

# VFST: Virtual and fully software based toolchain for PC interfacing education and research

Mokhtar A. A. Mohamed<sup>1,2</sup> 

<sup>1</sup>Computer Science & Engineering Department, Faculty of Electronic Engineering, Menofia University, Al Minufiyah, Egypt

<sup>2</sup>Computer System Engineering Department, Faculty of Engineering, MSA University, 6th of October City, Egypt

## Correspondence

Mokhtar A. A. Mohamed, Computer Science & Engineering Department, Faculty of Electronic Engineering, Menofia University, Al Minufiyah, Egypt.  
Email: [mokhtar.mohamed@el-eng.menofia.edu.eg](mailto:mokhtar.mohamed@el-eng.menofia.edu.eg)

## Abstract

In computer science and engineering education, teaching PC interfacing courses and laboratory experiments are strongly required. PC interfacing is the field of developing or simulating real/virtual solutions for exchanging data/commands between PC and the outer world of sensors and actuators. Recently, virtual experiments and Labs have been adopted by engineering educators to preclude the physical presence of students and tutors and to provide effective skills acquisition and hands-on experience in a virtual manner. High-cost LabVIEW from National Instruments company is the dominant platform to develop, execute, and animate virtual PC interfacing experiments similar to real ones. This paper presents a toolchain, the course contents, and virtual experiments that cover many different PC interfacing technologies such as UART, SPI, RS232, USB, SD card interfacing, and wireless GSM/GPRS. The virtual nature of VFST toolchain refers to enabling the students to design, animate, and run their prototypes or solutions inside only PC programs without the need to build physical or real ones until making sure of their prototypes' functionality. The main aim of the VFST platform is to make students able to develop C# based applications and to implement solutions with an aesthetic similar to LabVIEW capabilities to become a low-cost alternative of it. VFST consists of a group of free-of-charge and third-party software programs such as VSPD, Virtual USB driver, WinImage, and Real-Term alongside with Proteus simulator and C# programming language. This paper presents qualitative and quantitative evaluations of the benefits and effectiveness of VFST. Findings are collected via feedback responses using an online questionnaire survey. These findings clarify that the student cohort is, in general, supportive that VFST Lab's content and virtual experiments have a considerable effect on their career after and before their graduation.

## KEYWORDS

electronics education, engineering education, PC interfacing, virtual labs in CSE

## 1 | INTRODUCTION

Supporting the application of virtual laboratory tool-chains in computer science and engineering (CSE) programs has been a recent trend among many academics from different universities and higher educational institutions [6, 12]. One of the accreditation criteria of the Accreditation Board for Engineering and Technology (ABET) requires CSE programs to consolidate theory through practice to increase the hands-on experience of students [13, 14, 22].

Academics and syllabus designers are divided into two groups. The first group adopts the idea of using virtual laboratories and/or simulators in delivering theoretical knowledge, virtual experiments (VE), and problem-solving skills to avoid the burden of functional and timing verification of real hardware components [9, 16, 17, 28, 29, 30, 35, 36, 42]. VE and/or simulators can be used by students anytime and anywhere, allowing them to adapt to the learning speed of each student. The other group adopts physical laboratory experiments because they claim that remote or virtual laboratories will limit the hands-on experiences and skills of students due to the missing reality [26, 28].

PC interfacing refers to the process of exchanging sensory data/actuator commands between a computer and one or more external peripheral devices by using common and famous wired/wireless technologies. Wired technologies include universal asynchronous receiver/transmitter (UART), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), Controller Area Network (CAN), Ethernet (TCP/IP), and Universal Serial Bus (USB). Wireless technologies include Wi-Fi, Zigbee, Bluetooth, radio frequency (RF) and RF identification (RFID) technologies, and Global System for Mobile Communications (GSM)/General Packet Radio Service (GPRS). A plethora of computer types ranging from large computers (e.g., workstations and PCs) to small embedded computers (e.g., Raspberry Pi and ARTIK™ 053 starter kit) can be used on the host side. The design of any external peripheral devices requires the use of special purpose and small microcontrollers, such as PIC, AVR, ARM, and 8051. Consequently, academic institutions should provide their laboratories with a huge and diverse collection of real hardware components, modules, and/or devices. Such diversity of apparatuses will lead to high cost, the physical presence of students in laboratories, and the teaching of multiple software platforms to develop driving programs for such apparatuses. Nowadays, many universities and colleges prohibit the physical presence of students because of the COVID-19 pandemic. They encourage academicians to conduct online courses and virtual laboratories, which may be difficult when teaching PC interfacing.

To achieve virtuality when teaching PC interfacing laboratories, syllabus designers use either LabVIEW or Proteus simulators to emulate the entire hardware systems of PC interfacing laboratories in academia [22, 24, 28, 42, 43]. LabVIEW stands for “Laboratory Virtual Instrument Engineering Workbench.” It is a productive development environment for designing custom applications that interact with real data and signal acquisition devices by applying special programs called virtual instruments (VIs). This software is used by many academicians to provide graphical user interface (GUI) applications to PCs by using the drag-and-drop graphical components and textual programming approach (i.e., G-code) for test, measurement, and control applications. VIs differ from traditional textual programming environments, such as C#, Java, Visual Basic, and MATLAB. The VIs of LabVIEW are tools that can be used for acquiring, analyzing, displaying, and storing data as database reports in PCs. LabVIEW can also represent data in the form of charts, graphs, tables, 3D controls, 3D graphs, picture control, gauges, and meters. In short, LabVIEW is adequate and easy to use for nonprogrammers.

LabVIEW is owned by National Instruments (NI), and thus, embedded developers must pay to use its advanced libraries. The closed-sourced license of LabVIEW reduces the passion of students for designing and building hardware circuits for PC interfacing. Students and embedded developers are unable to use LabVIEW's G-code on their hardware solutions, and thus, they are forced to use expensive interfacing products offered by NI, such as NI6008 USB-based DAQ, NI6009 USB-based DAQ (price starts at US\$700), and wireless USRP-2922 bundle (price starts at US\$7458).

The Proteus simulator has been recently considered a powerful, easy-to-use, and productive electronic design automation tool. It has been extensively used in universities and engineering colleges because students can use it for designing (i.e., schematic capturing), simulating, animating, and verifying different types of passive or active electronic circuits, and then converting them into real circuits [21, 40]. The powerful virtual system modeling (VSM) of Proteus enables students and researchers to simulate the hardware and firmware of any microcontroller, including input/output ports, timers, interrupts, UART, and other built-in peripherals inside each supported microcontroller, such as PIC, AVR, and/or ARM. The Proteus simulator has approximately 785 microcontrollers and thousands of passive common parts. It allows the import of more components by using the BSD format. Recently, Proteus has been extended to enable visual designers to simulate complete embedded platforms, such as Arduino, Raspberry Pi, and IOT Builder, to design, monitor, and control remote systems

from a simple mobile application. Visual designers use a combination of their VSM, flowcharts, and virtual hardware, resulting in an integrated and intuitive development environment for Arduino, Raspberry Pi, and IOT builder. The functionality of the basic simulator to provide a full range of graph-based analyses, such as analog/digital transient analysis, frequency analysis, Fourier analysis, and noise analysis, is extended.

All models of Proteus VSM processors and microcontrollers can run a binary file that is developed and produced by supported third-party assemblers or compilers, such as MikroC compiler for PIC, AVR, and ARM, Hi-Tech software, CSS, MPLABX, RaisoNance, Keil, Arduino, AVR Bascom, and ImageCraft. One of the important features of Proteus is its 3D viewer, which allows users to look at their designed circuit from a 3D viewpoint, similar to a real one. In summary, students feel like they have a private electronic store at their disposal when working with the Proteus simulator, and they are working with real authentic electronic components and real microcontroller-based, Arduino-based, and Raspberry Pi-based systems.

However, despite all the capabilities described in the aforementioned section, Proteus remains inadequate for meeting the modern requirements of PC interfacing and presenting the results to the end user. LabVIEW provides an attractive approach for displaying sensory data into 3D-based graphics to the end user, introducing the isolation of users from back-end interfacing circuits. LabVIEW provides user-level abstraction during the exchange of sensory data or actuating commands between a desktop application and its premanufactured interfacing solutions and systems.

The current study addressed in detail the mature planning and design of a syllabus and VE for PC interfacing between a virtual device(s) by using Proteus with desktop applications developed using a general-purpose and textual C# programming language. This syllabus is called VFST, which is short for Virtual Laboratory and an Education Syllabus. In this syllabus, the student is responsible for developing firmware for any participating microcontroller and a complete GUI and textual source code for each desktop application in this context. A group of free-licensing and third-party programs can be used as the connection between Proteus simulator and C#-based applications to provide virtual communication channels, such as the virtual serial port driver (VSPD) from Eltima (a free version can be downloaded directly from its official website), virtual USB driver from Labcenter (free and comes with the same Proteus package.), WinImage program from Gills Vollant software (free-licensing program that can be downloaded directly from its website), and RealTerm program (free-licensing program that can be downloaded directly from the

SourceForge website). These third-party programs with the Proteus simulator and C# programming language comprises a toolchain dedicated for the VFST syllabus. The primary objective of VFST is to enable students and researchers to develop desktop applications that can monitor sensory data from different sensor types, send operational commands for different types of actuators, and implement solutions with an aesthetic appearance similar to LabVIEW capabilities. In the future, students can further pursue complex systems by involving Internet of things (IoT) and artificial intelligence platforms within their prospective applications.

The outline of this paper is as follows. Section 2 provides a comparison among numerous articles that introduce different methodologies and curricula on the consolidation between theory and practice during the teaching of CSE courses. Section 3 presents the problem statement and primary objectives of VFST. Section 4 describes the entire software platform and toolchain of our proposed VFST. The diagrams and C#-based desktop application for each VE of VFST are detailed in Section 5. The research methodology and qualitative and quantitative evaluation results are discussed in Section 6. Finally, a conclusion is drawn in Section 7.

## 2 | RELATED WORK

This section sheds light onto numerous articles that introduce different methodologies and curricula on the consolidation between theory and practice during the teaching of CSE courses. CSE Specialization, in particular, and Electronics Engineering, in general, focuses on the technical contents of any CSE syllabus. Accordingly, most of the included studies revolve around presenting and discussing technical content and how enrolled students respond to such content. Consequently, teaching methodologies, processes, and activities that traditional academicians are interested in are excluded from CSE specialization.

The articles present and support virtual laboratories (VLs), physical laboratories (PLs), or hybrid laboratories (HLs). Table 1 lists these articles. The issues included in this review are the general application field of each article and its type (i.e., virtual, remote, physical, or hybrid). In addition, the type of interface technology and hardware tools used are provided in this review.

## 3 | PROBLEM STATEMENT

The problem-based learning (PBL) approach is considered one of the most effective education techniques for developing 21st century capabilities [4, 5]. PBL focuses

TABLE 1 Literature review of consolidation between theory and practice in CSE programs

No	Reference	Application	Lab type	Learning type	Used interface protocol	HW tools	SW platforms
1	Heuiseak Lim et al. 2009 [29]	Embedded System Education	VL	PBL + TBL	Not Available (N/A)	N/A	Skyeye simulator
2	Ali Buldu et al. 2007 [15]	USB Kit for digital I/O Application	PL	PBL	USB	<ul style="list-style-type: none"> <li>PIC16C765</li> <li>PC</li> </ul>	Visual Basic Desktop App
3	Yasser Mechanna et al. 2006 [33]	Microprocessor, DSA, Microcontroller Course Education	PL	Not Clear	UART	dsPIC30F™	N/A
4	Peter Tierman et al. 2010 [42]	Control and DAQ application	VL	PBL + TBL	N/A	N/A	LabVIEW
5	Edward T.-H. 2015 [19]	Embedded O.S	N/A	PBL	RS232	<ul style="list-style-type: none"> <li>Panda Board</li> <li>PC</li> </ul>	CALEE
6	Kerem KucuK 2017 [28]	Wireless communication education research	HL	PBL	Wi-Fi using NI USRP 2921	<ul style="list-style-type: none"> <li>NI USRP 2921</li> <li>PC</li> </ul>	<ul style="list-style-type: none"> <li>LabVIEW</li> <li>MATLAB</li> </ul>
7	Savas Sahin and Isler 2013 [37]	Control and automation education	HL	PBL	<ul style="list-style-type: none"> <li>RS232</li> <li>USB</li> <li>TCP/IP</li> <li>Parallel</li> <li>(IEEE 1248)</li> <li>(D.25)</li> </ul>	LabVIEW	
8	Slim Chtourou et al. 2018 [18]	Embedded system education	PL	PBL	Bluetooth-based USB to connect with mobile App	<ul style="list-style-type: none"> <li>IOIO electrical board</li> <li>Cell phone</li> </ul>	APP inventor
9	Andres M. Vargaset et al. 2018 [24]	Control engineering education	PL	PBL	USB	<ul style="list-style-type: none"> <li>dsPIC30F4012</li> <li>PC</li> </ul>	<ul style="list-style-type: none"> <li>Lubuntu</li> <li>RTAI</li> <li>Scilab</li> <li>XCOS</li> </ul>
10	Ismail Uyanik et al. 2018 [43]	Control systems	PL	PBL	Arduino USB	Arduino Uno	<ul style="list-style-type: none"> <li>MATLAB</li> <li>SimuLink</li> </ul>
11	Shaftee Mayoof et al. 2020 [31]	E-Learning of circuits	RL + VL	BPL	N/A	Different virtual hardware for each experiment	Circuit cloud over internet (Amazon)
12	R. B. Coruk et al. 2020 [20]	Communication System	VL	BPL + TBL	N/A	N/A	Simulink

TABLE 1 (Continued)

No	Reference	Application	Lab type	Learning type	Used interface protocol	HW tools	SW platforms
13	T. Achalakul et al. 2004 [1]	Sattelite Image Fusion and Analysis	VL	N/A	HTTP protocol	N/A	DCOM product from Microsoft
14	Our proposal (VFST)	PC Interfacing	VL	BPL	<ul style="list-style-type: none"> <li>• UART</li> <li>• RS232</li> <li>• USB</li> <li>• SPI</li> <li>• Wireless GSM/GPRS</li> </ul>	Different virtual hardware for each experiment	<ul style="list-style-type: none"> <li>• MikroC</li> <li>• C#</li> <li>• Proteus</li> <li>• Virtual USB</li> <li>• VSPD</li> <li>• WinImage</li> <li>• Real-Term</li> </ul>

on boosting critical thinking, problem solving, information and media literacy, cooperation, leadership and teamwork, innovation, and creative skills that students can gain during and after each learning milestone. Teaching PC interfacing or similar syllabi in the CSE discipline faces two major challenges when it is accompanied by the PBL approach. The first problem is related to technical contents, including laboratory type and cost of hardware (HW)/software (SW) apparatus. The second problem is related to measuring factors of educational goals and the effects of PC interfacing syllabi. These factors include students' individual achievement (SIA) [2], students' cooperation outcome (SCO) [7], students' learning skills (SLS) [3, 8], and students general satisfaction (SGS) [3, 8].

As indicated in Table 1, most related studies focused on producing and presenting only technical content and experiments on similar topics related to PC interfacing. Most studies adopted LabVIEW and its accompanying NI interface boards to construct their laboratories. In the beginning of establishing our PC interfacing laboratory, we found that the cost of a LabVIEW-based laboratory is extremely expensive, reaching more than US\$130,000. The total cost includes LabVIEW program (US\$6365), 15 modules of USB-based NI 6009 DAQ board for USB interface only (US\$13,500), and 15 bundles of USRP2920 wireless board for Wi-Fi interface only (US\$111,870). In addition, most related work did not verify the effectiveness of the learning methodologies used during the teaching of their laboratory contents, such as the SIA, SCO, SLS, and SGS factors. VFST is sensibly designed to overcome the two aforementioned problems.

## 4 | DEVELOPMENT SOFTWARE TOOL CHAIN OF VFST

The proposed VFST is built by employing several programming platforms and software tools. They are carefully chosen to achieve the desired learning outcomes and to create an adequate virtual learning atmosphere. Some of these software platforms are commercial versions while others can be obtained as a free version from its website. The following sections will list briefly the description of each software platform/tool.

### 4.1 | Visual Studio 2019 from Microsoft

Visual Studio 2019 provides a rich and integrated development environment for creating attractive applications for Windows, Android, and iOS, along with modern web applications and cloud services. General-purpose C#

programming language is suggested to be used in this coursework for building and developing simple and complex Windows-based desktop applications in the context of PC interfacing. These applications are used for monitoring/controlling purposes with external virtual devices. Each student can install and activate Visual Studio 2019 as free software by using his/her official academic email because of the licensing agreement between Microsoft and the Supreme Council of Egyptian universities.

## 4.2 | MikroC compiler from MikroElektronika

The MikroC Pro for PIC compiler is a powerful and feature-rich development tool for PIC microcontrollers. It is designed to provide embedded developers with simple libraries for developing firmware applications for embedded systems without compromising performance or control.

MikroC Pro provides successful match featuring, highly advanced IDE, an ANCI C compliant compiler, a broad set of hardware libraries (i.e., 1045 library functions), comprehensive documentation, lifetime license with a commercial cost of US\$270, and plenty of ready-to-run examples (i.e., 168 code examples). In addition, it currently supports more than 808 PIC microcontrollers. In summary, MikroC Pro for PIC can be used to dramatically accelerate the development of driving programs (i.e., firmware) for different types of PIC microcontrollers to achieve data acquisition, memory, display, conversion, and communication with/from different hardware components.

## 4.3 | Virtual USB driver and Proteus simulator

A virtual USB driver is a small program that is integrated into the Proteus package. It is used to “deceive” a PC that a virtual USB peripheral schematic, which is entirely designed in the Proteus simulator, is plugged into the PC similar to real USB devices. The integration of a virtual USB driver into Proteus enables students to test and model all request/replay transactions between such virtual USB schematic and the levels of Windows USB drivers. This integration supports USB communication with the different classes (i.e., protocols) of USB devices, such as the human interface device class (HID), communication device class (CDC), or mass storage device class (MSD). The Proteus simulator is described in detail in Section 1.

## 4.4 | Virtual serial port driver (VSPD) from Eltima

VSPD is a powerful program that is designed specifically for those who develop, test, or debug serial port software and hardware. This program provides any Windows OS with as many virtual COM (i.e., serial) interfaces as the developer needs. In this coursework, VSPD aims to create a virtual pair of COM ports that can be used instead of real COM ports for serial communication between virtual serial schematic in Proteus and PC. VSPD is particularly convenient when all real serial ports are already occupied. The standard edition of VSPD is free.

## 4.5 | WinImage program from Gills Vollant software

WinImage is a full-fledged disk-imaging suite for easy creation, reading, and editing of many image formats and file systems, including FAT, ISO, NTFS, and Linux. With WinImage, students can recreate an image on the hard drive or other media, view its contents, extract files, add new files and directories, change the format, and defragment an image in a similar manner when working with real hard drives or SD memories. The primary objective of the WinImage program within this coursework is to create an SD image and integrate it into Proteus to simulate the interfacing experiments of virtual SD schematic applications. These experiments focus on storing/retrieving sensory data to/from SD memory image, particularly in monitoring applications. Moreover, the WinImage program can be obtained for free from its website.

## 4.6 | RealTerm: Serial/TCP terminal

RealTerm is an engineering terminal program that is specially designed for capturing, controlling, and debugging binary and other difficult data streams. In this coursework, RealTerm is used mostly to ensure that ATcommands for the adopted GSM/GPRS, Wi-Fi, and/or Bluetooth modules are generated correctly as a virtual tool before the actual execution of such AT commands on the aforementioned modules. ATcommands is a list of text-based commands with a predefined format that is required to run an available and specific task of such modules. ATcommands are defined and specified by the manufacturer of such modules and included in a manual sheet for each. The RealTerm program is free of charge.

## 5 | VE OF THE VFST

This section presents a brief description of six VEs. The descriptions include the objectives of each VE and schematic diagram for each VE.

### 5.1 | VE 1: Interfacing PC with a PIC-based device using serial communication

The main objective of VE1 is to understand the basic concepts and C#-based libraries of serial communication. Another objective is to enable students to develop a simple desktop application for sending control commands from a PC to a PIC-based external device. The schematic diagram of VE 1's external device is shown in Figure 1. See Supplementary Information: Video 1.

### 5.2 | VE2: Sending sensory data from PIC to PC using serial communication

The primary objective of VE2 is to enable the student to develop the MikroC code for PIC 18F452 and send temperature sensory information to the PC via the serial RS232 protocol. They are connected in a diagram as shown in Figure 2. Another objective is to familiarize the student with the concept of A/D conversion and apply this concept during the temperature calculations of an LM35 sensor. All the hardware components of VE2 are virtual. They include PIC18F452 [32], MAX232 chip [41], virtual "COMPIM" connector, and LM35 temperature sensor. The wiring method described in [10] is used to connect the MAX232 chip with virtual "COMPIM" connector. See Supplementary Information: Video 2.

### 5.3 | VE3: Simple paradigm for USB communication

The USB has become the most successful PC interface in recent years [11]. Every recent PC has USB ports that can connect to keyboards, mice, scanners, cameras, printers, SSD drives, and other modern devices. USB is reliable, fast, versatile, power-conserving, inexpensive, and supported by major OS such as Windows, Linux, and so forth. The new USB 3.0 and 4.0 SuperSpeed bus indicates that USB is likely to continue to dominate as the interface of choice for an ever-expanding selection of peripherals. The primary objective of VE3 is to enable students to learn new concepts about the complex USB protocol and implement a simple paradigm to test USB communication between a simple Proteus and PIC-based

external USB device and one with built-in programs within the MikroC editor, called "HID Terminal." The hardware components of VE 3 are virtual, and they include the PIC18F4550 microcontroller and the virtual "USBCONN" connector in Proteus. They are connected in a diagram as shown in Figure 3. See Supplementary Information: Video 3.

### 5.4 | VE4: Complete experiment for bidirectional PC/USB interface

The primary objective of VE4 is to enable students to build a complete USB-HID device that can communicate with a C# window form to achieve bidirectional USB communication. VE4 can be considered the first step to help students progress on to more complex USB-HID devices and solutions. A new C# library, called the "USB Generic HID Communication (USB-GHC)," should be employed. USB-GHC library version 2.0.0.0 [25] is a free software library that can be redistributed and/or modified under the terms of the GNU public license [23]. The operation of the USB-GHC library relies on calling for and dealing with four window DLL functions: setupapi.dll, kernel.dll, user32.dll, and hid.dll. These DLL files are responsible for all USB functions in Windows platforms, such as device discovery and identification, sending and receiving USB transactions, handling USB event notifications (i.e., attaching and detaching USB devices), and/or communicating with Windows' USB generic HID driver [11]. All the hardware components of VE 4 are virtual. They include PIC18F4550 microcontroller, "USBCONN" connector, one LED, and a push button switch. The diagram of all these connected components is shown in Figure 4.

### 5.5 | VE5: Interfacing SD/MMC memory cards with PIC

The primary objective of VE5 is to familiarize students with the interfacing paradigm between one of the PIC18F families (i.e., PIC18F4550) and SD/MMC memory cards. VE5 will guide them in putting files and folders (i.e., directories) in SD/MMC cards, write/read any data from these files, and transitioning between folders inside SD/MMC cards. All the hardware components of VE5 are virtual and provided by the Proteus simulator. They include PIC18F4550 microcontroller, two LEDs, a virtual SD card connector, and a DHT11 temperature and humidity sensor. The diagram of VE 5 is shown in Figure 5. See Supplementary Information: Video 4.

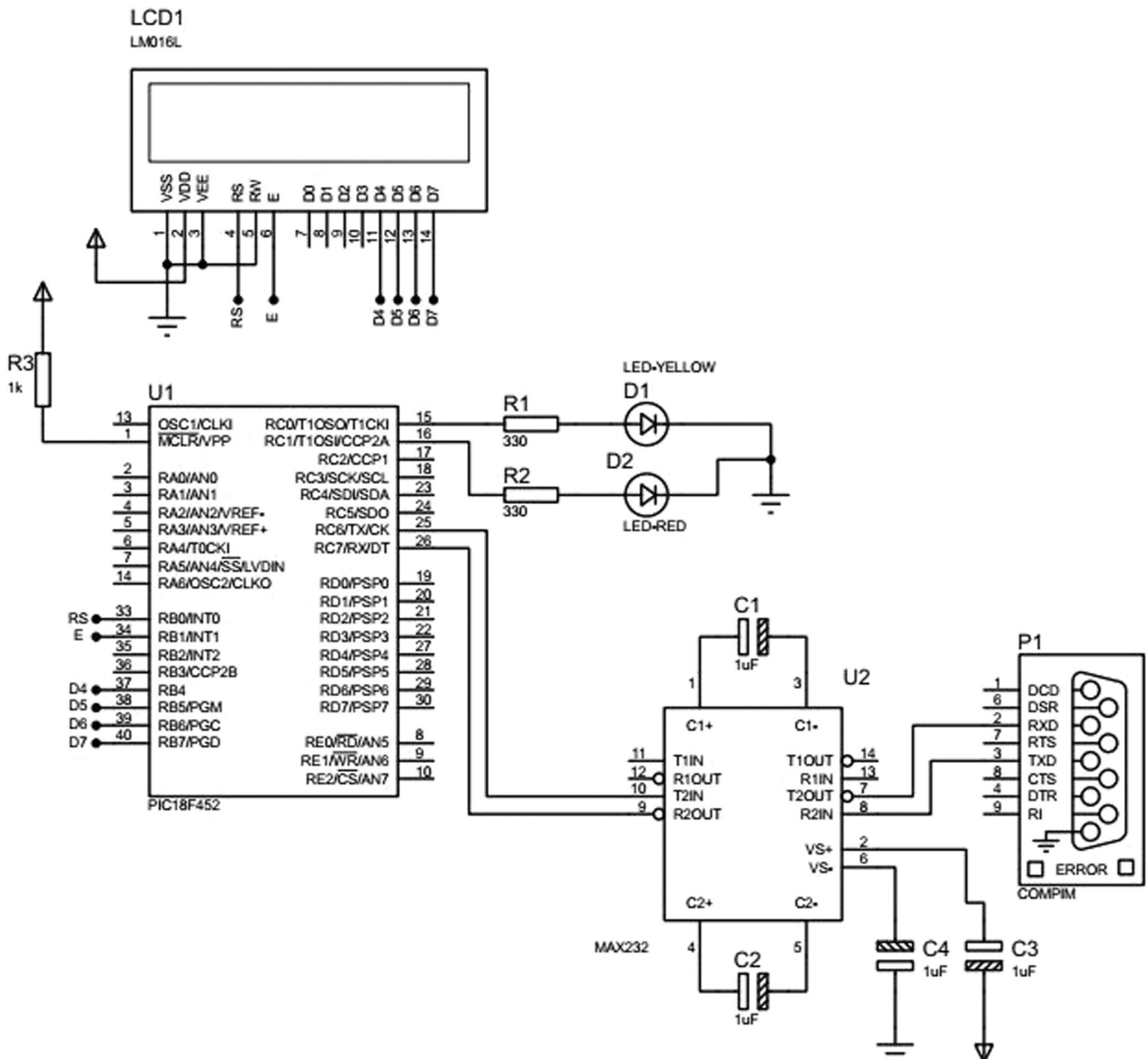


FIGURE 1 The schematic diagram of virtual experiment 1

## 5.6 | VE 6: Interfacing PC with wireless GSM/GPRS modem

The primary objective of VE 5 is to familiarize students with building wireless-based applications by adopting the capabilities of GSM/GPRS modems, such as the SIM800L modem [39]. The GSM standard protocol is used for handling activities, such as voice calls and short message service (SMS) exchange. The GPRS protocol is used for sending and receiving data packets (such as the Internet) via cellular network infrastructure. The use of GPRS expands the functionality of these small modems to

exchange data remotely with different and current technologies, such as emails, IoT, and cloud computing platforms. The hardware components in real experiments include the SIM800L modem [39], USB-to-TTL converter [38], and DC-to-DC converter. In the case of executing VE 6 virtually, the student does not need real components and can connect between his/her C# window form and the RealTerm program. Real-Term program is used for displaying ATcommands and string-based responses that are exchanged between itself and the C#-based window form on the PC. See Supplementary Information: Video 5.

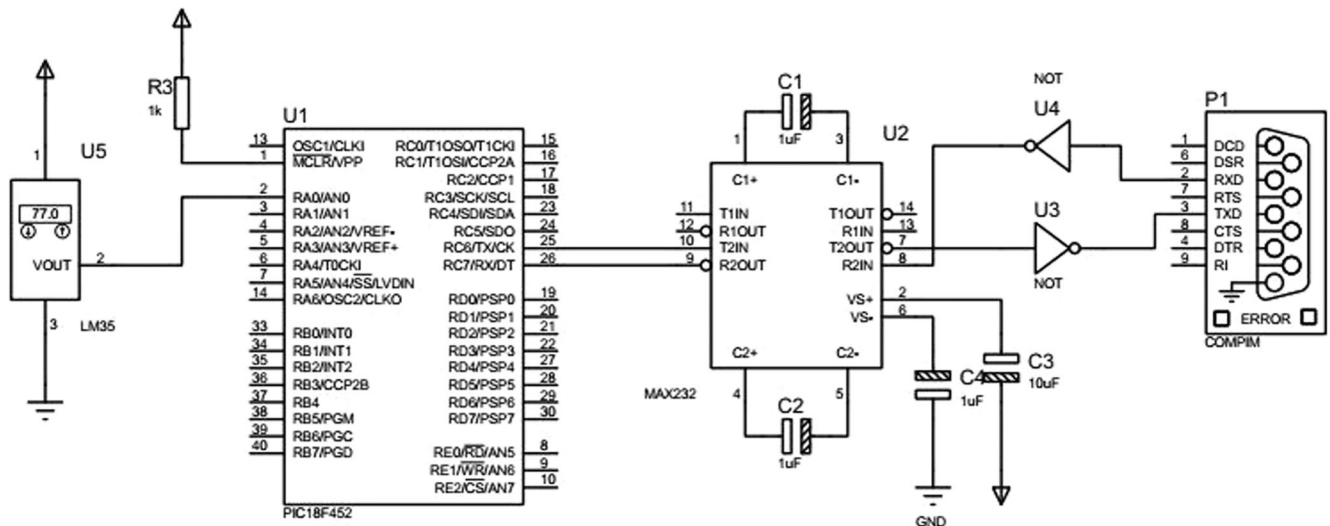


FIGURE 2 The Schematic diagram of virtual experiment 2.

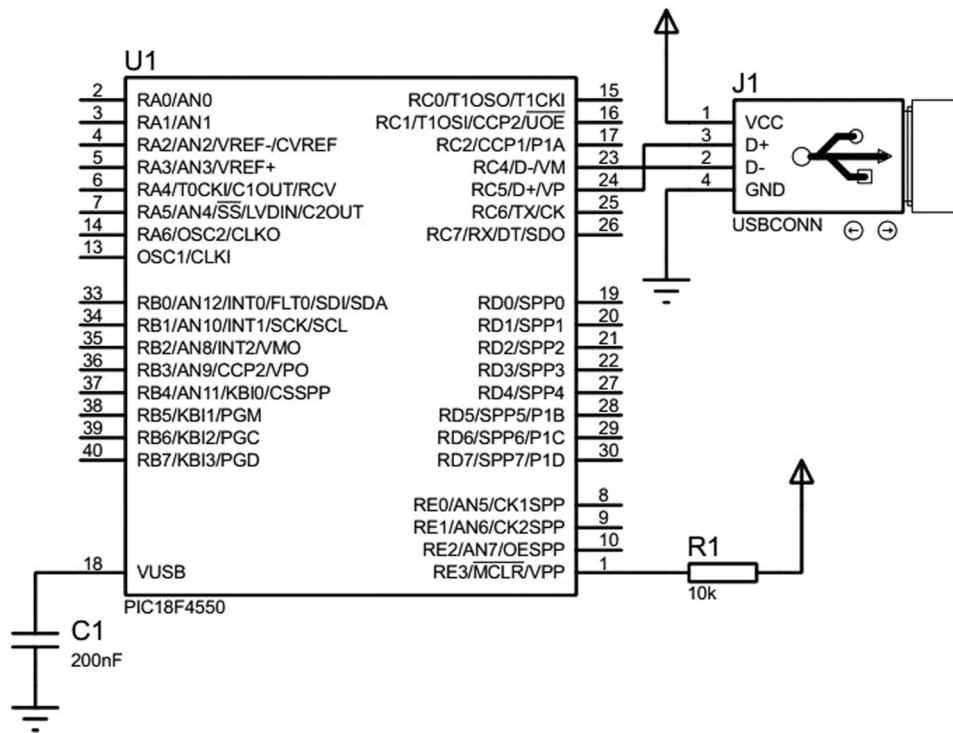


FIGURE 3 The schematic diagram of virtual experiment 3

## 6 | IMPLEMENTATION METHODS OF VFST

During Teaching development of VFST, each student is responsible for attending lectures/labs and performing the teacher-directed individual study (TDIS) stage aligned with the self-directed learning stage (SDLS). TDIS refers to the process of acquiring basic knowledge of any topic directly and under the supervision of teachers during traditional and/or online lectures/labs.

TDIS has been a subject of controversy in the field of CSE and engineering, with many academicians claiming that it should be used until the present, particularly in technical disciplines [34, 44]. SDLS is required to enable students to acquire additional knowledge by searching the Internet or other online resources.

In VFST, the workflow of the PC interfacing course is regulated into five consecutive stages, as shown in Figure 6. These stages are as follows: acquiring basic knowledge (i.e., TDIS), presenting and discussing

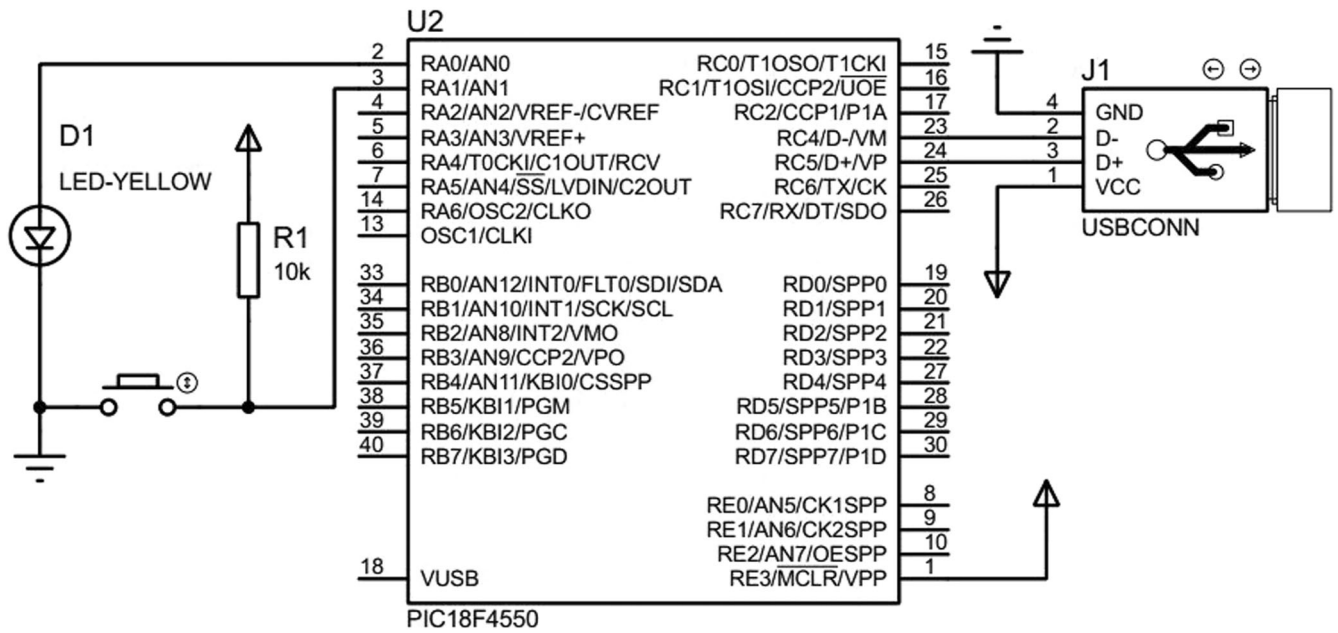


FIGURE 4 The schematic diagram of virtual experiment 4

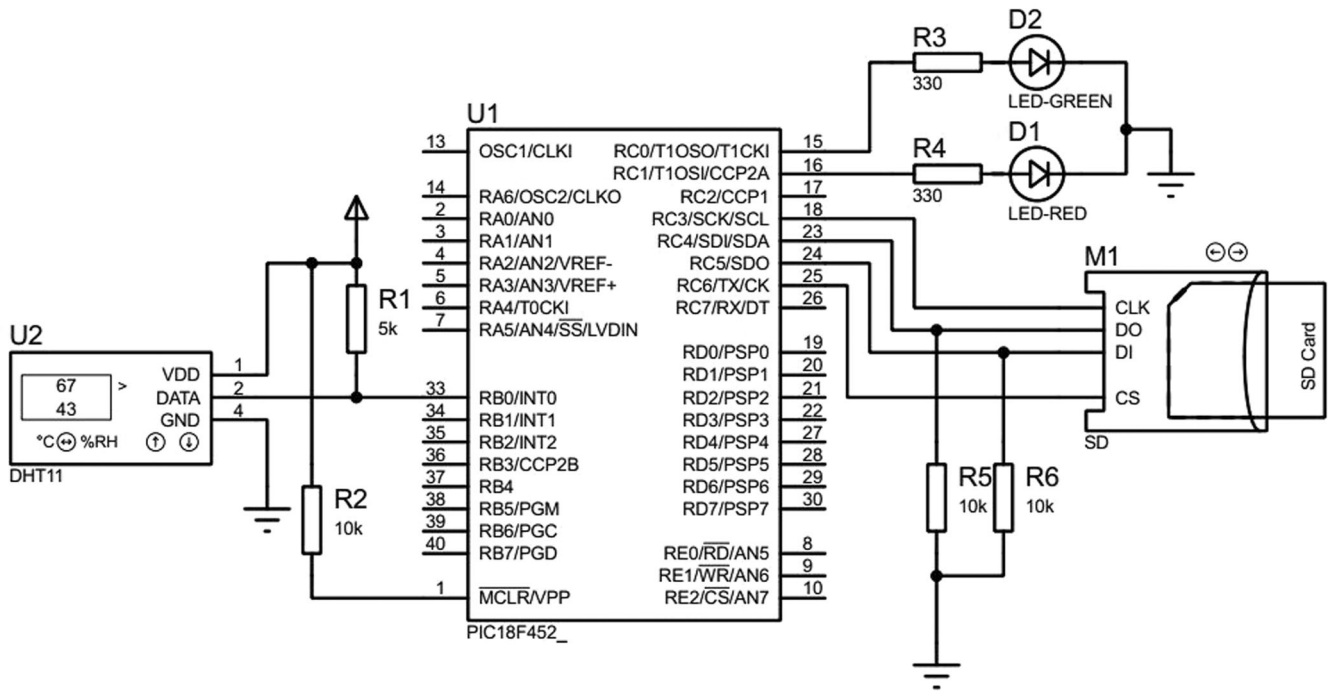


FIGURE 5 The schematic diagram of virtual experiment 5

problems, self-learning skills (i.e., SDLs), oral presentation of problems, and generation of simple/complex projects into real prototypes. The VFST course consists of 2 h of theoretical lecture and 2 h of practical lab per week. Students can acquire each VE content in two lectures (i.e., 2 weeks). Consequently, the VFST course will consume 15 weeks in total. 12 weeks to complete

technical content and 3 weeks for quantitative and qualitative assessments. The outlines of the VFST course are described in Figure 7.

When teaching each context of VFST, the involved teachers adopt most traditional educational methodologies, such as giving lectures in classrooms, providing online filmed lectures, observing students in laboratories,

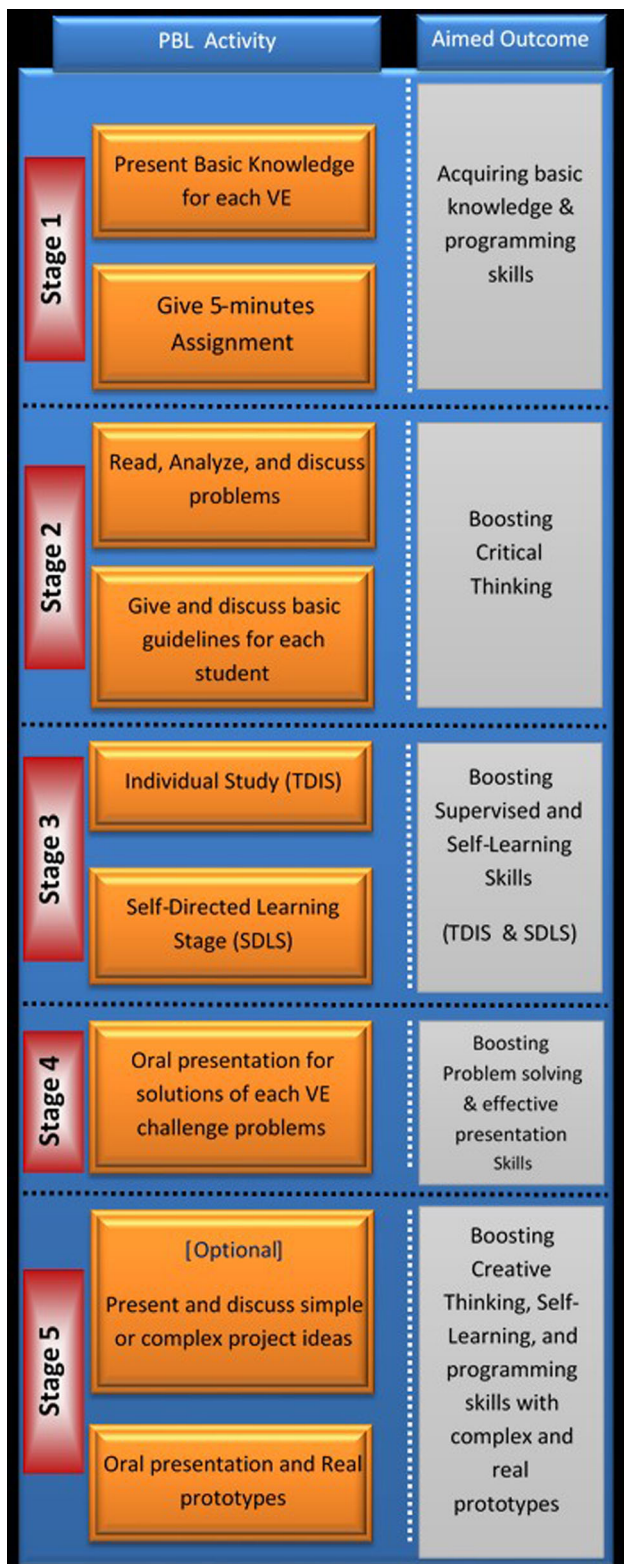


FIGURE 6 Five stages of PBL Workflow implementation in VFST. PBL, problem-based learning; VFST, virtual and fully software.

giving them a 5-min quizzes before the end of each lecture, and giving them large-scale assignments as homework. Traditional written exams, practical exams, and oral presentations are used as quantitative

assessment methods, while interviews, quizzes, and oral presentations are used as qualitative assessment methods for student evaluation. We did not record the statistics of such traditional teaching methodologies, However, we will consider them in the future.

## 6.1 | PBL methodology of VFST

The PBL methodology of VFST is applied to two challenging levels. The first level is related to giving technical problems (i.e., long-time lab Quizzes and large-scale homework assignments) to the students after finishing each VE content. Thereafter, each student must perform TDIS and SDLS activities to find optimal solutions for such problems. The second level revolves around generating and converting optionally simple or complex project ideas from simulated versions into real prototypes within a duration of four additional weeks to achieve two objectives. The first objective is to boost the students' ability in dealing with difficulties of real hardware components/sensors/actuators. The second objective is to give the students an opportunity to increase their performance score on the VFST course over the entire semester.

## 6.2 | Interviews

Interviews were conducted for the first level and the second level of PBL methodology. A first level interview is imperative. Each student should build simulated solutions of the technical problems by using the tool chain of VFST. He/She will give an oral presentation interview to his/her teacher. In case of failure to find solutions, teachers will pay more attention and provide further discussions to such nonactive students. The second level interview is optional for any group of students. Initially, students were asked to construct a team from 3 to 4 students per group. Second, each group can freely choose a project idea which can be simple or complex from pretailored problems. Third, they can work together to convert their chosen idea to a simulated version. At the end of this previous milestone, they should present such a simulated version to their teacher in an interview to test and make sure of project functionality. Finally, they are asked to convert simulated versions to real projects by using real hardware components. Also, they will be interviewed with their teacher again to discuss and present what they achieved in the final real project. During four academic years of teaching VFST, the students prove their abilities to produce amazing solutions/prototypes of such pretailored simple/complex projects.

**Menofia University**  
**Faculty of Electronic Engineering**  
**Computer Science & Engineering (CSE)**

**MODULE SCHED**  
**MODULE: CSE 315**  
**PC interfacing (Elec)**

Week	Topic	5-minutes & long-Term Lab Quizzes	Large-Scale assignment
1	Lec 1 – Review on main concepts of PIC microcontroller & OOP concepts using C#	5-minutes Quiz	
2	Lec 2 – Interfacing PC with a PIC-based device using serial communication - I (Theoretical)	5-minutes Quiz	
3	Lec 3 – Interfacing PC with a PIC-based device using serial communication - II (practical )	5-minutes Quiz + Long-Term Quiz	Assignment 1
4	Lec 4 - Sending sensory data from PIC to PC using serial communication - I (Theoretical)	5-minutes Quiz	
5	Lec 5 - Sending sensory data from PIC to PC using serial communication - II (practical )	5-minutes Quiz + Long-Term Quiz	Assignment 2
6	Lec 6 – Simple Paradigm for USB communication - I (Theoretical )	5-minutes Quiz	
7	<b>MIDTERM EXAMS</b>		
8	Lec 7- Simple Paradigm for USB communication - II (Practical )	5-minutes Quiz + Long-Term Quiz	Assignment 3
9	Lec 8 – complete experiment for bi-directional PC/USB interface ( Theoretical & practical)	5-minutes Quiz + Long-Term Quiz	Assignment 4
10	Lec 9 – interfacing SD/MMC memory cards with PIC - I (Theoretical)	5-minutes Quiz	
11	Lec 10 – interfacing SD/MMC memory cards with PIC - II (Practical)	5-minutes Quiz + Long-Term Quiz	Assignment 5
12	Lec 11 – interfacing PC with wireless GSM/GPRS Modem - I (Theoretical)	5-minutes Quiz	
13	Lec12 - interfacing PC with wireless GSM/GPRS Modem - II (Practical)	5-minutes Quiz + Long-Term Quiz	Assignment 6
14	<b>Discussion and Presentation of real project</b>		
15	<b>FINAL EXAMS</b>		

Course Instructors	Text Book	Course Grades
<b>Name:</b>	<b>Title:</b> USB Complete: The Developers Guide	<b>Project and presentation:</b> 10%
<b>Title:</b>	<b>Author:</b> Jan Axelson	<b>Quizzes:</b> 10%
<b>Tel (work):</b>	<b>Publisher:</b> Lackview research LLC	<b>Assignments:</b> 10%
<b>Address (work):</b>	<b>Edition &amp; Year:</b> 4th edition - 2009	<b>Attendance &amp; Participation</b> 10%
<b>email:</b>	<b>Reference Book</b>	
<b>Assistant Teacher:</b>	<b>Title:</b> Serial Port Complete	<b>Lab Exp</b> 0%
	<b>Author:</b> Jan Axelson	<b>Mid-term Exam:</b> 20%
	<b>Publisher:</b> Lackview research LLC	<b>Final Exam:</b> 40%
	<b>Edition &amp; Year:</b> 2nd edition - 2007	<b>Total:</b> 100%

FIGURE 7 The outlines of the VFST course. VFST, virtual and fully software.

### 6.3 | Lecture/Lab quizzes and assignments

Three types of quizzes and assignments were given to enrolled students of VFST; 5-min quizzes, Long-Term lab quizzes, and homework assignments. 5-min quizzes were given at the last 5 min in each lecture and the student should deliver it at the same lecture. The main aim of it is to motivate the student to pay attention and concentrate over all the time duration of lecture. In addition, this 5-min quiz helped teachers to consider individual abilities of assimilation among students in each next lecture. Six Long-Term Lab quizzes were given to the students over the entire VFST course. The time slot for each quiz may be from 20 to 30 min according to the included questions. Long-Term Lab quizzes are targeted to help students make simple and quick modifications for lab experiments in lab situ. Another aim is to inspect the speed level of creating solutions for pretailored simple problems as fast as the students can. The third type was giving students large-scale assignments as homework. Such large-scale assignments enable them to perform problem solving activities without time constraints

to produce semi-optimal solutions for complex problems. Generally, quizzes and assignments in our VFST course will be used to evaluate students' ability toward consolidation of practical experiments with theoretical concepts.

## 7 | RESEARCH METHODOLOGY AND EVALUATION OF VFST

This study and the PC interfacing course were applied to 3rd year students in the CSE department, Faculty of Electronics Engineering, Menoufia University, Egypt for 4 academic years (2016/2017 to 2019/2020) during the 1st semester. It requires participating students to pass two prerequisite courses (i.e., “Microcontroller applications” and “Object Oriented programming using C#”) in the 2nd year before it can be taken. The prerequisites courses are crucial for delivering knowledge and hands-on activities on microcontroller-based hardware and firmware solutions, and implementing simple desktop applications by using Java or C# programming language. Undergraduate students, in the CSE department—Menoufia university,

**TABLE 2** Lecture organization and competencies of PC interfacing course

Block name	Virtual experiment	Weekly lecture number	Competencies (i.e., learning outcomes)
Serial Communication RS232	VE1 and VE2	1, 2, 3, 4	A2, A3, A6, A7, A10
USB Communication	VE3 and VE4	5, 6, 7, 8	B2, B3, B4
Interfacing with SD memory card with PIC	VE5	9, 10	C1, C3, C4, C6
Interfacing PC with GSM/GPRS modems	VE6	11, 12	

**TABLE 3** Number of students that enrolled in VFST in each academic year

Academic year	Total number of enrolled students	Responses number	Response rate
2016/2017	25	20	80%
2017/2018	40	33	82.5%
2018/2019	32	22	68.8%
2019/2020	61	53	86%

Abbreviation: VFST, virtual and fully software.

need to study for a total 5 academic years. The main teaching methodology was based on Lectures/Tutorials/Laboratory sessions. The structure of the PC interfacing course proposed 12 Lectures organized in 4 different blocks as shown in Table 2. In addition, Table 2 summarizes the competencies (i.e., Learning Outcomes) of the PC interfacing course which clarifies that it followed most of the general competencies of the relevant CSE program. All competencies of the CSE program (i.e., department) are listed in Appendix A.

A total of 158 undergraduate students took this course and VFST for 4 academic years, as indicated in Table 3. For this study, 128 students responded to an online questionnaire by using Google Forms. The questionnaire was considered a qualitative evaluation methodology in the current work. It included MCQ, YES/NO, and a five-point Likert scale as rating questions. We shared Q1–Q7 survey questions in April 2020 to students of academic years 2016/2017 and 2017/2018 to measure their satisfaction ranges and the beneficial effect of VFST on their career after

graduation. We shared Q1–Q4 survey questions in February 2020 to students of academic years 2018/2019 (who are still in the CSE department and enrolled in the 4th level) and 2019/2020 (who are still in the CSE department and enrolled in the 3rd level). Table 4 presents the survey questions, type of each question, targeted academic year, and educational goal factor for each question.

The exam results for 4 academic years were used as a quantitative evaluation methodology for teaching the technical contents and experiments of VFST. The questions in the VFST exams were determined by considering Bloom's taxonomy [27]. The exam results of the students were controlled and improved in accordance with providing solutions of maturely designed problems to the teacher. In the first 2 academic years (2016/2017 and 2017/2018), providing such solutions is optional, but it has become imperative in the last 2 academic years (2018/2019 and 2019/2020). In 2018/2019, students were asked to deliver these solutions individually. In 2019/2020, students were asked to divide themselves into groups with four to five members. Each group was asked to work among themselves to deliver such solutions.

## 7.1 | Qualitative evaluation and discussion

The results of survey questions Q1–Q4 are presented in Tables 5–8. These results indicate that the student cohort is generally supportive of using virtual laboratory activities as part of their work. As stated previously, survey questions Q1–Q4 have multiple objectives. The first objective is to measure the general satisfaction of students regarding the content and assessment tools of the coursework (i.e., SGS and SIA factors). The second objective is to determine whether this course is adequate for CSE programs from the students' viewpoint (i.e., SGS factor). The third objective is to measure how the engagement of theoretical concepts and virtual practice will benefit the students when they produce and design complex applications in their final graduation projects (i.e., SCO factor).

In accordance with the results in Tables 5 and 6, the average mean values of questions Q2 and Q3 are 3.70 and 3.53, respectively, indicating that 75% of the students positively support the first objective of this course (i.e., SGS and SAI). In addition, the findings in Table 7 clearly indicate that more than 85% of the students find this course strongly adequate for the CSE program (i.e., second objective and SGS). Furthermore, the feedback of the student population presented an approximately even division (refer to Table 8 for Q4) regarding the third objective, reflecting a largely neutral stance. Approximately 50% of the students believed that this course helped them produce viable final projects

TABLE 4 Questionnaire survey of VLES–Lab

Item	Type of question	Applied for academic year	Educational Goal factor
Q1: From your view-point, are contents of course is suitable for the computer science and engineering (CSE) discipline	Yes/No	ALL academic Years	SGS
Q2: Are you satisfied from experiments and interfacing ideas that you learned? Give a degree of satisfaction?	Five-point Likert scale from 1 (not satisfied) to 5 (most satisfied)	ALL academic Years	SGS
Q3: Give a degree that represents how testing techniques and evaluation tools are efficient within this course during your academic Year	Five-point Likert scale from 1 (not satisfied) to 5 (most satisfied)	ALL academic Years	SAI
Q4: Are you found that this course benefits you in your final graduation in 4th level and helped you to implement a good project?	Yes/No	ALL academic Years	SCO
Q5: Determine the benefit percentage of the course after your graduation in your career.	Write percentage response (0% to 100%)	2016/2017 2017/2018	SLS
Q6: Which part of this course that you prefer and interested in?	(MCQ question) • Serial Comm. • USB Comm. • SD card Interface • GSM Modem • Write another response	2016/2017 2017/2018	SLS
Q7: What is the most tool that you benefit from it during this course and after graduation?	(MCQ question) • MikroC compiler • C# part • Proteus simulator • Write another response	2016/2017 2017/2018	SLS

TABLE 5 Number and percentage of students that respond to Q2

Academic year	Score count of Q2				
	1	2	3	4	5
2016/2017 and 2017/2018	1 (1.9%)	5 (9.4%)	14 (26.4%)	15 (28.3%)	18 (34%)
2018/2019	2 (9.1%)	2 (9.1%)	6 (27.3%)	7 (31.8%)	5 (22.7%)
2019/2020	1 (1.9%)	4 (7.5%)	10 (18.9%)	27 (50.9%)	11 (20.8%)

TABLE 6 Number and percentage of students that respond to Q3

Academic year	Score count of Q3				
	1	2	3	4	5
2016/2017 And 2017/2018	2 (3.8%)	6 (11.3%)	11 (20.8%)	19 (35.8%)	15 (28.3%)
2018/2019	3 (13.6%)	4 (18.2%)	0 (0%)	11 (50%)	4 (18.2%)
2019/2020	6 (11.3%)	8 (15.1%)	20 (37.7%)	13(24.5%)	6 (11.3%)

**TABLE 7** Number and percentage of students that respond to Q1

Academic year	Q1 Responses		
	Yes	No	Others
2016/2017 and 2017/2018	47 (88.7%)	1 (1.9%)	5 (9.4%)
2018/2019	19 (86.4%)	2 (9.1%)	1 (4.5%)
2019/2020	48 (90.6%)	2 (3.8%)	3 (5.6%)

**TABLE 8** Number and percentage of students that respond to Q4

Academic year	Q4 Responses		
	Yes	No	Others
2016/2017 and 2017/2018	23 (43.3%)	23 (43.3%)	7 (13.2%)
2018/2019	10 (45.5%)	9 (40.9%)	3 (13.6%)
2019/2020	31 (58.5%)	11 (20.8%)	11 (20.8%)

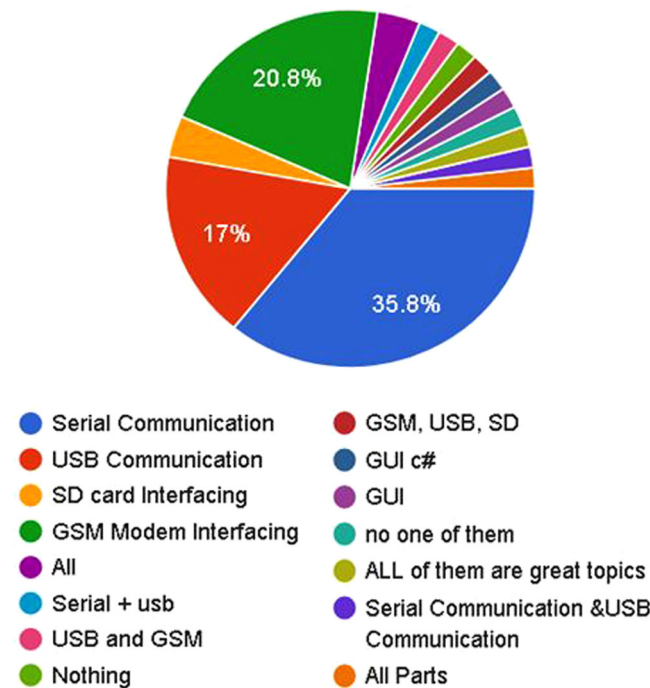
**TABLE 9** Responses to Q5

Benefit percentage	Number of students in both 2016/2017 and 2017/2018
0%–25%	13
25%–50%	5
50%–75%	17
75%–100%	18

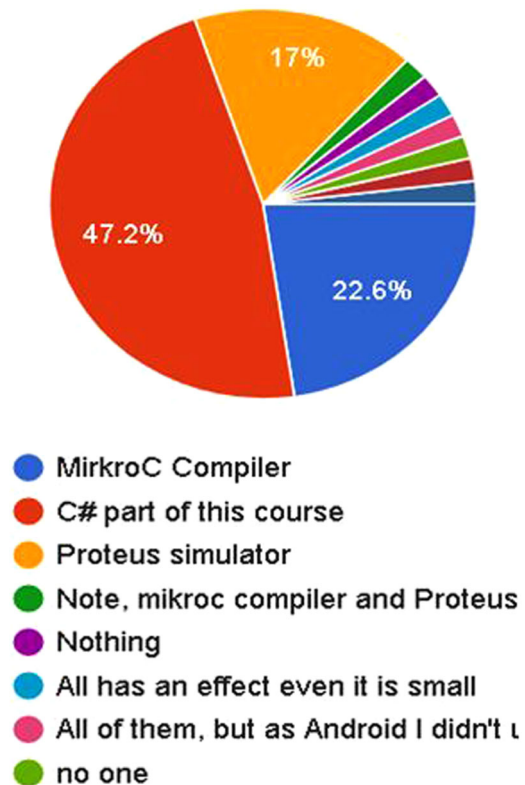
(i.e., SCO), whereas the remaining 50% of the students did not. Students work with an extensive variety of ideas for their final graduation projects. Each student attempts to apply and consolidate all the theoretical and practical knowledge he/she gained for more than 5 academic years to produce a viable project. When 50% of the students expressed that this course helped them effectively during their final project activities, such finding is considered a salient highlight with regard to the importance of this course.

We deliberately presented survey question Q5–Q7 to graduate students to measure the direct effect of this course on their professional life after graduation (i.e., SLS). Accordingly, Q5–Q7 were presented to students of academic years 2016/2017 and 2017/2018. The total number of students that responded to Q5–Q7 was 53 from both academic years. The answers to Q5 are provided in Table 9. From this table, 17 students benefitted from this course by a percentage of 50% to 75%, while 18 students reported a beneficial percentage of 75%–100%. That is, 66% of the students benefitted directly from this coursework in their career. The responses to Q6 and Q7 are presented as pie charts in Figures 8 and 9. The objective of Q6 is to determine which interfacing technologies students are most interested in and are most useful to them (i.e., SLS).

The student population was interested in serial communication technology (35.8%), USB communication (17%), and



**FIGURE 8** Responses of Q6



**FIGURE 9** Responses of Q7

TABLE 10 Statistical analysis of exam results of VFST

Group	Years	Mean mark	SD	Min Mark	Max Mark
Optional delivering solutions	2016/2017	49.61	28.65	20	87
	2017/2018	53.82	27.36	28	90
Imperative delivering solutions individually	2018/2019	61.53	22.13	35	98
Imperative delivering solutions In Groups	2019/2020	74.25	18.21	54	100

Abbreviation: VFST, virtual and fully software.

GSM modem interfacing (20.8%). In addition, the objective of Q7 is to identify which tool benefitted the students from the most during the course and on their career after graduation (i.e., SLS). From Figure 8, the C# part of this course elicited the most interest, that is, 47.2% of the participating students. Meanwhile, the MikroC compiler and Proteus simulator elicited the interest of 22.6% and 17% of the participating students, respectively.

## 7.2 | Quantitative evaluation and discussion

Table 10 presents the quantitative evaluation (i.e., statistical exam results) of teaching VFST. It provides the mean grades and standard deviations over 4 academic years of enrolled students and students who finished the course. Notably, a substantial improvement in mean grades is observed when we make providing challenging solutions imperative. The reason behind such improvement is that students in 2018/2019 engaged in more thinking and mental activities to find suitable solutions for the predesigned problems of VFST. Moreover, grouping students encourages them to exchange and discuss their ideas within their dedicated group to come up with remarkable solutions as an achievement of the SCO factor. A significant number of students in the last 2 academic years are capable of correctly answering exam questions compared with students in academic years 2016/2017 and 2017/2018.

## 8 | CONCLUSION AND FUTURE WORK

Supporting the concept of converting traditional laboratories into virtual laboratory toolchains has been a recent trend in many universities and colleges. Applying such concepts generally to CSE courses, and particularly to PC interfacing courses, poses a challenge. Teaching such courses requires consolidation between theoretical knowledge and laboratory experiments to provide effective skill acquisition and hands-on experience to students. To solve this discrepancy, PC interfacing

teachers are motivated to find methodologies and full software platforms that can help students perform experiments virtually with an aesthetic appearance and manner similar to real experiments.

This study addressed in detail the planning and design of a syllabus and VE for the topic of PC interfacing among virtual device (s) by using Proteus with desktop applications developed using a general-purpose and textual C# programming language. This syllabus is called VFST. In this syllabus, students are responsible for developing a firmware for any participating microcontroller and developing a complete GUI and textual source code for each desktop application in this context. A group of third-party programs can be used as a link between the Proteus simulator and C#-based applications to provide virtual communication channels, such as VSPD, virtual USB driver, WinImage program, RealTerm program, and MikroC compiler. These third-party programs with the Proteus simulator and C# programming language comprise a toolchain dedicated to the VFST syllabus. The primary objective of VFST is to enable students and researchers to develop desktop applications that can monitor sensory data from different sensor types or send operational commands to different types of actuators and implement solutions with an aesthetic appearance similar to LabVIEW capabilities.

This study also presented qualitative and quantitative evaluations of the benefits and effectiveness of VFST on students before and after their graduation. The findings of such evaluations were collected via feedback responses from students from 4 academic years (2016/2017 to 2019/2020) using an online questionnaire survey (Google Forms). These findings indicate that 75% of the student cohort is extremely satisfied with the content and VE of VFST. In addition, 66% of students experience direct benefit from this coursework on their career after graduation. In the future, we aim to include theoretical concepts and VE related to other wired interface technologies (e.g., I2C, LIN, Ethernet, and CAN protocols) and other wireless technologies (e.g., Wi-Fi, Bluetooth, and Zigbee protocols).

## DATA AVAILABILITY STATEMENT

Data available on request from the authors.

## ORCID

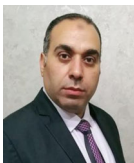
Mokhtar A. A. Mohamed  <http://orcid.org/0000-0002-1522-4571>

## REFERENCES

1. T. Achalakul, B. Sirinaovakul, and N. Nuttaworakul, *Virtual laboratory: a distributed collaborative environment*, *Comput. Appl. Eng. Educ.* **12** (2004), 44–53. <https://doi.org/10.1002/cae.20008>
2. M. M. Alamri, *Students' academic achievement performance and satisfaction in a flipped classroom in Saudi Arabia*, *Int. J. Technol. Enhanced Learn.* **11** (2019), no. 1, 103–119.
3. W. M. Alenazy, W. Mugahed al-Rahmi, and M. S. Khan, *Validation of TAM model on social media use for collaborative learning to enhance collaborative authoring*, *IEEE Access* **7** (2019), 71550–71562. <https://doi.org/10.1109/ACCESS.2019.2920242>
4. M. A. Almulla, *The efficacy of employing problem-based learning (PBL) approach as a method of facilitating students' achievement*, *IEEE Access* **7** (2019), 146480–146494. <https://doi.org/10.1109/ACCESS.2019.2945811>
5. M. A. Almulla, *The effectiveness of the project-based learning (PBL) approach as a way to engage students in learning*, *Sage Open* **10** (2020), no. 3, 215824402093870. <https://doi.org/10.1177/215824402093870>
6. B. Alorda, K. Suenaga, and P. Pons, *Design and evaluation of a microprocessor course combining three cooperative methods: SDLA, PjBL and CnBL*, *Comput. Educ.* **57** (2011), 1876–1884.
7. W. M. Al-rahmi, M. S. Othman, and L. Mi Yusuf, *The effect of social media on researchers' academic performance through collaborative learning in Malaysian higher education*, *Mediterranean J. Soc. Sci.* **6** (2015), no. 4, 193–203. <https://doi.org/10.5901/mjss.2015.v6n4s1p193>
8. W. M. Al-Rahmi, N. Yahaya, U. Alturki, A. Alrobai, A. A. Aldraiweesh, A. Omar Alsayed, and Y. B. Kamin, *Social media-based collaborative learning: the effect on learning success with the moderating role of cyberstalking and cyberbullying*, *Interact. Learn. Environ.* **30** (2020), no. 3, 1434–1447. <https://doi.org/10.1080/10494820.2020.1728342>
9. A. A. Altalbe, *Performance impact of simulation-based virtual laboratory on engineering students: a case study of Australia virtual system*, *IEEE Access* **7** (2019), 177387–177396. <https://doi.org/10.1109/ACCESS.2019.2957726>
10. J. Axelson, *Serial Port Complete*, 2nd edition, Lackview Research LLC, 2007.
11. J. Axelson, *USB Complete: The Developers Guide*, 5th edition, Lackview research LLC, 2017.
12. B. Balamuralithara and P. C. Woods, *Virtual laboratories in engineering education: the simulation lab and remote lab*, *Comput. Appl. Eng. Educ.* **17** (2009), 108–118. <https://doi.org/10.1002/cae.20186>
13. B. J. S. Barron, D. L. Schwartz, N. J. Vye, A. Moore, A. Petrosino, L. Zech, and J. D. Bransford, *Doing with understanding: lessons from research on problem- and project-based learning*, *J. Learn. Sci.* **7** (1998), 271–311. <https://doi.org/10.1080/10508406.1998.9672056>
14. K. Bartscher, B. Gould, and S. Nutter, *Increasing student motivation through project-based learning* Master's Research Project, SaintXavier and IRI Skylight, 1995.
15. A. Buldu and H. Korkmaz, *A USB kit for digital I/O applications in a digital electronics lab designed by using PIC16C765 microcontroller*, *Comput. Appl. Eng. Educ.* **17** (2009), 131–138. <https://doi.org/10.1002/cae.20172>
16. N. E. Cagiltay, E. Aydin, C. C. Aydin, A. Kara, and M. Alexandru, *Seven principles of instructional content design for a remote laboratory: a case study on ERRL*, *IEEE Trans. Educ.* **54** (2011), 320–327. <https://doi.org/10.1109/TE.2010.2058115>
17. N. E. Cagiltay, E. Aydin, R. Oktem, A. Kara, M. Alexandru, and B. Reiner, *Requirements for remote RF laboratory applications: an educators' perspective*, *IEEE Trans. Educ.* **52** (2009), 75–81. <https://doi.org/10.1109/TE.2008.919806>
18. S. Chtourou, M. Kharrat, N. Ben Amor, M. Jallouli, and M. Abid, *Using IOIOAI in introductory courses to embedded systems for engineering students: a case study*, *Int. J. Electr. Eng. Educ.* **55** (2018), no. 1, 62–78. <https://doi.org/10.1177/0020720917750959>
19. E. T. H. Chu and C. W. Fang, *CALEE: a computer-assisted learning system for embedded OS laboratory exercises*, *Comput. Educ.* **84** (2015), 36–48. <https://doi.org/10.1016/j.compedu.2015.01.006>
20. R. B. Coruk, B. Yalcinkaya, and A. Kara, *On the design and effectiveness of simulink-based educational material for a communication systems course*, *Comput. Appl. Eng. Educ.* **28** (2020), 1641–1651. <https://doi.org/10.1002/cae.22344>
21. R. Dak, S. Khoje, and P. J. Pardeshi, *A review on cost-efficient virtual laboratory: a teaching aid*, *International Conference on Computing Methodologies and Communication (ICCMC)*, 2017.
22. L. D. Feisel and A. J. Rosa, *The role of the laboratory in undergraduate engineering education*, *J. Eng. Educ.* **94** (2005), no. 1, 121–130. <https://doi.org/10.1002/j.2168-9830.2005.tb00833.x>
23. GNU General Public License, <http://www.gnu.org/licenses/>
24. A. M. González-Vargas, J. M. Serna-Ramirez, C. Fory-Aguirre, A. Ojeda-Misses, J. M. Cardona-Ordoñez, J. Tombé-Andrade, and A. Soria-López, *A low-cost, free-software platform with hard real-time performance for control engineering education*, *Comput. Appl. Eng. Educ.* **27** (2019), 406–418. <https://doi.org/10.1002/cae.22084>
25. Simon Inns, [http://www.waitingforfriday.com/index.php/Open\\_Source\\_Framework\\_for\\_USB\\_Generic\\_HID\\_devices\\_based\\_on\\_the\\_PIC18F\\_and\\_Windows](http://www.waitingforfriday.com/index.php/Open_Source_Framework_for_USB_Generic_HID_devices_based_on_the_PIC18F_and_Windows)
26. S. Kacar and C. Bayilmis, *A web-based educational interface for an analog communication course based on MATLAB builder NE with web figures*, *IEEE Trans. Educ.* **56** (2013), 346–354. <https://doi.org/10.1109/TE.2012.2236329>
27. D. R. Krathwohl, *A revision of Bloom's taxonomy: an overview*, *Theory Pract.* **41** (2002), no. 4, 212–218. [https://doi.org/10.1207/s15430421tip4104\\_2\\_C](https://doi.org/10.1207/s15430421tip4104_2_C)
28. K. Küçük, *RTWiFi-Lab: a real-time Wi-Fi laboratory platform on USRP and LabVIEW for wireless communications education and research*, *Comput. Appl. Eng. Educ.* **26** (2018), 111–124. <https://doi.org/10.1002/cae.21865>
29. H. Lim, H. Yu, and T. Suh, *Using virtual platform in embedded system education*, *Comput. Appl. Eng. Educ.* **20** (2012), 346–355. <https://doi.org/10.1002/cae.20401>
30. E. D. Lindsay and M. C. Good, *Effects of laboratory access modes upon learning outcomes*, *IEEE Trans. Educ.* **48** (2005), 619–631. <https://doi.org/10.1109/TE.2005.852591>
31. S. Mayoof, H. Alaswad, S. Aljeshi, A. Tarafa, and W. Elmedany, *A hybrid circuits-cloud: development of a low-cost secure*

- cloud-based collaborative platform for A/D circuits in virtual hardware E-Lab, *Ain Shams Eng. J.* **12** (2021), no. 2, 1197–1209. <https://doi.org/10.1016/j.asej.2020.09.012>
32. Microchip Technology Inc. <http://ww1.microchip.com/downloads/en/DevicesDoc/39564c.pdf>
  33. Y. Mohanna, M. Hamad, R. Jabr, A. Alaeddine, and O. Bazzi, *Teaching microprocessors, microcontrollers, and digital signal processing courses using only one target processor: the newborn dsPIC30F™ from Microchip™*, *Comput. Appl. Eng. Educ.* **15** (2007), 185–191. <https://doi.org/10.1002/cae.20109>
  34. C. Perrenet, P. A. Bouhuijs, and J. G. Smits, *The suitability of problem-based learning for engineering education: theory and practice*, *Teach. High. Educ.* **5** (2000), no. 3, 345–358. <https://doi.org/10.1080/713699144>
  35. V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, *Virtual laboratories for education in science, technology, and engineering: a review*, *Comput. Educ.* **95** (2016), 309–327. <https://doi.org/10.1016/j.compedu.2016.02.002>
  36. E. A. Qaralleh and K. A. Darabkh, *A new method for teaching microprocessors course using emulation*, *Comput. Appl. Eng. Educ.* **23** (2014), 455–463. <https://doi.org/10.1002/cae.21616>
  37. S. Sahin and Y. Isler, *Microcontroller-based robotics and SCADA experiments*, *IEEE Trans. Educ.* **56** (2013), no. 4, 424–429. <https://doi.org/10.1109/TE.2013.2248062>
  38. Silicon laboratories, <https://www.silabs.com/interface/usb-bridges/classic>.
  39. SIMCOM wireless Solutions Limited, <https://www.simcom.com/products/sim800.html>.
  40. B. Su and L. Wang, *Application of proteus system modelling (VSM) in teaching of microcontroller*, International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), 2010.
  41. Texas Instruments, <https://www.ti.com/lit/ds/symlink/max232.pdf>
  42. P. Tiernan, *Enhancing the learning experience of undergraduate technology students with LabVIEW software*, *Comput. Educ.* **55** (2010), 1579–1588. <https://doi.org/10.1016/j.compedu.2010.07.001>
  43. I. Uyanik and B. Catalbas, *A low-cost feedback control systems laboratory setup via Arduino-Simulink interface*, *Comput. Appl. Eng. Educ.* **26** (2018), 718–726. <https://doi.org/10.1002/cae.21917>
  44. A. Van Barneveld, and J. Strobel, *Problem-based learning: Effectiveness, drivers, and implementation challenges*, *Research on PBL practice in engineering education* (X. Du, E. De Graaff, and A. Kolmos, eds.), Sense Publishers, Rotterdam, Netherlands, 2009, pp. 35–44. [https://doi.org/10.1163/9789087909321\\_005](https://doi.org/10.1163/9789087909321_005)

## AUTHOR BIOGRAPHY



**Moukhtar A. A. Mohamed** received his BSc, MSc, and PhD degrees from the Computer Science & Engineering (CSE) Department, Faculty of Electronic Engineering, Menoufia University, Egypt, in 2002, 2008, and 2016 respectively.

Since 2016–2022, he has been an Assistant Professor with the CSE Department, Menoufia University, Faculty of Electronic Engineering. Since 2022, he has joined the CSE department in MSA University (one of the leading universities for British standards in higher education in Egypt). He is the author of more than 15 research articles. He is serving as a reviewer in various Journals and conferences. His research interests include Embedded systems using microcontrollers, Internet Of Things (IOT) applications, computer peripherals and interfacing, biometrics and security.

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** M. A. A. Mohamed, *VFST: Virtual and fully software based toolchain for PC interfacing education and research*, *Comput. Appl. Eng. Educ.* (2022), 1–19. <https://doi.org/10.1002/cae.22590>

## APPENDIX A: COMPETENCIES OF CSE DEPARTMENT

In this section, comprehensive details will be presented about program competencies (Learning Outcomes) of CSE dept. These competencies are skimmed from Egyptian Standards NARS-2009 and NARS-2018 (<http://cmp.eng.cu.edu/wp-content/uploads/sites/8/cmp1/2014/EngineeringNARS18.pdf>) for engineering education which is regulated by the National Authority for Quality Assurance and Accreditation of Education “NAQAAE”—Egyptian Ministry of Higher Education (<https://naqaae.eg/en/>). The categories of competencies in our CSE Department are as follows: General Competencies (which are related to the whole Engineering Education), Specialty Competencies (which are related to the discipline of Electrical Engineering Education), Subspecialty Competencies (which are related to the subdiscipline of Computer Science and Engineering Departments). These Competencies are formulated to be general sentences as described in the following sections.

- **General Competencies of CSE Department:** it is symbolized with Letter “A”

**A1.** Identify, formulate, and solve complex engineering problems by applying engineering fundamentals, basic science and mathematics.

**A2.** Develop and conduct appropriate experimentation and/or simulation, analyze and interpret data, assess and evaluate findings, and use statistical analyses and objective engineering judgments to draw conclusions.

**A3.** Apply engineering design processes to produce cost-effective solutions that meet specified needs with consideration for global, cultural, social, economic, environmental, ethical and other aspects as appropriate to the discipline and within the principles and contexts of sustainable design and development.

**A4.** Utilize contemporary technologies, codes of practice and standards, quality guidelines, health and safety requirements, environmental issues and risk management principles.

**A5.** Practice research techniques and methods of investigation as an inherent part of learning.

**A6.** Plan, supervise and monitor implementation of engineering projects, taking into consideration other trades requirements.

**A7.** Function efficiently as an individual and as a member of multi-disciplinary and multicultural teams.

**A8.** Communicate effectively – graphically, verbally and in writing – with a range of audiences using contemporary tools.

**A9.** Use creative, innovative and flexible thinking and acquire entrepreneurial and leadership skills to anticipate and respond to new situations.

**A10.** Acquire and apply new knowledge; and practice self, lifelong and other learning strategies.

- **Specialty Competencies of CSE Department:** it is symbolized with Letter “B”).

**B1.** Select, model and analyze electrical power systems applicable to the specific discipline by applying the concepts of: generation, transmission and distribution of electrical power systems.

**B2.** Design, model and analyze an electrical/electronic/digital system or component for a specific application; and identify the tools required to optimize this design.

**B3.** Design and implement: elements, modules, sub-systems or systems in electrical/electronic/digital engineering using technological and professional tools.

**B4.** Estimate and measure the performance of an electrical/electronic/digital system and circuit under specific input excitation, and evaluate its suitability for a specific application.

**B5.** Adopt suitable national and international standards and codes to: design, build, operate, inspect and maintain electrical/electronic/digital equipment, systems and services.

- **Subspecialty Competencies of CSE Department:** it is symbolized with Letter “C”).

**C1.** Acquire engineering fundamentals in the fields of computer systems hardware and software.

**C2.** Analyze computer systems using appropriate mathematical tools, professional technological tools, and computing methods based on computer science and engineering disciplines.

**C3.** Design and implement of computer systems hardware and software using technological and professional tools.

**C4.** Identify, formulate, and model computer science and engineering problems and provide multiple, practical and professional solutions with a high level of competence.

**C5.** Apply and evaluate different techniques and strategies for solving computer science and engineering problems.

**C6.** Acquire a sound theoretical approach to deal with modern trends and new advances in the field of computer hardware and software.