

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



LabVIEW and Open Embedded System

Trinh Quang Duc

Abstract

In this chapter, discussions about the applications based on LabVIEW with a typical open embedded system such as Arduino. Instead of the applications with use of National Instrument hardware, an user can build an low-cost microcontroller-based system which user interface and procedure designed in LabVIEW. Recently, Arduino-based applications are one of interested developments, the embedded system is designed with the low-cost microcontroller of ATMEGA2560 and easy to program with IDE (Interface Development Environment). However, the difficulties of Arduino-based applications are user interface design while LabVIEW is an excellent utilities for the panel designs. The limitation of some LabVIEW applications is the requirements from combination with high performance NI hardware. Under interactions between LabVIEW and IDE library designed by Arduino, the low-cost system can be easily built for experiments or prototype designs.

Keywords: LabVIEW Embedded Systems, Biosignal Acquisition, LIFA toolbox, LINX toolbox, LabVIEW Arduino Interfaces

1. Introduction

Embedded system requires the interaction between hardware and software installed in PC (Personal Computer). Before LabVIEW, a software designed for hardware devices seems to be hard work job because of the communication protocols are need to be programmed in both hardware device and PC. For instant applications which aim to measurement data acquisition for analysis algorithm development or process control as well as controller design, the hard work in programming field may restrict the destination job. To solve the problem, National Instrument introduced LabVIEW as a graphical programming tools with a series of compatible hardware devices [1]. This tool has been widely chosen by the non-professional user community. The NI hardware communicate with PC through some communication protocol such as CAN (Controller Area Network), GPIO (General Purpose Input/Output), SPI (Serial Peripheral Interface), PROFIBUS (Process Field Bus), PROFINET (Process Field Network) and LIN (Local Interconnect Network), are officially equipped with software called driver licensed by National Instrument. The independent hardware developments for LabVIEW compatibility, hence, are not easy.

Recently, some multiple purpose low-cost open-source hardware modules based-on micro-controller were developed by commercial companies such as Arduino [2], NodeMCU [3]. This device connects with PC using USB port and transceivers data through a simple embedded program set on the flash memory of the micro-controller. With data acquisition and control applications which the

discrete sampling as well as response frequency not over than 100 kHz, this device will be the considered options for experimental arrangement. Since the first time released in 2005 [4], Arduino is used to develop many applications in automation systems. However, until 2013, a proposal for LabVIEW interface toolkit as LIFA (LabVIEW Interface for Arduino Toolkit) was announced [5] and introduced [6]. Another interface named as LINX designed by Digilent [7]. So far, these toolkit is widely known for the users using Arduino and Raspberry Pi in LabVIEW application for embedded system.

The LabVIEW VI is programmed to communicate with hardware devices though a package called NI-VISA [8]. This package provides some interface with communication protocol such as PXI, GPIB, VXI, Serial, USB. The LabVIEW compatible hardware devices has to be set their communication procedures in the flash or ROM memory to handshake with the VI program, then, the data can transfer from the devices to PC and the command can receive from the PC to the devices. The main problems of any embedded system is how to build the communication procedure set on the device to match with the communication procedures installed in LabVIEW designed by National Instrument. This is solved by some independent projects such as LIFA and LINX which the communication procedure is designed.

2. LINX and LIFA and the embedded model

LIFA and LINX are the open-source but their approaches are different. LIFA provides a base succeed procedures installed in the file named as LabVIEWInterface.h, the users, based-on this procedure can program other procedures in the hardware devices to control their functions. To use LIFA, beside the graphical programming in LabVIEW, the procedure programming for the device firmware is required. On the other hands, LINX considers the LabVIEW graphical programming rather than the firmware programming, the compatible firmware are designed the loaded to the correspondent hardware devices respectively, therefore, with a simple functions, the embedded system with LINX is simpler than use of LIFA. Because of that, LINX is better error handling but does not flexible to develop functions like LIFA. For some unique embedded system, LIFA is preferred than LINX.

Figure 1 show diagram of typical embedded system using LINX and LIFA. To use LINX, users should install the package into LabVIEW as a function of the LabVIEW and then choose a compatible firmware to load into the hardware flash

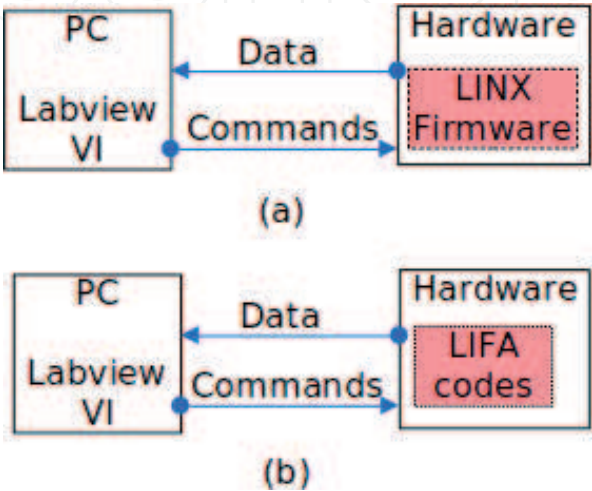


Figure 1.
The diagram of LINX system (a) and LIFA system (b).

memory, the diagram shows in **Figure 1(a)**. The LIFA users, instead of uploading the firmware as the LINX users do, edit the LIFA codes in IDE software (Integrated Development Environment) to function their system then upload to the hardware devices. For the graphical programming, both users have to download the packages and install them through VIPM (VI Package Manager), then the modules included in LINX and LIFA will appear in the menu windows of LabVIEW program. To communicate with the hardware devices, the LabVIEW graphical program can use the modules of read and write presented in the packages to upload commands to the hardware devices or download data to the PC.

Because of the independent projects to National Instrument, both LINX and LIFA occurred errors which caused by the compatibility with LabVIEW versions especially in LIFA applications, the common errors are 5002 and 5003 represented by the lost connection establishment. The LINX applications do not present the such errors and contains more various hardware firmware compatible with Arduino versions and Raspberry PI as well, however, the setup for sampling frequencies for data acquisitions are limited and much depended on the loop LabVIEW cycle. Therefore, depending on the requirements of applications, users can choose the model of LINX or LIFA for their embedded systems.

There are number of studies used LINX to apply Arduino module for low-cost embedded systems such as 3D object reconstruction with 3-axis scanning [9], temperature acquisition and control [10], DC motor control using PID (Proportional Integrated Derivative) controller [11], study transient response of RC circuit [12], an control the traffic signal [13]. Almost of the studies focused on the process control where the objects responses are limited in few milliseconds considering to few kilohertz sampling frequency. In the case of requirement for faster sampling frequency which has to be set on an optimum code to obtain, because of the fixed configuration, the LINX toolbox should not be chosen.

For the real-time applications using Arduino and LabVIEW, a number of users considered LIFA as the toolbox for LabVIEW which toward embedded systems based on Arduino. Some typical applications such as a real-time simulation of the PV (Photo-Voltaic) panel [14], a ECG 12 leads (ElectroCardioGraphy) signal real-time observation [15], a digital stethoscope for body sound signal real-time collection [16], a design of real-time battery monitoring [17], Robotic arm real-time controller design [18], and a water level observer implementation [19]. Because of open source in LIFA and the users have to program the firmware procedures in Arduino code, the sampling frequency can be improved. Here is the reason that LIFA is preferred for professional users who are requested some skills in programming for hardware device such as micro-processors or micro-controllers.

For the LINX users, after the correspondent firmware chose and uploaded to the flash memory of the hardware devices, the graphical programs designed in LabVIEW response for all functions of the embedded systems. An example of the program is shown in **Figure 2** represents a typical analog signal acquisition using LINX. To initiate the program, a block named as “Open” addresses the in/out port of the connected hardware devices, here, is used of Arduino Uno which the port is defined as serial port through USB connection.

As the figure detailed, the block called “Analog Read” collects the signal from the channel corresponding to the analog input pin of the Arduino Uno module. The signal acquisition procedure is performed in the while loop until users stop it through the stop button or an error occurred. When the while loop is stopped, the block of “Close” terminate the program procedure. Here, to control the sample frequency a block named as “Wait” can be used. When the “Wait” block added with a constant or control parameter, the cycle of the while loop is controlled, hence, the sampling frequency is handled. Because of the dependency of the while loop cycle

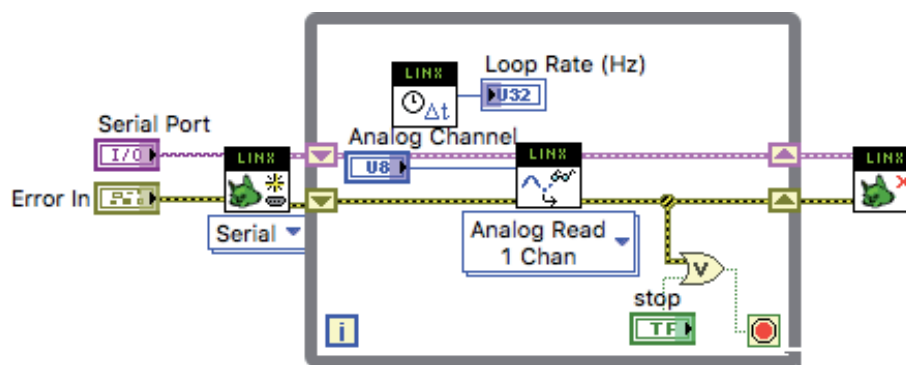


Figure 2.
Diagram of a typical acquisition LabVIEW program using LINX.

to the computer system cycle, the sampling frequency in the case of LINX using is not stable. Moreover, the hardware sampling frequency is set as constant in LINX firmware, therefore the embedded system sampling frequency should be slower than the value set in the firmware.

To use the LIFA, the applicants should compile a code through the IDE software. The mostly simple code can work with LabVIEW program called “LIFA_Base.ino” which usually packed in the LIFA package software. The codes are listed in **Figure 3** below:

```
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include "LabVIEWInterface.h"

void setup()
{syncLV();}

void loop()
{checkForCommand();
if(acqMode==1)
{sampleContinuously();}
}
```

Figure 3.
The codes written in LIFA Base.Ino file.

where, the `syncLV()` is a function defined in `LabVIEWInterface.h` which provides a handshaking procedure with LabVIEW NI-VISA port configuration. For Arduino hardware devices, the USB port assigned as serial communication is used. When the connection with LabVIEW program is synchronized, both loops of the LabVIEW program and the Arduino firmware are operated. The loop in the Arduino firmware always check the code sent from LabVIEW then run the correspondent procedures otherwise the code of “`sampleContinuously()`” is operated. The sampling procedure of the LIFA firmware requires a sampling frequency parameter to control the sampling cycle. Therefore, for the LabVIEW program, the users can completely control the sampling time which is one of the significant parameters for embedded systems especially in discrete control and data acquisition.

Figure 4 shows a typical acquisition LabVIEW program. The connection process initiated by the block called “Init”. Throughout the initialization procedure, the parameters include baud rate, communication, Arduino board type, communication protocol, and data packet sized should be configured. When the connection is successful established, the analog signal is read via an analog input pin with

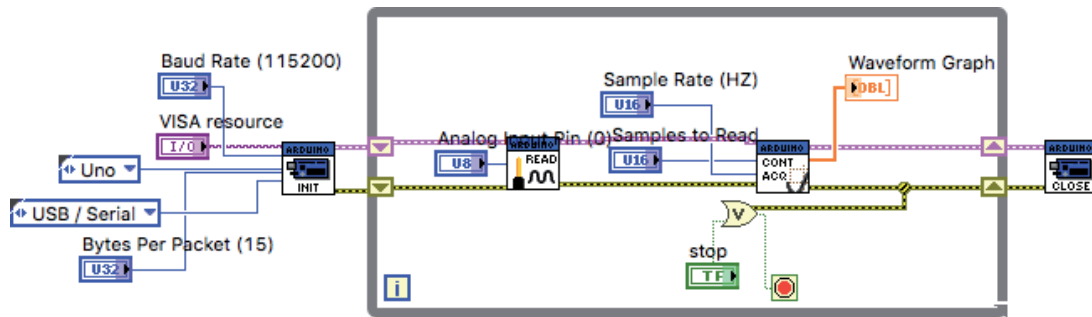


Figure 4.
 Diagram of a typical LabVIEW program using LIFA.

the block of “Analog Read”, then, the converted numerical data collected through the block named as “Continuous Acquisition” located on the Arduino menu in the popup programming menu. The data acquisition procedure requests parameters such as “samples to read” and “sample rate” to collect data with a certain sampling cycle and a pack of samples. The collected data is used to display with a graph panel called “Waveform Graph” originated in LabVIEW software. The data acquisition is stopped by the block named as “Close” to release the PC memory.

Unlike the procedure using LINX, the data acquisition program using LIFA does not depend on the loop cycle in PC since the sampling frequency is configured to the Arduino board and with requirement of each issues, the sampling time can be adjusted appropriately. However, the users have to code the program for hardware device to adapt with the LabVIEW program. Beside the hardware coding skills, the compatibility of LIFA firmware with LabVIEW is also a problem. With each version of LabVIEW software, some LIFA package version has bugs errors of connection such as 5002 and 5003 but this errors does not occurred while use of another LIFA package version. Therefore, the user should test their program with some LIFA package version to complete the embedded system.

3. Examples in signal processing

One of the application categories in use of embedded system is digital signal processing. The embedded system with numerical processes is much convenient than the analog or hardware process even if the low response time in the numerical system. LabVIEW is one of the options for embedded system development. The integration of LabVIEW program with hardware devices such as Arduino is great deal to reduce the cost. In this section, there are two examples are introduced: ECG signal processing and body sound signal processing.

ECG signal is a biomedical signal used to diagnose pathology related to cardiovascular disease. A simple ECG measurement system is designed with 3 leads placed on to human limbs to observe flutter wave and detect the complex waves include P, Q, R, S, T. To obtain the spatial signal of a heart, the electrodes should be placed on the chest with 6 leads assigned from v1 to v6. Here, with 4 electrode placed on all limbs, the 12 leads ECG measurement system is built. The 10 electrodes are used to monitor 12 signal of a cardiovascular system from the limbs to heart.

The low-cost hardware design for 12 leads ECG measurement based on Arduino Uno was reported [15]. To obtain the signal for storage and analysis, the signal is numerical converted and transferred to PC with software. For convenience, LabVIEW programming software was chosen. Because of the low spectrum range of ECG signal (0 to 100 Hz), the required sampling frequency is also low as well. The amplitude of ECG signal is weak (millivolts level), to observe the signal the

amplitude have to be amplified with gain ranged from 30 dB to 40 dB, therefore, the ECG signal is contaminated with a number of the gained electromagnetic noises in the spectrum range.

To reduce the electromagnetic noises as well as increase SNR (signal-to-noise ratio) of the signal, beside the analog filters [15], digital filters is also considered since the difficulties of analog high-order filters design. After the signal purification, the wave of signal should be displayed then stored in PC. Therefore, the function diagram of a LabVIEW program contains signal acquisition, filtering, display, and storing. To compare the filtered signal and the raw signal, two waveform graph panel are used. The diagram has shown in **Figure 5**.

As **Figure 5** presented, the data is transferred to the data acquisition block and displayed in a graph plotting as well as filtered by the signal filtering. After the noise reduction processes, the filtered data is displayed and then stored in PC storage. Because of the low sampling frequency requirement, the LabVIEW-Arduino Uno interface can be chosen as LINX. By use of LINX package, the sampling cycle can adjusted through the “wait” block in LabVIEW to limited the while loop cycle. Typically, with the spectrum range of ECG signal, the maximum frequency reaches to 100 Hz, hence, the minimum sampling frequency should be 200 Hz as the Nyquist criteria requirement. However, the enlargement of sampling frequency provides better frequency resolution under frequency analysis.

The 12 leads ECG system contains 8 channels, includes V1 to V6, lead I and lead III. **Figure 6** shows the analog read program which was built with LINX and LabVIEW The data acquired through a block called “Analog Read N Channels”, this block enable to read multiple channels simultaneously. Because the sampling cycle of the embedded system depends on the LabVIEW while loop cycle, therefore the

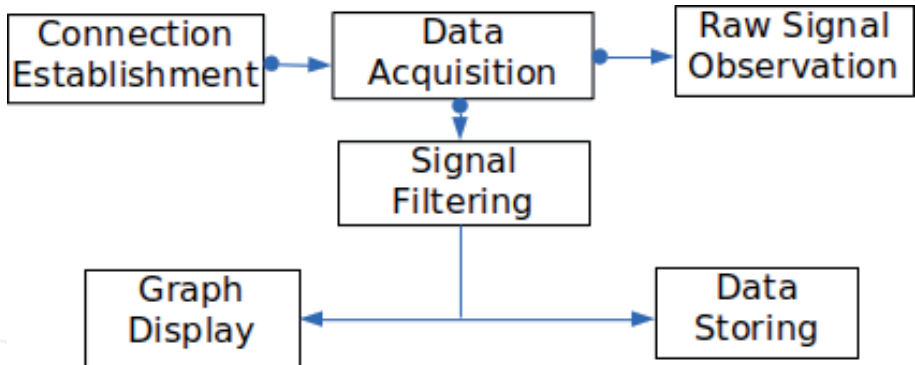


Figure 5.
Function diagram of the LabVIEW ECG signal acquisition.

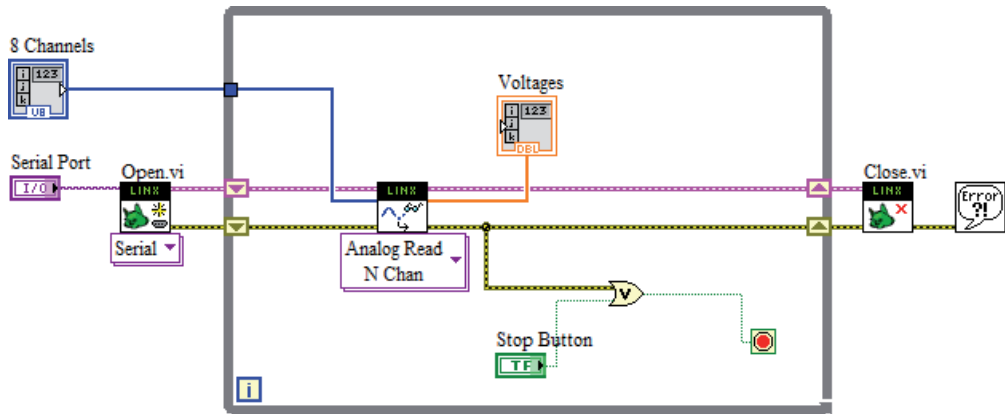


Figure 6.
The LabVIEW analog read program for 12 leads ECG measurement.

sampling time is set via “Time Delay” block. For this development, the sampling cycle is set with value of 0.003 seconds, corresponding to sampling frequency of 334 Hz approximately. The setting of the sampling cycle is shown in **Figure 7**.

Because of the multi channels reading, the data is packed in to a 1D array. To separate the data in each channel, a block named as “Index Array” was used. With each indexed number, a correspondent numerical value is extracted to an indicator. Based on the extracting in multi loops, the numerical values is arranged as a data 1D array which can be used to plot the raw data of each channel. The extraction program is indicated in **Figure 8**. Here, because the data indexed from 0, the channel number of 0 to 7 is assigned to extract the data of correspondent channels. The lead indicator named as the analog input pin corresponding to the electrode of the hardware device.

To reduce the noises from the collected data, a bandpass Butterworth filter was used. The noises contain both DC (Directed Current) components and AC (Alternative Current). In a signal waveform, AC components are the significant information that the observer can use for analysis. To eliminate the DC components in the ECG signal, a highpass filter with zero-close cut-off frequency should be addressed. To maximize the spectrum range, the zero-close cut-off frequency is chosen as 0.05 Hz while the high cut-off frequency of the low-pass filter is located at 100 Hz. The combination of the low-pass and the high-pass filter is the band-pass filter with the passed frequency range of 0.05 Hz to 100 Hz. The filter LabVIEW diagram is shown in **Figure 9**. Since the narrow range from stop frequency to cut-off frequency requests pitched edge, the filter order should be high. For this ECG spectrum range, the order of the band-pass filter is set as 6.

LabVIEW waveform graph block plots a waveform with 1D data typically in a linear variation of counted number of sample index. Therefore, the waveform

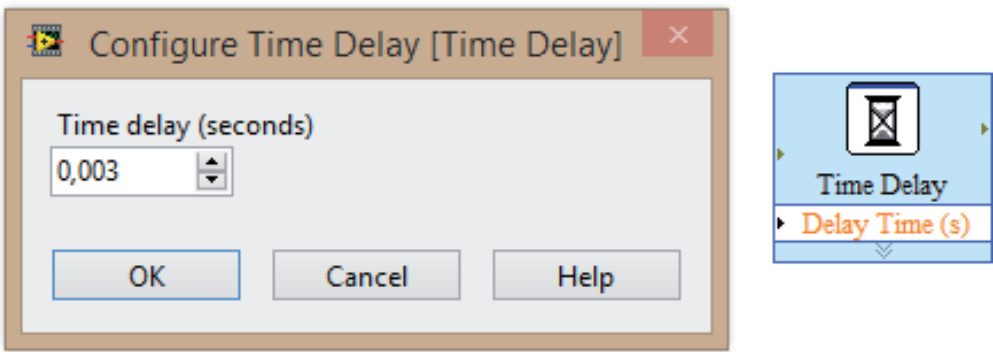


Figure 7.
The sampling cycle setting time with “time delay” block.

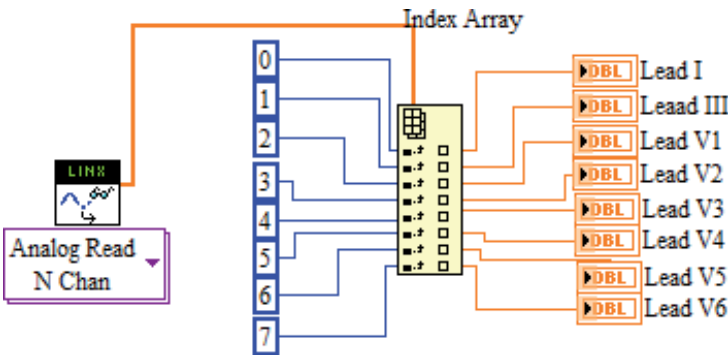


Figure 8.
The multi-channels data extraction program.

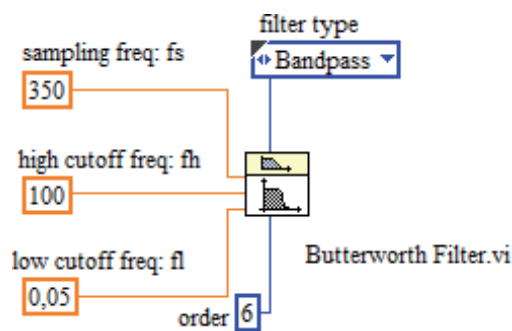


Figure 9.
LabVIEW diagram of the signal filtering program.

cannot be observed in the time-domain precisely. To indicate the graphs amplitude distribution in time-domain, the real-time clock can be combined with the ECG numerical data to be 2D data array, this process so called waveform building. Based-on the waveform building, the data plotted in a graph panel named as “Waveform Chart” in multiple channels display. The diagram of waveform building of amplitude data and real-time clock combination has shown in **Figure 10** while the array of the such 8 channel is presented in **Figure 11**.

The function of data storing can be designed with “creat file” block in “File I/O” menu. However, the data acquisition and file create in the same “while loop” may delayed the cycle of the loop. Therefore a structure Producer-Consumer which allow the parallel jobs; independent loops. By the structure, the received data in the first loop is enqueue then transmitted to the second loop named as dequeue and queue for data obtaining. The procedure is shown in **Figure 12**. With such program structure, the obtained data can be stored parallelly with the data acquisition

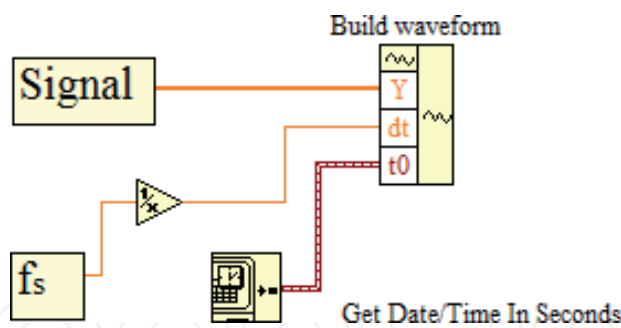


Figure 10.
The waveform building of a single channel amplitude data with the correspondent real-time clock data.

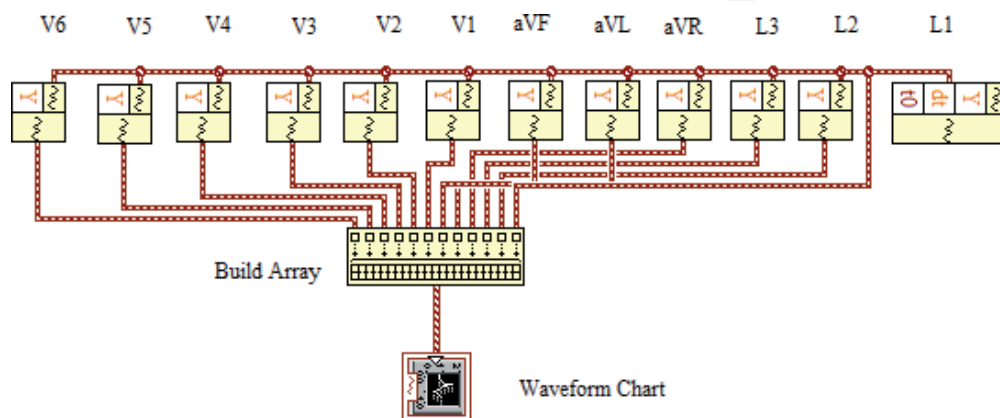


Figure 11.
The waveform building of 8 channels with amplitude and real-time clock data combination.

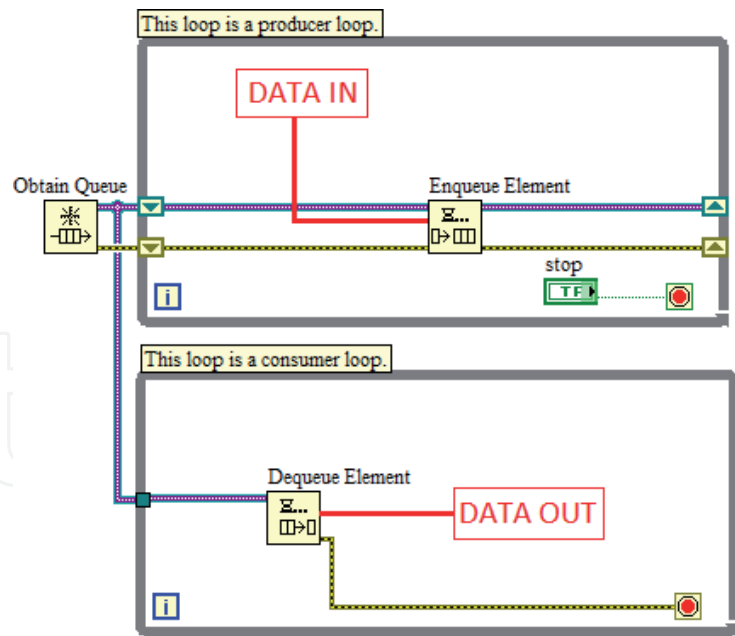


Figure 12.
The producer-consumer structure program for data storing.

procedure with the jobs of signal filtering, waveform building, graph plotting. This structure is also used in the case of that the variable need to be processed in multi loop arranged in the same program with synchronization requirement.

Another approach for data acquisition system with precise sampling frequency is LabVIEW applications with LIFA package. For LIFA users, as above discussion, the compatibility with each LabVIEW version should be checked since the NI-VISA of each LabVIEW version does not support for all LIFA packages, causing to error connection. Eventhough, the embedded system development community with LabVIEW still prefers LIFA because of simplicity and precision. With a requirement of fast sampling frequency, LINX might not be satisfied. In the second example, the observed objects are body sounds which their spectrum ranged in 0 to 20 kHz, the maximum frequency is much higher oscillation than the ECG signal. To observe the whole spectrum, at least, the minimum sampling frequency should be set at 44 kHz to satisfied Nyquist criteria. A test with this parameter based-on LINX application resulted the instability of the sampling frequency.

To obtain experiments of embedded systems collecting the body sound data such as heart sound and lung sound, a wide range stethoscope was designed [20]. The body sound is transduced to electromagnetic wave then converted to numerical data by Arduino board and read by a LabVIEW program. Instead of LINX, LIFA is used in this development. Similarly to the procedure has shown in **Figure 4**, the

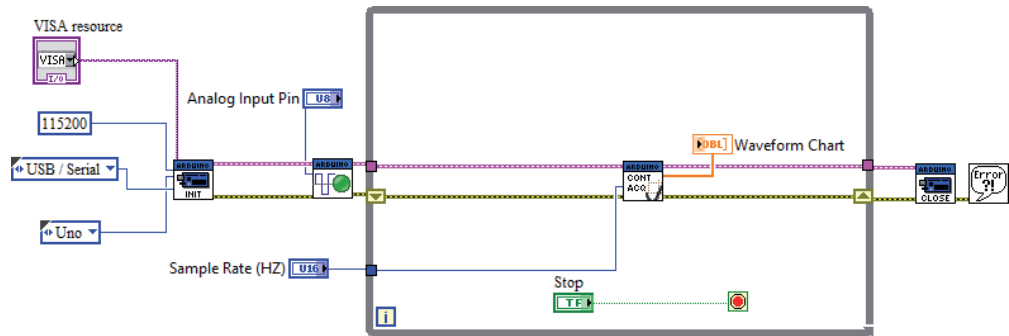


Figure 13.
The LabVIEW body sound data acquisition based on LIFA applications.

LabVIEW data acquisition program based-on LIFA is shown in **Figure 13**, where the block of “Analog read pin” is located outside of the while loop to avoid the replication of the setup pin mode for each cycles. Here, the sampling frequency can directly input to the “control” block as 88 kHz to satisfy the requirement of whole spectrum range observation. For the compatibility, this experiment was check with all current LIFA package with LabVIEW 2018 sp1. The result suggested that the package of 1.3.0.26 is great deal for this development without any errors.

Because of the conversion from the acoustic wave to the electromagnetic wave, the signal is contaminated with electromagnetic wave noises, hence, the noises should be reduced to enhance the SNR for the observation. To ensure the noise minimization, the digital band-pass filter is applied with cut-off frequencies are set at 5 Hz for low frequency cut and 35 kHz for high frequency cut, respectively. This block is different to the “Butterworth band-pass filter” which used in ECG acquisition system that the order of the filters are calculated based on the cut-off frequency and stop frequency parameters. The procedure application is much easier than the use of the filter block such as above Butterworth configuration. This block is packed in the toolkit named as “Digital Filter Design Toolkit”. The filtering procedure diagram is shown in **Figure 14**.

During the sound data acquisition, beside the storage for future use, the sound as well as the quality can be checked directly by the hearing of users. To reconstruct the acoustic oscillation, the data converted from the discrete format to analog wave through 24 bit DAC (Digital to Analog Conversion) sound card integrated in PC. The signal generates the body sound with a speaker with built-in amplifier. The functional procedure is shown in **Figure 15**, where the sound amplitude possibly adjusted with a digital amplifier called as “Set Volume” and generated to the output phone jack through the block of “write”. Both blocks are located in Graphics and Sound menu.

Both applications based-on use of LINX and LIFA are introduced for examples to visualize the low-cost embedded system design with Arduino and LabVIEW.

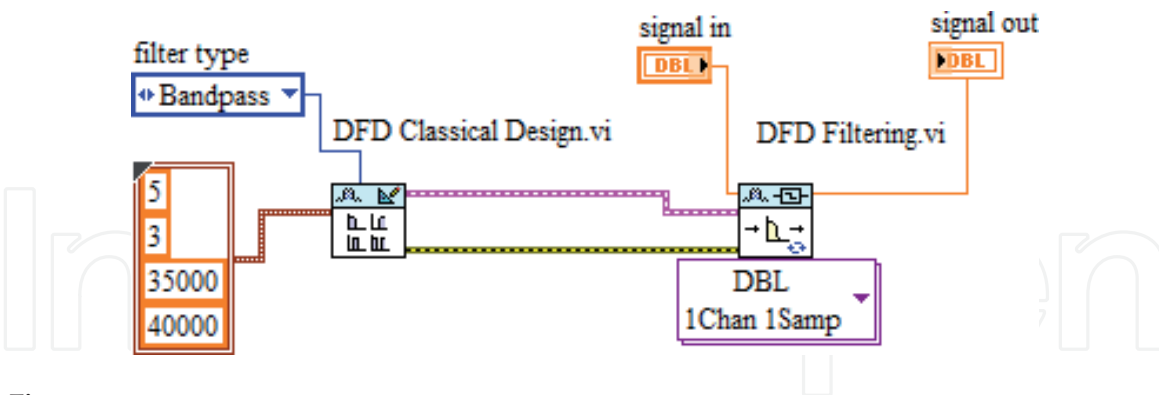


Figure 14.
The filter applying to the body sound signal in the LabVIEW program.

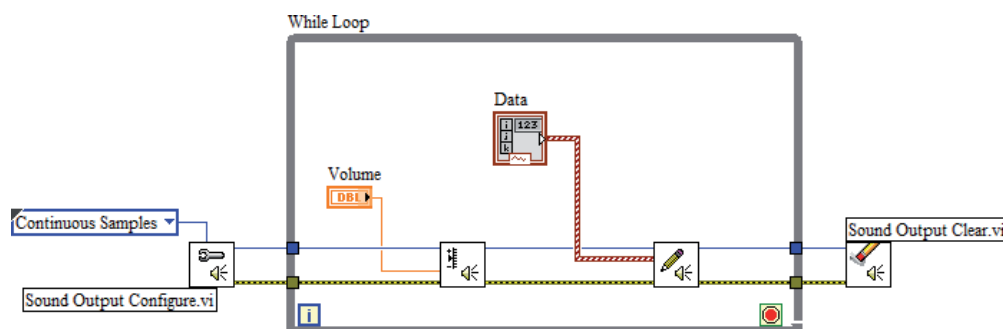


Figure 15.
The program with function of direct sound regenerating.

Even if the current drawback of the toolboxes are available, the development still can be solved in certain circumstances. With the experiment conditions and by the demonstration, these developments suggest another applications which toward the embedded system building without the traditional NI hardware devices. However, these development still need more work to complete the issues since the bug and error may occurred in some unknown conditions.

4. Another approaches

Beside Arduino boards, there are many commercial products are still developing such as Raspberry pi for remote access network or IoT (Internet of Things) system, chipKit for function enhancement, ESP8266 for wifi applications. Recently, to open a change for mobile AI (Artificial Intelligent) system, NVIDIA company released Jetson nano kit which specified with quad-core CPU and 128-core GPU. Almost of current embedded hardware devices use USB connection with asynchronous communication protocol for data exchange. This means with the definition of NI-VISA which support for the hardware using USB port, the commands and the data can be full-duplex transferred between PC and the embedded hardware devices. The requirement for LabVIEW interface can be solved through firmware library programming.

In the case of large amount of logic components integration required such as parallel filters, numerical array convolution, the embedded system FPGA-based (Field Programmable Gate Array) with LabVIEW can be considered. Unlike the LabVIEW interface package such as LINX and LIFA, the FPGA-based embedded system has to be originated by NI hardware devices or use of other hardware compatible with NI-myRIO platform. With the optimization of logic circuit with specified functions or procedure, the fast response of the FPGA-based embedded systems possibly satisfy the real-time requirement. An example of such application was reported in 2010 with real-time response for DC motor controller design [21].

The advantage of LabVIEW embedded systems that is the enriched update of libraries and block module in each version which support for data processing such as signal processing, image processing, network routing, control system, computer vision, IoT, and more. In large scale system level, when the real-time characteristics is required, the distributed system with multiple computers network possibly the solution. Therefore, for the LabVIEW embedded system in the near future, a trend of functionalizing hardware device is potential development. With the standardization of the data transfer protocol, the computer peripheral design based on embedded system will focus on the firmware and software design rather than the hardware design as before.

With the development of the operating system for smart devices such as Android and iOS, NI LabVIEW also provide a solution that the smart devices users can observe acquired data and control system objects distantly through a service called WebVI, which means the embedded systems in the future can be observed and control through smart devices such as smartphone, tablet when they are connected to internet, an IoT solution of LabVIEW application development [22]. Based-on this enhancement, the category of applications toward IoT system can be widely developed in the near future.

5. Conclusions

In this chapter, embedded systems based-on LabVIEW applications with low-cost hardware devices are introduced and discussed. For the application

developments, the topic of LabVIEW interface with hardware device is focused with use of LIFA and LINX package. The similarity and the difference between both interface package is also discussed and analyzed. To visualize the embedded system design process LabVIEW-based, two experiments in data acquisition and signal processing with use of LINX and LIFA package were introduced as well. LIFA package provide the simplicity in use but remain the problem of compatibility with each LabVIEW version while LINX package is wide range of hardware devices support but some significant characteristics are very difficult to configure. To adapt with a number of modern smart devices currently, LabVIEW also open a service to support IoT application category. This support promises the potential development in the near future.

Acknowledgements

This work is contributed by students who used to study in Biomedical Engineering Laboratory belonging to Hanoi University of Science and Technology. The author also appreciate the editor and secretary for encouragement to complete this chapter during the Covid-19 epidemic.

Author details

Trinh Quang Duc
Hanoi University of Science and Technology, Hanoi, Vietnam

*Address all correspondence to: duc.trinhquang@hust.edu.vn

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] LabVIEW. Available from: <https://en.wikipedia.org/wiki/LabVIEW> [Accessed: 2021-04-15].
- [2] Arduino. Available from: <https://en.wikipedia.org/wiki/Arduino> [Accessed: 2021-04-15].
- [3] NodeMCU. Available from: <https://en.wikipedia.org/wiki/NodeMCU> [Accessed: 2021-04-15].
- [4] David Kushner. The Making of Arduino. IEEE Spectrum. 26 Oct. 2011.
- [5] LIFA (LabVIEW Interface For Arduino). Available from: <https://github.com/labviewhacker/lifa> [Accessed: 2021-04-15].
- [6] Marco Schwartz and Oliver Manickum. Programming Arduino with LabVIEW. Packt Publishing; 27 Jan, 2015.
- [7] LINX: How LINX Works. Available from: <https://blog.digilentinc.com/how-linx-works/> [Accessed: 2021-04-15].
- [8] NI-VISA: Overview. Available from: <https://www.ni.com/en-vn/support/documentation/supplemental/06/ni-visa-overview.html> [Accessed: 2021-04-15].
- [9] Rob A, Panoiu C. Using LabVIEW Linx for Creating 3D Objects. IOP Conf. Series: Materials Science and Engineering; 447(2019) 012027. DOI: 10.1088/1757-899X/477/1/012027.
- [10] Yusuf Ayuba. Temperature Control and Data Acquisition Method for Factory using LabVIEW. International Journal of Computer Engineering and Technology; 2016, 7(2), pp. 01-14.
- [11] G. Gaşparesc. PID control of a DC motor using LabVIEW Interface for Embedded Platforms. *12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania, 2016, pp. 145-148, DOI: 10.1109/ISETC.2016.7781078.
- [12] N Suwondo and D Sulisworo. Hands-on Learning Activity Using an Apparatus for Transient Phenomena in RC Circuit Based-on Arduino UNO R3-LINX-LabVIEW. International Journal of Online and Biomedical Engineering. 2017, 13(1), pp. 116-124.
- [13] Mihai Bogdan. Traffic Light Using Arrduino Uno and LabVIEW. The 12th International Conference on Virtual Learning ICVL 2017; 2017, pp. 287-290.
- [14] A El Hammoumi, S Motahhir, A Chalh, A El Ghzizal and Derouich. Realtime virtual instrumentation of Arduino and LabVIEW based PV panel characteristics. International Conference on Renewable Energies and Energy Efficiency (REEE2017). IOP Conf. Series: Earth and Environmental Science 161 (2018) 012019 DOI:10.1088/1755-1315/161/1/012019.
- [15] Son Nguyen Van, Duc Trinh Nguyen, Giang Nguyen Hoai. Development of a Low-Cost Arduino-Based 12-Lead ECG Acquisition System And Accompanied LabVIEW Application. International Journal of Engineering and Advanced Technology. 2019, 09, pp. 1641-1648.
- [16] Son Nguyen Van, Duc Trinh Quang, Giang Nguyen Hoai, Quynh Nguyen Thi Huong, Khanh Pham Xuan. Software Design Collection and Handling of Signal Sound Body, International Journal of Engineering and Advanced Technology. 2019, 09, pp. 260-262.
- [17] A. Jamaluddin, L. Sihombing, A. Supriyanto, A. Purwanto and M. Nizam. Design real time Battery Monitoring System using LabVIEW Interface for Arduino (LIFA). *2013 Joint International*

Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T), Bandung, Indonesia, 2013, pp. 1-4, doi: 10.1109/rICT-IceVT.2013.6741525.

[18] B. Harish, U. Jyothsna. LabVIEW Interface with Arduino Robotic ARM. *International Journal of Science and Research*. 2015, 4(11), pp. 2423-2426.

[19] K Premsagar, M. Akhila, P. Prashanthi Reddy, P Srinivas. Unused Water Level Observing Structure Implementation Using LabVIEW and Arrduino. *Journal of Critical Reviews*. 2020, 7(12), pp. 1344-1350.

[20] Trinh Quang Duc, Nguyen Van Son, Nguyen Hoai Giang. A Developed Design to Enlarge the Spectrum for Analog Electronic Stethoscope. *Journal of Engineering Science and Technology Review*. 2019, 12, pp. 179-184.

[21] F. H. Ali, M. M. Hussein and S. M. B. Ismael. LabVIEW FPGA implementation of a PID controller for D.C. motor speed control. 2010 1st International Conference on Energy, Power and Control (EPC-IQ). 2010, pp. 139-144.

[22] LabVIEW support for iOS and Android Operating Systems. Available from: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019NNjSAM&l=en-VN> [Accessed: 2021-04-15].