Embeded systems

Hamish Sams

November 3, 2017

Abstract

This report investigates the use of LabVIEW for programming embedded systems.

1 Introduction

This lab is based around the use of LabVIEW, a graphical programming language used throughout the industry to quickly program complex tasks that would otherwise take days to plan and write in languages such as C, albeit less efficiently, yet still efficient enough to power systems including CERN. In this case we will be programming using the NI hardware I/O interface the Elvis board and the myRIO FPGA board. concise introduction

1.1 Aims

The main aim of this lab is to learn how to use LabVIEW in the programming of prototype embedded systems with input and output systems. We also would like to get used to creating and designing hardware circuits from specifications or circuit diagrams.

1.2 Objectives

1. Program a working input and output system using the NI Elvis board to interface with a servo using LabVIEW.
2.Create a working program that uses the NI myRIO FPGA board to interface outputs with an external breadboard circuit.

2 Background

LabVIEW is a visual programming language using a graphical programming language called G. It is most commonly used for specialized data acquisition and processing for time Dependant systems.Data acquisition is the process of process of requesting inputs from an external source (not hard coded in) followed by processing this information to change how the circuit acts depending on the input. Every interactive circuit in the world uses this idea from a thermostat to a complex computer.Usually these circuits then use an output system whether this be LEDs for a monitor or a coil to turn up the heating.The idea of hardware interfacing is a little strange as it's defined as joining hardware to software but generally software (at least in our case) is just an arranged FPGA and therefore technically hardware but we can still say we are interfacing our hardware with our software and this conjoined interface is called an embedded system where some program is embedded in the control hardware such as a thermistor for our example heating solution in this lab report we are using the myRIO as our programmable system.

3 Theory

A myRIO uses an FPGA to store code, this is critical to certain systems as an FPGA is the best way to get a real time system meaning if something is time critical such as a cutoff safety button it must react as soon as the button is pressed whereas if you used something such as an Arduino and a the code was stuck in a loop you could not turn off they system. The rest of the board is simply used to interface this FPGA with the analogue pins on the side (and a few on-board

inputs/outputs such as LEDs and an accelerometer). Each pin on the myDAQ that is used for an input takes the true analogue value and outputs it into the system as a binary string of booleans where the only differences are for the audio pins where capacitors are used to block DC and to create a bandpass to block inputs outside of the audible frequency.[1] The same system is used for the inputs except instead logically a digital to analogue converter(DAC) and no bandpass on the audio output (but still a capacitor to block the DC component).Now using these inputs and outputs with a hardware interface there is the ability to take data and output more data in real time which is very useful in everyday life in almost everything you use today from your TV to your fridge to control temperature (general control systems) or display an image.

4 Method and Results

The lab talked through the programming of LabVIEW using modules to connect to NI Elvis and NI myRIO hardware to our program which allows for specialized and completely interchangeable systems where you can choose your inputs and outputs. As the programming language is 100% visual it is very easy to trace back and understand what the program is doing and therefore is easy to pickup, read and improve code that wasn't written by you.The code is programmed to the myDAQ by USB port meaning no strange connections and is easily done and after the program is uploaded the system can be powered anywhere and have the same functionality whereas the Elvis board the program is run on the PC and data sent and received via the Elvis-PC USB connection. As all 3 systems (Elvis, myDAQ and LabVIEW) are all made by NI(National instruments) it is as simple as plug in and program(assuming you have basic hardware and software knowledge).

5 Discussion

5.1 NI Elvis II+

The Elvis board is generally used for educational purposes for lots of general purpose data acquisition systems as it has on-board oscilloscopes, power supplies etc but it also has analogue outputs and a switchable PCB slot so you can design or use any system available. This data acquisition system was used to get an analogue value from a potentiometer which would then be sent to the program we design to send an analogue output to a servo. In this case our data acquisition and data output are from the NI Elvis board but the processing of data is done by the PC processor, this is good for systems like this where we are in a lab testing inputs and outputs as we can get the hardware working without having to pay for a nice fast processor when you already have paid for one in your PC it also means that they system shouldn't start to become obsolete as only the PC really needs to be updated.As the system breaks up inputs control and outputs is can be easy to debug where the error is especially with the on-board oscilloscope so you can say whether your program is malfunctioning, your input (potentiometer) system has broken or if your output(servo) is refusing to work. While this system is cheaper and faster than a FPGA for every solution this wouldn't be useful for something such as a washing machine where you would need to have a PC connected and so a standalone system such as the myDAQ would be more appropriate (or even just an Arduino as the system could really be classed as non critical as if there is an error you can just turn the switch off).[2]

5.2 NI myDAQ

The myDAQ system is used throughout industry for time dependent systems (systems where the time something occurs cannot vary) generally in safety critical situations such as a controller for a nuclear reactor if a measurement isn't taken when it should be the system has no idea what the inputs should be (control rod insertion) and could lead to a meltdown situation. The myDAQ does everything the system needs in one package, getting the input values, the logic on these inputs and then the output values, the only thing needed is the hardware to interface these inputs and outputs into your system. The myDAQ is very useful as you can program the system anywhere and then plug it in somewhere else and it would act exactly as it did at the point of programming due to the FPGA(Field Programmable Gate Array) storing this data this makes it good for medium sized systems that are specialized so updates can be done on the system easily by reprogramming on site or by sending off the module to the programmers however with smaller systems such as

a fridge that takes an input and outputs a value the company would not like to re-program each time by coming in to every household (nor do such appliances need updating) instead you would just buy a new one and thankfully you don't need to add £1000 for a system such as the myDAQ. Which brings us onto the next point, price, for a system that is a one-off meaning it needs to be programmed for one person/company the expensive part is the software not the hardware even when these boards are in 4 figures as the hardware isn't made for one customer it is spread out across thousands of purchases and so the fee of £1000 can be negligible for certain specialized systems and a much cheaper solution than completely re-designing hardware for a function. So overall the hardware is rather cheep for medium/large sized systems but for a small scale non time dependent system such as everyday appliances you are far better off using a Arduino style device.

5.3 Other systems

Generally if a company needs a time dependent system building and don't use a DAQ it would be built using pure hardware meaning the system physically could not lag (using digital logic usually). Otherwise if a system isn't time dependent you could create the circuit using hardware but it is much easier and generally cheaper to program a circuit using a micro-controller like a PIC chip or even a micro-controller which has already had the interfacing designed for you such as an Arduino or Raspberry Pi (around £20-£40 each for the branded or personally i use the £1.36 Chinese knockoff).

6 Conclusion

In terms of aims and objectives in both systems the programs were made as instructed and worked perfectly and on top of that functionality was added on top and so the experiment met and went beyond the objectives set. As these aims required the knowledge of how to use LabVIEW the aims of the lab were also met and therefore a success. With more time and access to the resources of the lab these programs could be enhanced to either work more efficiently or have added functionality especially with the myRIO as of the freedom with the outputs you could wire up almost any module as long as the RIO could supply enough current/voltage even then you could use relays to control high power systems. The main conclusion from this is that for time dependent systems in one-off or expensive systems the myDAQ is probably the fastest way to get complex functionality and generally a cheap solution in terms of design and creation. Whereas for systems that are non time dependent/safety critical it's cheaper (and more efficient) to use smaller commercially available platforms such as an Arduino.

7 Appendices

References

[1] N. Instruments, "myrio specifications sheet." Manual, May 2016.

[2] D. M. Foster, "Labview and embeded systems lab sheets."