

Development of Real-Time Signal Generator Graphical User Interface Using Matlab 6.5

MUHAMMAD SHARFI NAJIB, MOHD SHAWAL JADIN, MOHD RAZALI DAUD
Faculty of Electrical & Electronics Engineering,
Universiti Malaysia Pahang,
Karung Berkunci 12, 25000 Kuantan, Pahang,
MALAYSIA

Abstract: - Real-time application using MATLAB is very difficult to be implemented since the implementation requires very long and tedious programming code. This paper presents an approach to design real-time signal generator system that uses Graphical User Interface (GUI) development tools provided in MATLAB 6.5. The real-time signal generator GUI aimed to compensate the difficulties of writing programming code that enables a beginner to use the application without having to write programming code. The system uses Humosoft AD612 data acquisition card as the hardware part. The signals generated from the GUI are compared with the external oscilloscope to verify the reliability of the GUI. The example of application described in this paper is virtual oscilloscope with 3 independent channels display for real time signal generation.

Key-Words: - Real-time, GUI, Generator, Oscilloscope, Matlab

1 Introduction

At the moment, data-acquisition techniques are changing so rapidly these days that it is difficult to summarize all the possibilities. Some time ago these techniques were reserved for computer professionals who used assembly language and deep technical information to produce packages that were hard to modify and understand. However, the field is gradually becoming more user-friendly, making it more accessible to those who use data acquisition. It is possible now to program many acquisition cards in the C language, and the visual LabView language developed by National Instruments which is becoming more common.

AD612 is one of the latest versions of data acquisition card that had been produced by the Humosoft. This card had been designed for standard data acquisition and control applications and optimized for use with Real Time Toolbox for MATLAB. MATLAB has a command window that gives interactive windows to a user to operate this data acquisition card. However, the command window is not a problem for a professional user to operate this card using Real Time Toolbox but this will not happen to a beginner or an intermediate user. For example, this card is widely used in educational laboratory for the purpose of experiments or practical courses. The beginner will not be able to familiarize a new tool in short time. Thus, there must

be a solution to overcome this problem. One of the better possible solutions would be Graphical User Interface (GUI) development for the Real Time Toolbox. The GUI that was developed can overcome the above problem. Therefore, the beginner user will fascinatingly learn the any related course using Real Time Toolbox in very easy way without consuming a lot of time.

Majority of the tools the author presented herein deals with the implementation issues associated with the development of a graphical user interface in the MATLAB environment. GUI is a matter of complication since it requires high level language programming. Graphical User Interface Development Environment (GUIDE) is one of the interesting alternatives to solve complicated coding or programming such as Visual C and Visual Basic that came after FORTRAN or BASIC which are more trivial in developing GUIs [1]. Approaches to provide the microcontroller with GUI capabilities are by interfacing it with MATLAB and by exploiting MATLAB's abundant GUI tools. The proposed Matlab-based GUI environment for the microcontroller relies on the use of serial communication between the microcontroller and a personal computer. One of the advantages of microcontrollers is its cost-effectiveness which is purposely build for specific

application. However it does not has graphical user interface that contributes to serious limitation for the microcontroller-based developers. In dealing with this problem third party GUIs tools are often inadequate for sophisticated applications [2]. All functions and their parameters are controlled via graphical user interface and the information or data can be imported, exported and graphically plot. Interestingly, the results shown in the GUI can be saved to file [3]. Furthermore, a GUI is a graphical display that contains devices, or components, that enable a user to perform interactive tasks. To perform these tasks, the user of the GUI does not have to create a script or type commands at the command line. Often, the user does not have to know the details of the task at hand [4] [5]. The GUI components can be menus, toolbars, push buttons, radio buttons, list boxes, and sliders — just to name a few. In MATLAB, a GUI can also display data in tabular form or as plots, and can group related components [4] [5] [6].

2 Approach and Methods

The previous section presents the MATLAB GUI design tools and the various GUI controls were used extensively throughout the GUI development process. A discussion here focuses on the design of tutorial stage. It is during this stage that the functionality of the GUI is assessed, and the complete layout determined. These assessments form an iterative process in the sense that the overall functionality of any design tool is greatly dependent on the ability of the user to understand the design layout. The purpose of the MATLAB Real Time Scope GUI is already explained in the introduction. The various figure windows of the GUI will be thoroughly discussed and explained.

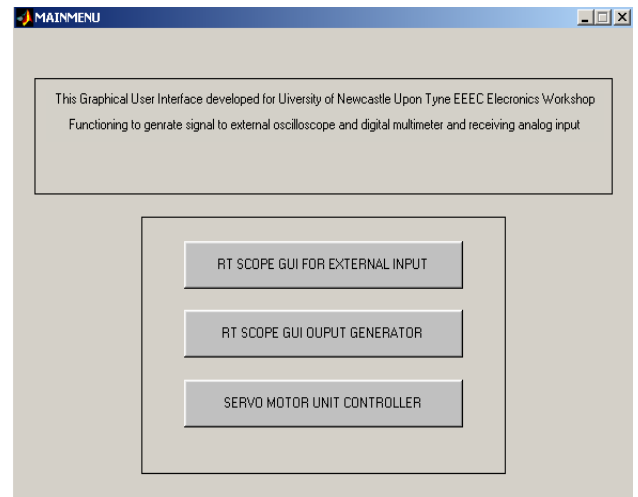


Figure 1: Real Time Scope GUI Main Menu Window

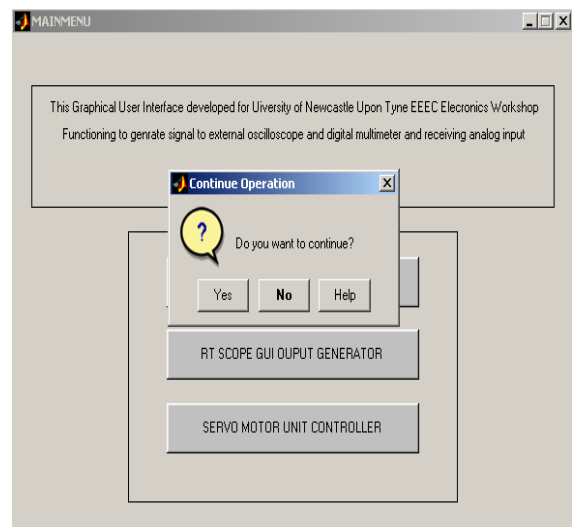


Figure 2: Real Time Scope GUI Main Menu Window Question Dialogue

The above window is the main menu of the GUI. It contains callback to link to Receiving Signal Input Window (RSIW) M-file and Sending Output Signal Window (SOSW) M-file. This window also contains menu that provides user with help menu. The help menu assists users how to operate both RSIW and SOSW.

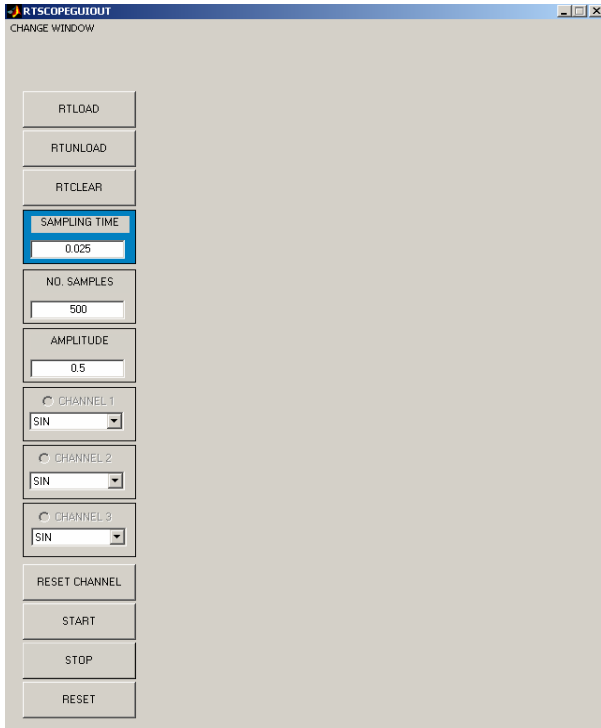


Figure 3: Real Time Scope GUI Sending Output Signal Window

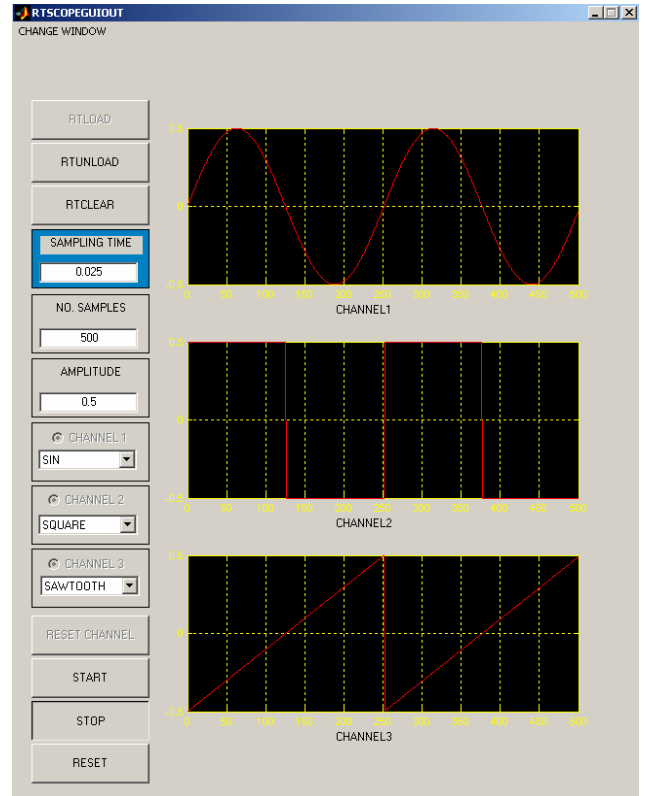


Figure 4: Multichannel Real Time Scope GUI Sending Output Signal Window

In this window, there are several components created to load, unload and clear AD616 driver instantly and interactively.

3 Result

To initialize the signal generator, RTLOAD button must be clicked in order to automatically configure the DAQ card and access the M-file programming code. Subsequently, users must specify the desired parameters in the edit text box of sampling time and number of sample. Then user can select desired channel to be generated. As shown in Figure 4, three channels are selected for a different types signal generator. The signals are plotted based on sampling time of 0.025 seconds with 500 numbers of samples. Then these signals are compared with the signal display in the external oscilloscope. Comparatively, it shows that the monitoring of GUI and external oscilloscope are almost the same as shown in Figure 5.

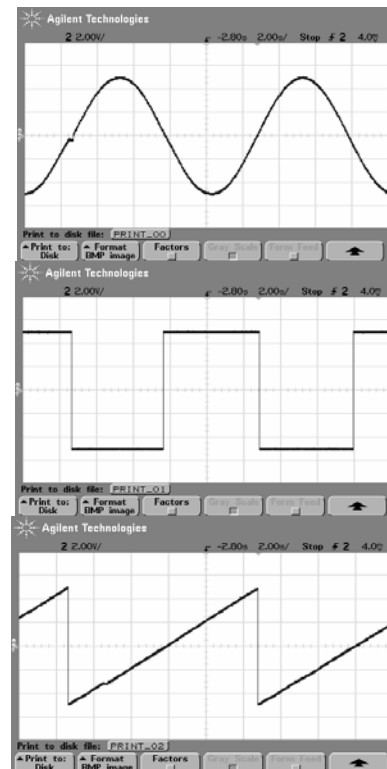


Figure 5: Output Three Channels Signal to the External Oscilloscope

4 Discussion

MATLAB has a tool called SIMULINK. This tool provides necessary block libraries such as source and sink for simulation purposes. In sink libraries, there is block called scope. Basically, the design of the real time signal plotting using M-file based on this scope block. This scope is the initial step to simulate using M-file but with some limitations of optimisation. Among of the limitations are zooming in and out, auto-scale properties and a smooth plotting. The real time plot looks flickering while running. This problems solved by using "set(gcf,'DoubleBuffer','on')" under the function varargout.

5 Conclusion

The Real-time Scope GUI for signal generator was successfully developed. The Real-time Scope GUI for signal generator can be used without having to compile and execute any code. One of the main advantageous of this GUI is that it can monitor various signals from different channels at the same time. Since this GUI is M-file-based, optimization of certain features can be upgraded easily. For the future improvement, this research can be upgraded with additional number of channels instead of three channels. This will improve the flexibility of the GUI itself. Another improvement that can be made is by applying edit text box to generate special signal with either complex linear or non-linear equation. Finally, there are many types of controller such as fuzzy controller or neural network can be added to the controller unit to have more options for the controller unit.

References:

- [1] Demoyer, Demoyer, R., Mitchell, E.E. (1999) Use of the MATLAB Graphical User Interface Development Environment For Some Control System Application. Proceeding. ASEE/IEEE Conf. Frontiers in Education, San Juan : 12b3-7 - 12b3-11.
- [2] Yan-Fang, L., Saul, H., Hong, W., Vikram, K. (2004) Matlab-Based Graphical User Interface Development For Basic Stamp 2 Microcontroller. Proceeding. AAC, Boston : 3233-3238.
- [3] Weiss, M., Rees, D.; Evans, C.; Jones, L. (1995) The Design Of A Graphical User Interface To Perform Complex Identification Tasks. IEE Coll on Man-Machine Interfaces for Instrumentation, London : 6/1-6/4

[4]http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/buildgui.pdf

[5] Chapman, Stephen J, (2001) MATLAB Programming for Engineers, Brooks Cole.

[6], Jeffrey A. Webb, Creating Graphical User Interfaces (GUI's) with MATLAB, Jeffrey A. Webb OSU Gateway Coalition Member