using equipment of a standard electronic laboratory

Logistics of RT-DSP Course in a "normal" semester

In "normal" times students of the electronic department of Braude Academic College of Engineering enrolled in the course "Real-Time Digital Signals Processing" (RT-DSP) use real hardware – dedicated development boards containing microcontrollers and additional electronic components, and dedicated software (installed to the personal computer) to program those development boards.

Specifically, as described above, pair of students get "a kit" containing "real electronic boards": "EasyPIC Fusion v7 development board" , "Arduino DUE board" (used as a software-controlled signal generator), a set of cables and wires, and USB dongle needed to use dedicated software needed to program above boards by using PC in accordance with the relevant assignment requirements.

By using the above "kit", students can practically implement (by writing a C code) at his/her home theoretical concepts (learned in the basic courses) in the frames of Home Works and Micro Projects. Figure 5 shows how exactly, by using a laptop and elements of the kit, students can arrange all needed equipment on the small table. It is the job of the students, to

show input (raw) and output (processed) signals on the graphical TFT screen of the Development Board. It can be seen that at least three USB ports are needed. Not all modern laptops have this number of USB ports. In that case, USB Hub must be used.
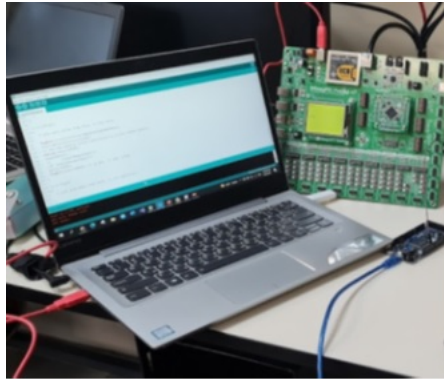

Figure 5. The exemplary layout of the laptop and two development boards on the small table at the student's home. All needed is on the table.

In the frames of the course RT-DSP, students attend 13 two-hour frontal lectures and 13 one-hour exercises. Typically the lecturer uses slides and a blackboard to explain relevant theoretical concepts and practical details of their real-time implementation. Additional materials like videos and ready-to-use project' examples can be freely downloaded from the course site. Presence in the lectures and in the exercises is not a must, but highly recommended.

To practice the theoretical concepts learned, students must execute three Home Works. In a regular semester, students are expected to create simple, but practically usable systems by using real boards and/or simulations and compile detailed PowerPoint reports on what, how, and why was done.

Additionally, students must prepare two Micro Projects. While Home Works assignments are the same for all pairs of students, Micro Project assignments differ for every pair of students – every pair must learn, implement and present in the class some RT-DSP algorithm that was not learned in detail in the lectures and exercises.

It is clear that when students work in pairs, it is very hard to understand what the impact of every student in the assignment preparations is. Ten in-class micro-works (that can be provided at any time during the lecture) may help to reward hard-working students that visit the lectures and exercises and demonstrate their understanding of the algorithms learned.

Providing RT-DSP course under COVID19 restrictions

At the beginning of the 2020 spring semester, strict COVID 19 restrictions were imposed. Lecturers of Braude Academic College of Engineering were given three days to provide alternatives to working with real electronic components. It was even impossible to prepare the kits and distribute them to the pair of students. Furthermore, even then, only one of the students in the pair would be able to work "as before". It was also clear that in case of a hardware problem, replacement of a faulty equipment would be practically impossible. After short tests, the TinkerCad simulator was selected as a proper alternative.

TinkerCad advertisement claims that: TinkerCad is a free, easy-to-use (really easy to use) web application (no need to install) that equips the next generation of designers and engineers with the foundational skills for innovation: 3D design (not used in this course) analog and digital electronics + some microcontrollers, and coding. Community of TinkerCad is about 35 million users.

It was found that the cloud-based reporting technique, developed for other electronic courses, can be used for remote reporting and grading of the student's RT-DSP assignments with minor modifications (Kosolapov, 2019, 2022). When simulation became the only possible option, students, as proof of execution, added to their report screenshots of the simulation's screens instead of adding photos of the working system. Additionally, a link to the working stimulation must be added to the first page of the PowerPoint report, so that the educator was able to check remotely all the details: how exactly students executed specific assignment. Unfortunately, the current version of TinkerCad supports only an extremely simple Arduino UNO R3 board. The small size of the RAM limits the size of data that can be processed. However, considering that Arduino UNO R3 has, as timers, as hardware interrupts, and supports direct operation with the ports (albeit using this requires a deep understanding of the microcontroller's operation), still, most RT-DSP algorithms (like ring buffers, filtration by convolution in the TD and by FFT at FD) can be implemented by using TinkerCad. Figure 5 presents an exemplary slide from the student' report, demonstrating that students have built an electronic layout containing "plain" signal generator, Arduino UNO R3 board, and "fast and dirty DAC" built from resistors and OA (it was necessary, because Arduino UNO R3 has no DAC).

Important that Tinker Cad can simulate as analog as digital electronics. Hence Arduino UNO R3 can be programmed by writing standard C code by using integrated into TinkerCad an Arduino simulator. An educator can see the code and check its execution, including graphs of the signals created by the Arduino simulator. Specifically, this example demonstrates the important DSP algorithm "normalized correlation". The goal of the students, in this case, was to implement this algorithm without RT-DSP libraries, and, evaluate the time limits of the created application.
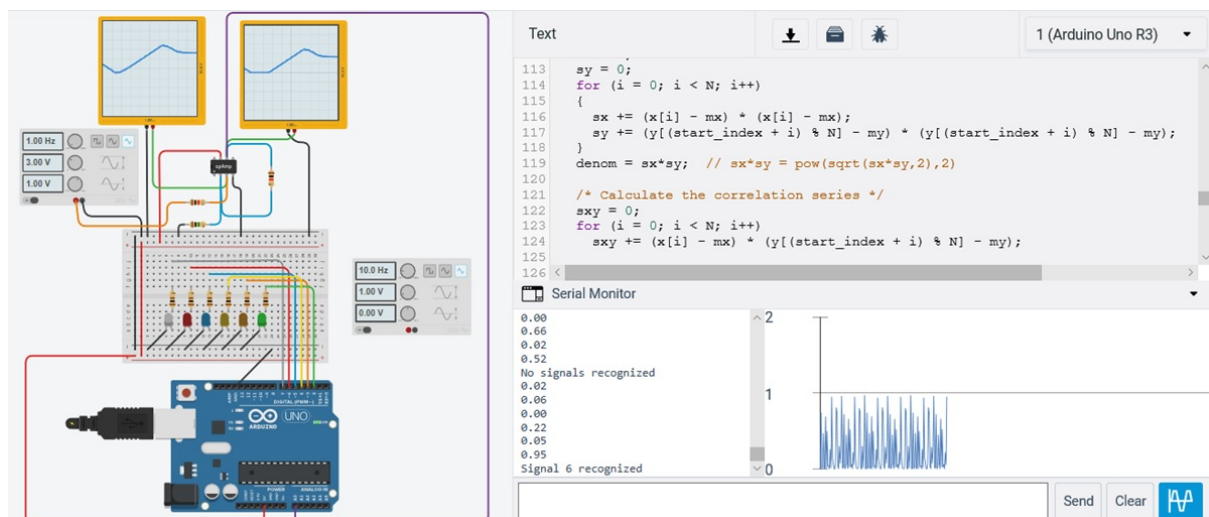

Figure 6. Example of TinkerCad simulation

Providing RT-DSP course in the Hybrid semester

It so happens that semester 2021-10 – 2022-03 was started as an online semester, but after three weeks, it became a hydride semester: lectures, exercises, and Micro Projects presentations were provided in-campus, but students had the right to stay at home and continue to use cloud services including ZOOM and Email. In the "normal" semester students were required to present in-class MicroProject #1 (which deals with different RT-DSP algorithms for different pairs of students). However, with the non-strict presence rules of the hybrid semester, a plurality of situations was created: for example, one student of the pair was physically present in class during the presentations, whereas the second student participated by using ZOOM. This situation created a number of logistics problems, and, thus, some logistics changes were needed to be done in order to ensure fair and non-biased grading, as for the students who were physically present during the lectures, exercises, and presentations, as for the students who participated remotely. In any case, in the frames of this course, each group of students prepared three Home Work reports and two Micro-Project presentations. Additionally, only students that were physically present in the class were able to participate in the short nonobligatory micro-exams.

Results and Conclusion

In "normal" times electronic students enrolled in the course "Real-Time Digital Signals Processing" (RT-DSP) use real hardware: dedicated development boards containing microcontrollers and additional electronic components, and dedicated software (installed on the personal computer) to program those development boards.

Because of COVID 19 limitations, the distribution of development boards and dedicated software to the students became problematic. Even more, technical assistance in case of installation and/or hardware problems became practically impossible. Design and testing of electronic systems is an important part of Electronic Engineer's education; hence it was decided to adapt the RT-DSP course to distant learning by using a cloud-based TinkerCad simulator. It was found that by using some tricks and simplifications, reasonably sophisticated electronic projects can be implemented. Additionally, it was found that the cloud-based TinkerCad simulator can run in parallel with the ZOOM session, while the Internet speed of the lecturer's computer is at least 50 Mbps – which means that providing frontal lectures and exercises (containing plain PowerPoint presentations and live simulations demonstrations) is feasible, at least for the groups of 30 students.

After evaluating and grading homework and micro-projects created by students in the frames of this course, it was concluded that the professional level of the designed systems is good enough to continue this course in the frames of distant learning, in case "normal" options are not available.

Unfortunately, collecting students' opinion (students' feedback) about specific courses under COVID19 restrictions become problematic, as practically, as ideologically – in the end, only a small number of students provided this feedback. Hence, this parameter cannot be used to evaluate the quality of the selected cloud approach. However, privately, after getting final grades, many students claimed that this specific course could be provided by using cloud services and by using TinkerCad simulations.

Some features of TinkerCad simulator were so good that it is planned to use the TinkerCad simulator (while it is free and cloud-based) even in "normal" semesters in parallel with "real" development boards.

References

Auer, M. E., and Gallent, W. (2000). *The "Remote Electronic Lab" as a part of the telelearning concept at the Carinthia Tech Institute*. Proceedings of the ICL2000, Villach/Austria, 28/29.09.2000.

Diya, J., Chavla, P., Chander, H., Rawat, T.   (2017) *A multipurpose toolkit for teaching DSP in an undergraduate course*. Computer Applications in Engineering Education. 06 April 2017.

Hoffbeck, J. (2012). *Enhance your DSP Course with These Interesting Projects*. 2012 ASEE Annual Conference & Exposition.

Kosolapov S., and Sabag, N. (2019) *Logistics of Electronic Home Project-Based Course*. International Journal of Information and Education Technology", vol. 9 no. 6, pp. 445-448. 2019

Kosolapov S., (2022) *Time-Effective Logistics of Project-Based Course Electronic Instrumentation*. eLmL2022: The Fourteenth International Conference on Mobile, Hybrid, and On-line Learning, Proceedings, (35-41), PORTO, Portugal, July 26-30, 2022.

Nikolic, S, Vial, P., Ros, M., Stirling, D., and Ritz, C*., (2015) *Improving the Laboratory Learning Experience: A Process to Train and Manage Teaching Assistants*, IEEE Transactions on Education, vol. 58, no. 2, 2015.

Yi, Z., J. Jian-Jun, and Shao-Chun, F., (2005) *A LabVIEW-based, interactive virtual laboratory for electronic engineering education*, International Journal of engineering Education, vol. 21, no. 1, pp. 94-102, 2005.