121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014
Paper ID #8394

LabVIEW: A Teaching Tool for the Engineering Courses

Dr. Alireza Kavianpour, DeVry University, Pomona

Dr. Alireza Kavianpour received his PH.D. Degree from University of Southern California (USC). He is currently Senior Professor at DeVry University, Pomona, CA. Dr. Kavianpour is the author and co-author of over forty technical papers all published in IEEE Journals or referred conferences. Before joining DeVry University he was a researcher at the University of California, Irvine and consultant at Qualcom Inc. His main interests are in the areas of embedded systems and computer architecture.

LabVIEW: A Teaching Tool for the Engineering Courses

Abstract

Computer programming in languages such as Visual Basic, C++, or JAVA follow a control flow model of program execution. In the control flow model, the sequential order of program elements determines the execution order of a program. A program written in LabVIEW uses a slightly different approach compared to the conventional method known as dataflow programming.

LabVIEW stands for **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench. This powerful graphical development system, developed by National Instruments (NI), is a widespread teaching tool and is used in many industries. LabVIEW can be used for data acquisition, machine control, instrument control, and a wide variety of other application needs. Programs written in LabVIEW are known as Virtual Instruments (VIs). A single VI consists of two parts: a front panel and a block diagram. The front panel consists of controls and indicators which include knobs, push buttons, graphs, and many other objects. The controls of the front panel are inputs into the VI while indicators are outputs from the VI. Away from the conventional programming model of line by line code execution, VI programs are based on data flow programming.

This paper discusses the use of LabVIEW in the several engineering courses such as data microprocessors, communication, signal processing, programming languages, and digital circuits. Most of the examples presented in this paper were implemented on the microcontroller Stellaris LM3S8962 based on ARM processor. LabVIEW based classwork explained in this paper has been implemented in several courses and has helped students further their understandings on the fundamentals of computer hardware and software.

1-Introduction

Computer programming in languages such as Visual Basic, C++, or JAVA follow a control flow model of program execution. In control flow, the sequential order of program elements determines the execution order of a program. A program written in LabVIEW uses a slightly different approach to than programs written in other languages[1,2]. This is called dataflow programming and, when a function such as addition or subtraction executes it receives all required inputs before it can be executed. A function will produce output data and passes the data to the next node in the dataflow path. The movement of data through the nodes determines the execution order.

LabVIEW is an acronym name for Laboratory Virtual Instrumentation Engineering Workbench[3,4,5]. It is a graphical developments system designed by National Instruments. It is used in many industries. It can be used for data acquisition, machine control, instrument, and a wide variety of other applications. The use of LabVIEW reduces development time for a design[6].

LabVIEW provides engineers with the tools needed to create and deploy measurement and control systems[7,8]. You can use LabVIEW to simulate the program and compare the outcomes to the expected results. Once the program simulates correctly, you can download the program to the hardware and test its functionality. LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitates physical instruments, such as oscilloscopes and multi-meters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help troubleshoot codes.

LabVIEW can be used with IO interfaces such as GPIB, PXI, VXI, RS-232, and RS-484. It also has built-in features for connecting your application to the web using a web server and software standards such as TCP/IP networking and ActiveX. You also can create stand-alone executables and shared libraries, such as DLLs.

LabVIEW is also designed to be extendible[9]. You can add modules through various means. A manufacturer of an interface card or an instrument may provide you with a driver which appears as a VI representing the card and its functionality in the LabVIEW environment. You can also write a module and save it as VIs (Virtual Instruments) to be used in other programs. You can also write modules which interface with LabVIEW in other languages such as C and C++. These are known as “sub-VIs” and are no different from VIs except that the interface has been defined to the next level. Sub-VIs in C or C++ is very useful if you have a complex numerical procedure to perform on the data which is not covered in a standard LabVIEW routine.

A single VI in LabVIEW consists of two parts: A front panel and a block diagram[3]. The front panel consists of controls and indicators. Controls include knobs, push buttons, graphs, and many other objects. Controls are inputs into the VI. Indicators include lamps, progress bars, meters, gauges, and many other objects. Indicators are outputs from the VI. The block diagram contains front panel terminals, constants, functions, structures, and wires that carry data from one object to another. Front Panel Toolbar has the options for reorder, resize, distribute, and align objects. Text settings, pause, and run are the other options. Block Diagram Toolbar has similar functions.

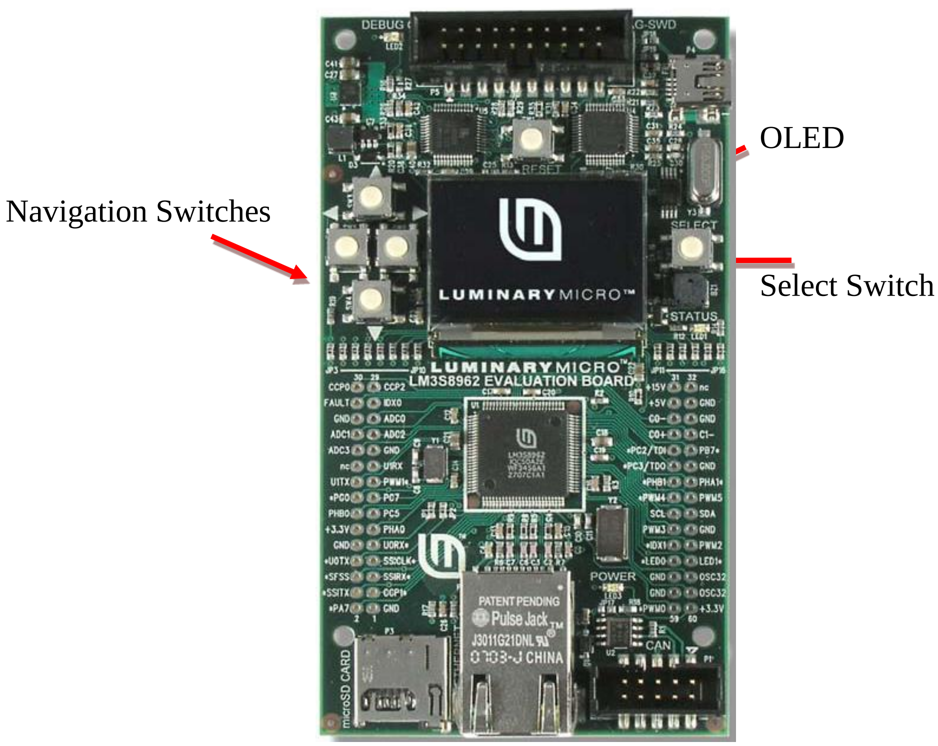
LabVIEW contains three palettes called tools, controls, and functions[5]. To create, modify, and debug a VI, tools palette are used. The controls palette is used to place objects on the front panel. The palette consists of top-level icons representing sub-palettes. The functions palette is used to place objects on the block diagram. The palette consists of top-level icons representing sub-palettes. Functions are grouped in sub-palettes based on the type of function. Boolean controls provide True/False input into a VI. Numeric controls provide numeric data input into a VI. Numeric indicators provide a numeric output from a VI. Each time a control or indicator is placed on the front panel, a corresponding terminal is placed in the block diagram. Some controls can be changed to indicators. Also, some indicators can be changed to controls.

2- Stellaris® LM3S8962 ARM Target Board

Most of the examples/homework presented in this paper were implemented on the microcontroller Stellaris LM3S8962 based on ARM processor. The Stellaris LM3S8962 Evaluation Board is a compact and versatile evaluation board for the Stellaris LM3S8962 ARM

Cortex M3 microcontroller. This board contains several I/O components and an OLED graphics display.

In this board, Navigation switches and the Select switch are used as the controls. Controls are inputs to a VI. The OLED display and the Status LED are used as the indicators or outputs from the VI. The Navigation switches, Select switch, OLED display, and the Status LED are connected to the ARM microcontroller. If the microcontroller is programmed correctly, we can control program operation using the switches. The output from the microcontrollers can be displayed on the OLED or the Status LED. In LabVIEW, fixed elements of an ARM microcontroller, such as switches and LEDs, are called Elemental I/O. Following figure is the image of Stellaris LM3S8962 ARM Cortex M3 microcontroller.



3- The use of LabVIEW in the different courses

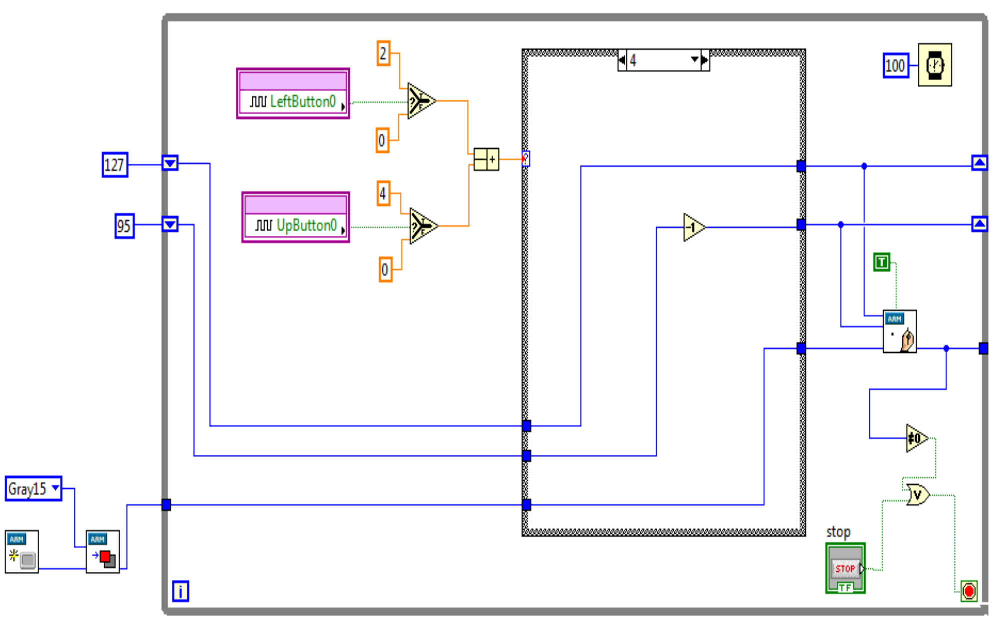
LabVIEW could be used to enhance your teaching skills. In the following sections several homework will be explained. The author used these homework in his courses.

(a)-Microprocessors Course.

LabVIEW has a module called Elemental IO. With the use of this module and a microcontroller, several concepts in the microprocessors course could be implemented. In one of the homework students asked to write a program in LabVIEW to implemented Etch-a-Sketch on Stellaris

LM3S8962 Evaluation Kit with ARM processor. The classic Etch-A-Sketch toy project, taught student how to control OLED display of the microcontroller. Following displays front and part of block diagram portion of this homework.

The front panel is essentially the display of the program. It shows which button is pressed using the Up, Down, Left, and Right LED indicators. The direction indicator shows the direction of pixel movement (i.e. right, down + left, etc.) . The X and Y axis coordinates display in real time the position of a pixel on the OLED display.



Recreating the Etch-A-Sketch in the form pixel movement is a great learning experience for students. Similar homework could be used for teaching interrupts and input/output operations.

(a)- Digital courses

LabVIEW could be used to enhance the teaching of digital courses. It is very helpful in explaining many important topics such as: Binary to decimal, decimal to binary, decoders, multiplexors, RS flip flop oscillations (Case RS=11), and many other digital topics.

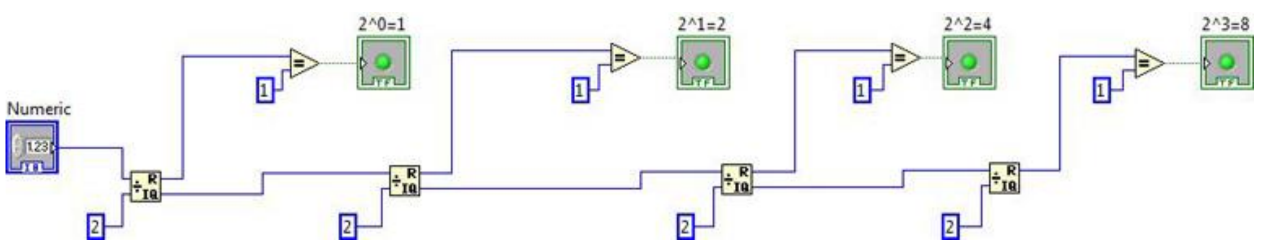
Homework : Design a decimal to binary convertor.

1-Divide the decimal number by two, this result a quotient and a reminder.

2-The remainder will be a bit in the four-bit binary number.

3-Successively divide the quotient by two, this result a new quotient and new remainder. Repeat this step until the quotient is 0.

4-The first remainder obtained is the first bit of the binary number. The second remainder obtained will be the second bit of the binary number. The third remainder obtained will be the third bit of the binary number. The fourth remainder will be the fourth bit of the binary number. Following figure is the block diagram of this homework.

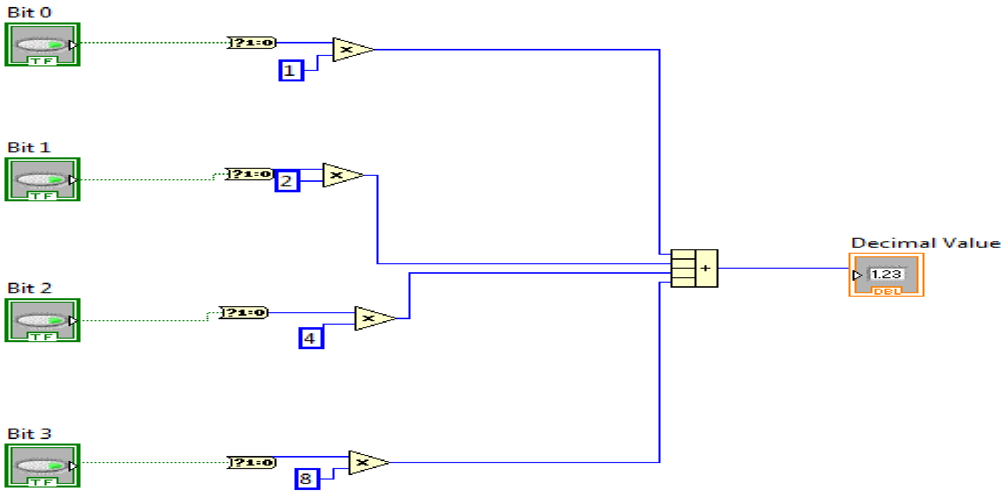


Homework : Design a binary to decimal convertor.

1-Multiply bit #i by 2^i.

2-Add all the results obtained from step 1.

Following figure is the block diagram of this homework.



These homework could be expanded for teaching the procedure for converting a number from other bases such as octal or hexadecimal.

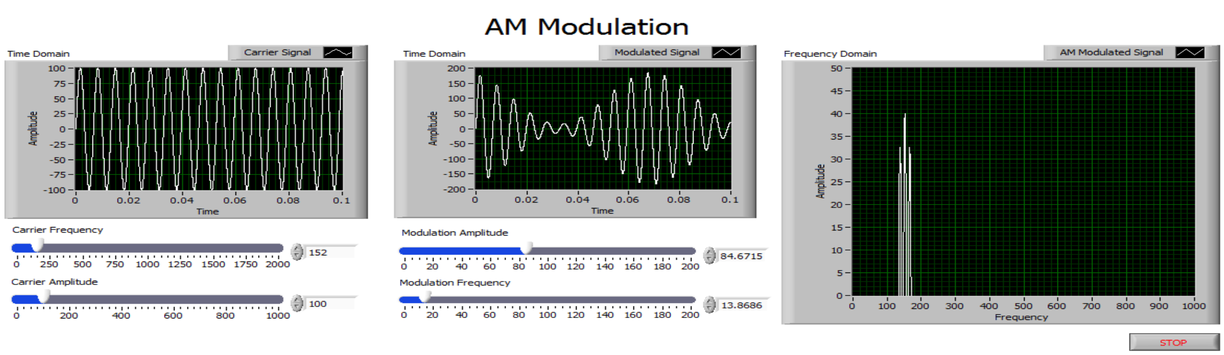
(b)- LabVIEW in the Data Communications courses

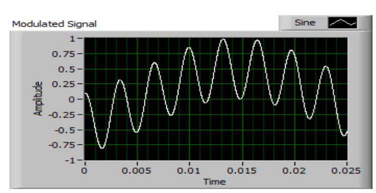
LabVIEW could be used to enhance teaching communication courses. It is very helpful in explaining many important topics such as: AM, FM, and PSK, and many other communication topics.

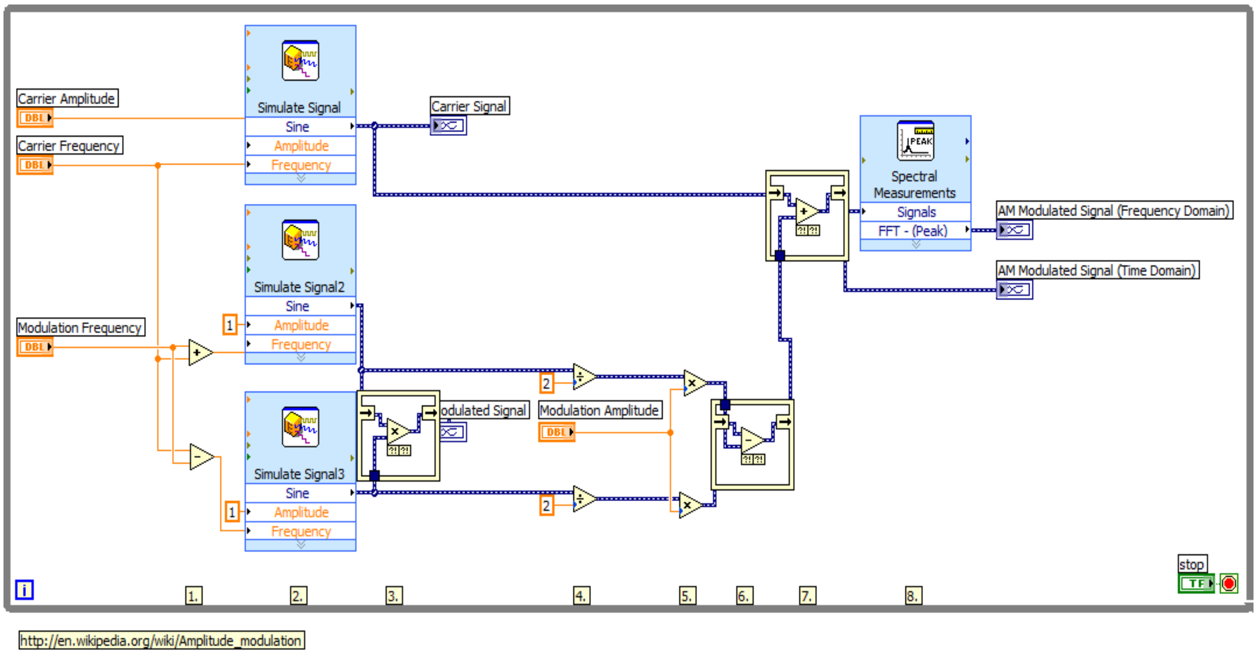
Homework : Design AM modulation with LabVIEW. Use the following formula:

$$y(t) = C\sin(\omega_c t) + M\frac{\cos(\phi - (\omega_m - \omega_c)t)}{2} - M\frac{\cos(\phi + (\omega_m + \omega_c)t)}{2}$$

Following figure is the Front and block diagram of this homework.





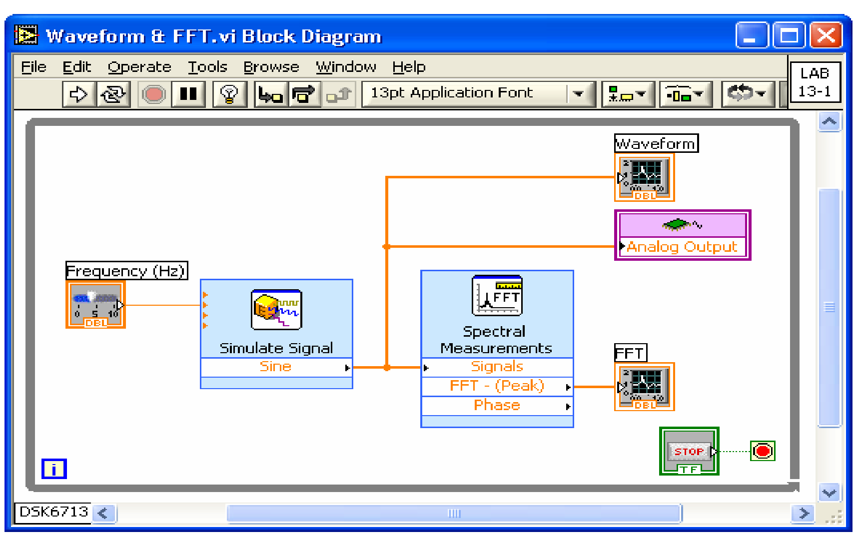


$$y(t) = C\sin(\omega_c t) + M\frac{\cos(\phi - (\omega_m - \omega_c)t)}{2} - M\frac{\cos(\phi + (\omega_m + \omega_c)t)}{2}$$

Similar homework could be used for teaching Shannon's Theorem and the channel capacity.

©- LabVIEW in Digital Signal Processing courses

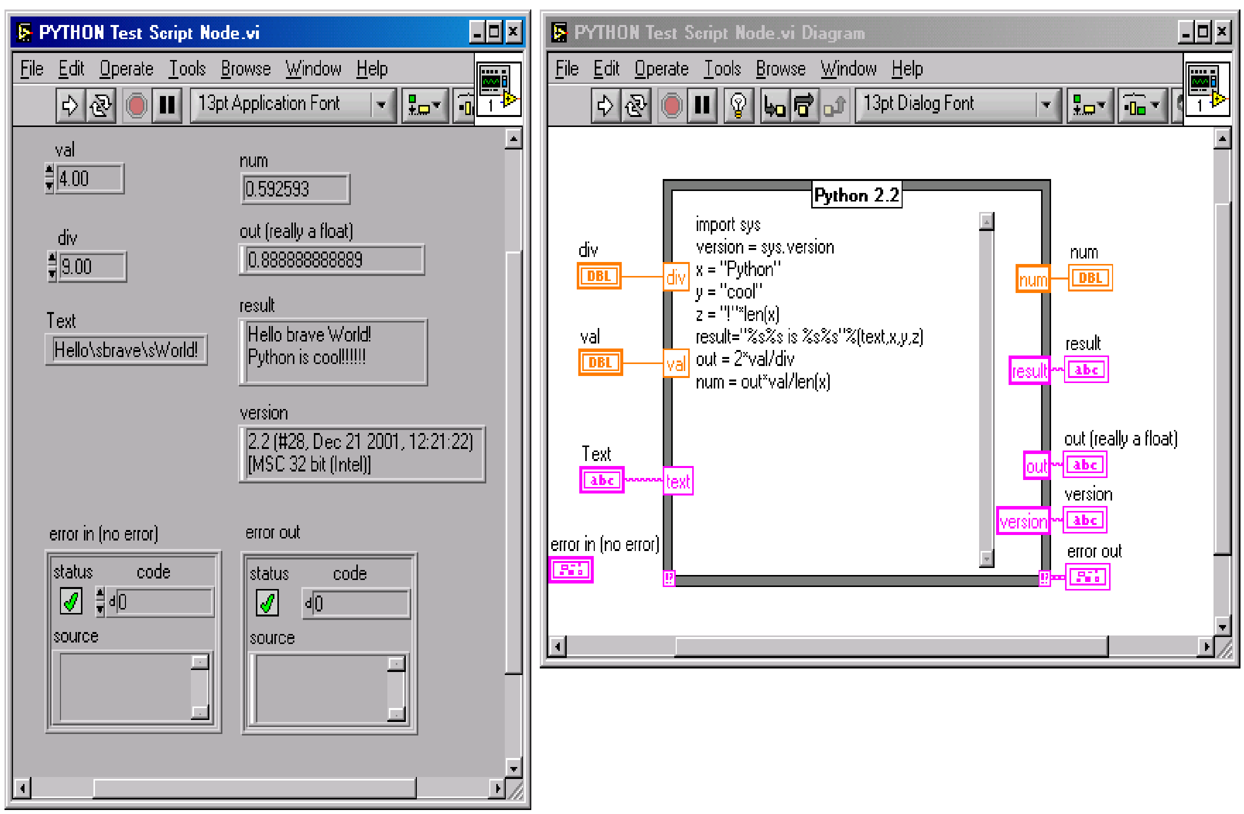
LabVIEW has a DSP Module and could be used to run DSP graphical codes directly on a DSP target board without performing any C programming. In the DSP course the author asked student to implement the following waveform generation and digital filtering on The Stellaris® LM3S8962 Evaluation Board.



Similar homework could be used for teaching low, high, and band pass filters..

(d)- Scripting Languages and LabVIEW

LabVIEW gives users the ability to directly access and integrate code written in a variety of languages. LabVIEW can be used in the hybrid approaches where a script can be invoked by a LabVIEW application. Scripting languages are commonly used for analysis and visualization. LabVIEW software offers different options for using scripting languages, such as MATLAB and Python. Following figure is the Front and block diagram of a homework in Python programming.



4- Summary

This paper discusses the use of LabVIEW in the several engineering courses. Different examples are given for the possible use by instructors in their courses. The teaching experience of the author shows the importance of LabVIEW in a classroom setting to further understand the fundamental concepts of engineering courses. Engineering students with computer science background are more receptive to using LabVIEW for coursework and projects. In comparison to the courses taught without LabVIEW, the author observes an increase in academic performance when LabVIEW is used.

The implementation of LabVIEW was conducted for an onsite four years program. Further studies are needed to better understand the impact of LabVIEW on different courses in an online setting.

5-References

1. LabVIEW Digital Signal Processing by Clark Cory, McGraw-Hill, 2005

2. LabVIEW for Everyone: Graphical Programming Made Easy and Fun (3rd Edition) by Jeffrey Travis and Jim Kring Prentice Hall 2006

3. Learning with LabVIEW by Robert Bishop, Prentice Hall, 2009

4. LabVIEW for Electrical Engineers and Technologists by Stephen Philip Tubbs, Stephen P. Tubbs 2011

5- Hands-On Introduction to LabVIEW for Scientists and Engineers by John Essick, Oxford University Press, 2012

6- Introduction to Data Acquisition With LabVIEW 2ND Edition By Robert King, McGraw Hill,2012

7- Arduino+LabVIEW by Spark Fun, 2011

8- Learn LabVIEW Fast by Doug Stamps, 2012

9-LabVIEW Graphical Programming Cookbook by Yik , 2014