Name of Scientific area: CONTROL AND AUTOMATION ENGINEERING

CONTROL A PNEUMATIC SEQUENCE USING WATERFALL METHOD IN THE LABVIEW PLATFORM

José Domingos Senra Simões, A65378@alunos.uminho.pt, Mechanical Eng. Student

Andre Monteiro Fernandes, A65381@alunos.uminho.pt, Mechanical Eng. Student

Eurico Augusto Rodrigues de Seabra, eseabra@dem.uminho.pt, Mechanical Eng. Professor

ABSTRACT

The project chosen for this paper was one that the student had to construct a program using the LabVIEW platform in which it would control electro pneumatic systems using the waterfall method. As a result a program was created in LabVIEW that would do just that with various other functionalities like an interactive GUI (guided user interface) which is easy to follow and use along with microcontroller communication and programming capabilities. This added functionality allows the user to export the pneumatic code to a microcontroller like the NI USB 6009 (used in this project) and test or execute the code using pneumatic cylinders and end position sensors.

KEYWORDS

LabView, NI Controller, Waterfall Method, Pneumatic Sequence, NI USB 6009

1 Introduction

In this project the student had to create a program in LabVIEW that would control a pneumatic system using the waterfall method. As with every project, students must first research every topic associated with the project as well as find related products or services already on the market that have to do with the project. In addition, any software or equipment needed to complete the project must be adequately learned and used in a way for the student to be well familiarized with it.

The developed program consists of two main parts which include the input of data (pneumatic sequence) with data processing and the output, sending/receiving and processing of data to/from the microcontroller. Throughout the creation of the program various problems were encountered and various improvement ideas were suggested because like always, there is constantly room for improvement.

2 Contextualization

2.1 Automation Systems

Automation systems (Groover & P., 2007) consists of two main parts which include the controller and the hardware or physical parts (Cooper & D., 1996). While the controller is used to process data or just execute commands depending on the type of system, the hardware is used as an output executing commands from the controller or as an input giving the controller data to process, or both. (Machado)

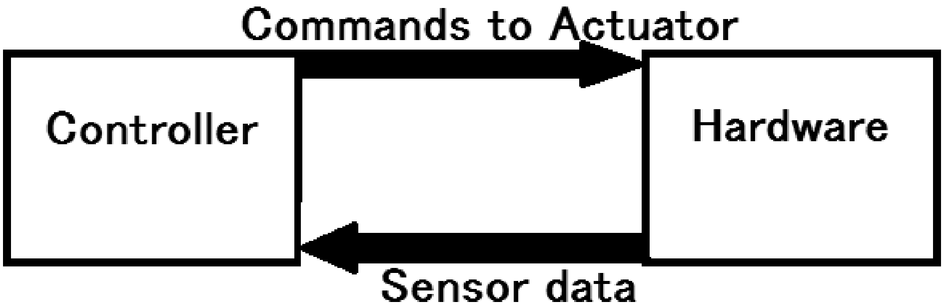


Figure 1- Controller and Hardware Communication (Machado)

These systems are usually controlled or moved by various means including mechanical, hydraulic, pneumatic, electrical, electronics and computers, usually in combination. Complicated systems, such as modern factories, airplanes and ships typically use all these combined techniques. These systems started gaining popularity for various reasons including but not limited to: Easy to control, higher production rate, producer higher paying jobs for maintenance workers, etc. (Machado)In this work, the automation system (Sheridan, 2002; John & Tiegelkamp, 2010) included a microcontroller (NI 6009), computer as the data or command input, electro pneumatic cylinders as the actuators and the cylinder end stops as the sensors that tell the microcontroller that the cylinder actuators have reached the destination.

2.2 Waterfall Method

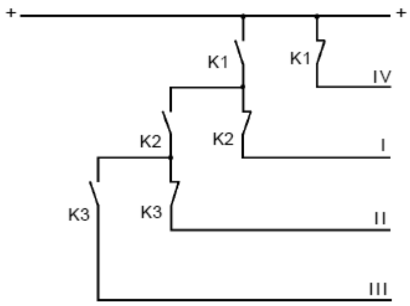


Figure 2- Waterfall Method Diagram Representation (Jacareí, 2001)

The empirical method of solving sequences acquires command equations for any determined sequence which the user decides to implement, it utilizes the end stop sensors associated to each cylinder of the current cylinder movement as a way of starting the next movement. After each command is sent to activate or deactivate a cylinder the program waits for the end stop sensor to acknowledge that the cylinder has reached its destination, only then will the next command be executed. This is an effective method when the sequence of commands for the forward movement of the cylinders is the exact same as the reverse movement. (Jacareí, 2001)

Since this is a very scarce situation to come across, there will obviously be a handicap situation in the circuit which will lead to deadlock of the whole system. Therefor the waterfall method was established, through a determined number of 4/2 valves, which prevents that a system with any kind of sequence stops working because the forward movement are not in tune with the reverse movements. (Jacareí, 2001)

- The waterfall method consists of the following steps:
- Establish a sequence of cylinder movements
- Divide sequence in groups in which cylinders must not be repeated in the same group
- Establish matrix with the necessary conditions of each command
- Determine the selector
- Identification of the selector signs
- Analytical identification of the selector pilot signs
- Analytical identification of command movements
- Pneumatic and electric circuit modeling

2.3 Types of cylinder Actions

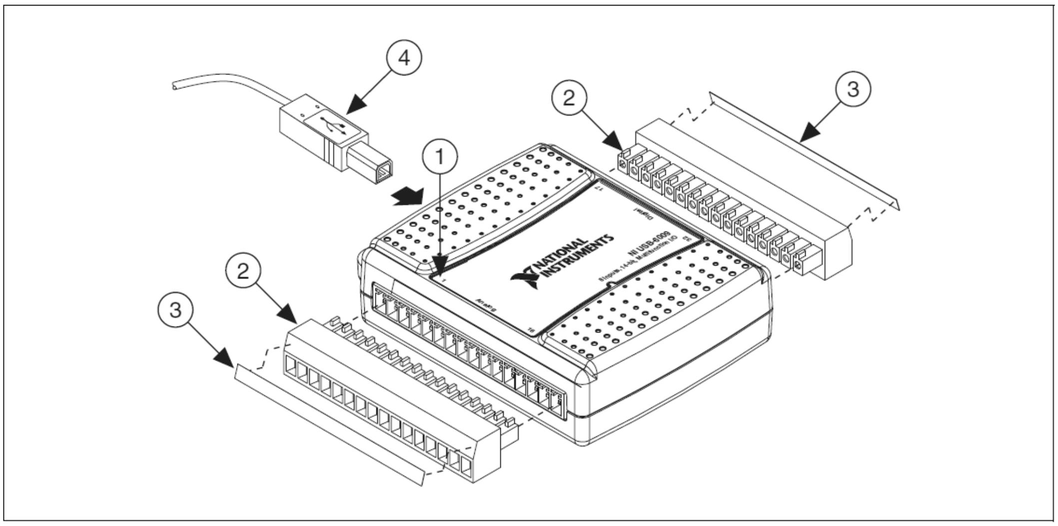
When controlling pneumatic cylinders (United States Patent No. US4000718 A, 1977) and creating the electro pneumatic system, the user is required to know if the cylinders are of single or double action. A single acting cylinder consists of a cylinder casing with a piston or pushrod which has fluid pushing on only one side. When the fluid stops exerting a force on that side there is a mechanical device, such as a spring, that returns the piston to the “home” position. (Huber, Fleck, & Ashby, 1997)

A double acting cylinder, similar to the single acting, which includes a cylinder casing and a pushrod also consists of a fluid pushing on both sides. (Love, 2007) To extend the cylinder, fluid must flow and push on one side of the piston. When the fluid stops exerting a force on the piston, the piston does not return to the “home” position. For the piston to retract fluid must flow and exert a force on the other side of the piston. (Huber, Fleck, & Ashby, 1997)

2.4 Microcontroller Board

In this project the NI 6009 microcontroller board was used to control pneumatic cylinders and acquire data from the end position sensors.

The board, pictured below, utilizes various components to operate such as the screw terminal connector pins to connect all the wires of the actuators and sensors, signal labels to label each pin and a USB cable to allow communication with the computer. (Instruments, USER GUIDE AND SPECIFICATIONS NI USB-6008/6009)



1	Overlay Label with Pin Orientation Guides	3	Signal Label
2	Screw Terminal Connector Plug	4	USB Cable

Figure 3- NI USB 6009 Operating components (Instruments, USER GUIDE AND SPECIFICATIONS NI USB-6008/6009)

The board consists of the following specifications: (Instruments, USER GUIDE AND SPECIFICATIONS NI USB-6008/6009)

- 8 Analog inputs (12-bit, 10 kS/s);
- 2 Analog outputs (12-bit, 150 S/s);
- 12 Digital I/O;
- USB connection without need for external battery;
- NI-DAQmx driver software;
- LabVIEW compatible

3 Project Development

The following will explain how the program in LabVIEW was made as well as the path taken to allow communication between LabVIEW and the microcontroller. (Li, Liu, & Sun, 2007) Also, all the user interfaces will be shown and explained as the user would encounter them when executing the program.

3.1 Program Structure

The developed program contains two main parts, these are, data acquisition from the user and data analysis and processing from user's input along with communication to/from the microcontroller. The image below shows the first thing the user sees when running the program. There are two tab panes, one pertaining to the user input and the second that has to do with the controller communication. Below the user input pane is a section that displays the cylinder sequence from the user inputs. The top one displays the cylinder's initial position, underneath is the pneumatic sequence the user entered. This is updated as the user enters the data. After the data input, the sequences are treated as dictated by the waterfall method. The sequence must be divided into groups to make the sequence more efficient. This group data is then displayed to the left of the submitted sequences indicator.

Below is a flow chart showing the program execution process. All the data convergences are AND convergences, meaning that the program needs all user inputs to continue.

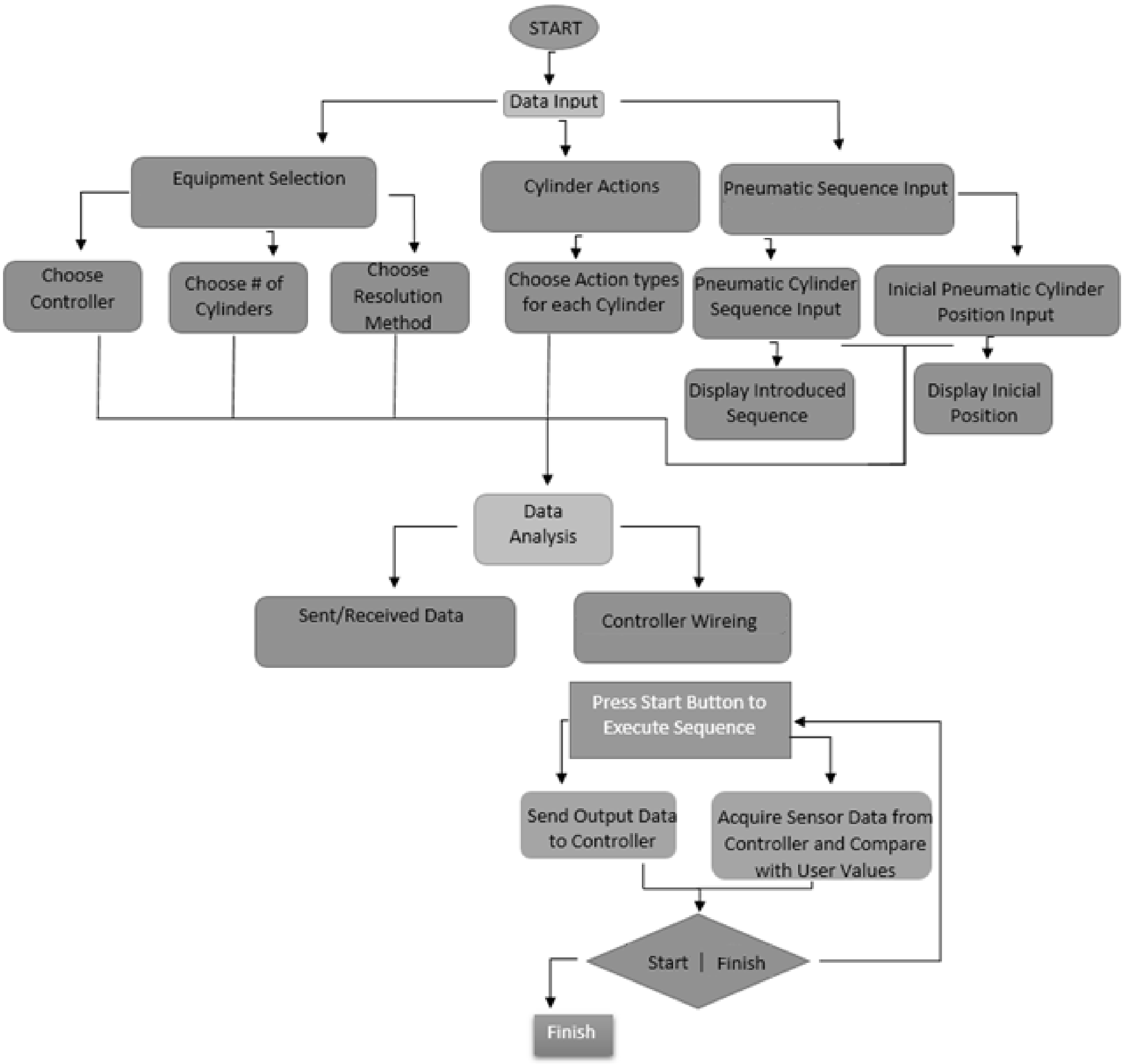


Figure 4-Program flow chart

3.2 User Data Input

When the user first executes the program, a dialog appears telling the user what has to be filled out first in the program. These first data inputs include the number of pneumatic cylinder to be used, the type of microcontroller and the sequence resolution method.

The first tab the user is confronted with is the the equipment selection tab. Here the user picks the controller that will be used, currently there is only one programmed board which is the NI USB 6009 from National Instruments. Next depending on the number of inputs and outputs of the controller, the maximum number of cylinders is limited. After choosing the number of cylinders to use, the user must select the resolution method of the input sequences. For the case of this project, the method used is the waterfall method, but more methods could also be added in the future as updates to the program.

After clicking the "OK" button, a dialog box appears and the tab is automatically changed to the next tab, cylinder actions. The dialog box informs the user to input the known cylinder action(s). In the tab, the user can choose the type of cylinder action for each cylinder. The cylinder can either be a single action or a double action, but by default the single action is chosen for every cylinder. The user can only change the cylinder action for the number of cylinders chosen in the previous tab, the rest will be grayed out and disabled. The program needs to know this information so that it can generate the correct code for each cylinder type.

After choosing the cylinder actions for all the desired cylinders, the user must press the "OK" button. By doing this, a dialog box appears and the user is automatically taken to the next tab sequence input. The dialog text informs the user to enter each cylinder's initial position, this position will be consider each cylinder's "home" position and can be different for every cylinder. By default, the initial position for each cylinder is retracted. For example, if we look at cylinder A, "A+" is extended and "A-" is retracted, there is also a third option that is to keep the current position, but this option is disabled and grayed out since every cylinder must have a defined initial position.

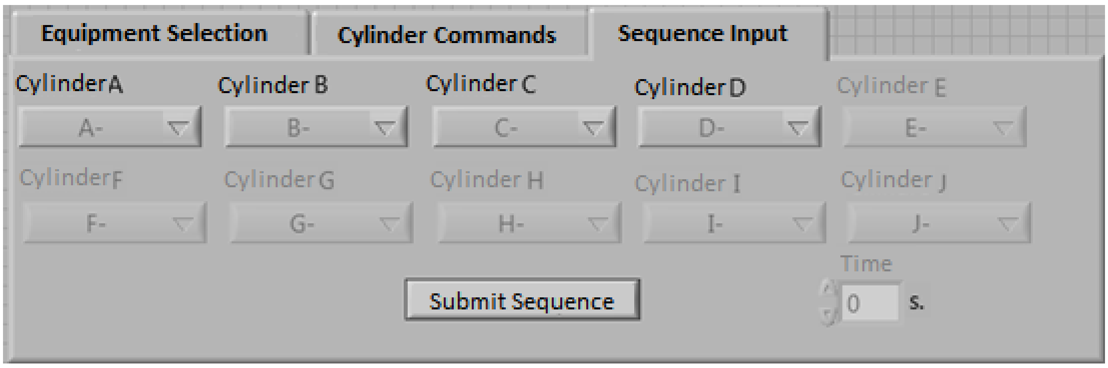


Figure 5- Cylinder's initial position input

After indicating the desired initial position for each cylinder, the user must click the submit sequence button. After submittal of the initial position sequence, the exact sequence is shown in an indicator box for the user's future reference.

Once the initial cylinder position is submitted, the program will enter a loop where the dialog box below appears and asks if more cylinder sequences should be added.

If another sequence should be added, then the user must enter the cylinder position for each cylinder and click submit sequence. By default, the selected input is to keep the current position. For every cylinder there is an option that is disabled and grayed out, this option corresponds to the last command sent to the cylinder. For example, if A+ was previously selected, the user is only able to select A- or keep position. Since cylinder is already A+ there is no reason to resend the command A+. This is a way of eliminating this type of user input error. If the user needs a wait time after a sequence is executed, this wait time in seconds should be inserted in the time input shown in the image below, and by default this time is 0 seconds.

While the user enters the sequences, they appear after each submittal in the text indicator box. This box is used for user reference to the submitted sequences and wait times and will later indicate to the user which sequence will be in execution when sending and receiving information from the controller. This will be done using the LEDs to the left.

If the user tries to submit a sequence where all cylinders keep their position, a message will appear and warn the user of the input error. The user is then asked to retry the sequence input, in every sequence at least one cylinder must change its position.

When the user finishes entering all the sequences, the program will check for any user input errors. For example, if the user indicated to use more cylinders than needed or the user does not use all the cylinders provided, a dialog box will appear to inform the user of this error and continue to ask if more sequences should be added.

All cylinders must also end in the “home” position, if this rule is not followed a dialog box will appear. The user will have the choice to either let the program correct this error automatically which would be the yes button, or to add another sequence to correct this error, no button.

If the user chooses to fix this error automatically, the program will send the “home” command to all the cylinders not already there and send the keep position command to the cylinders already in the home position.

Once all sequences are added and all pass the validation rules, the sequence is then divided into groups pertaining to the waterfall method. These groups once divided are shown for user reference in the text indicator box.

3.3 Sensor Data Acquisition and Data Processing

Once all data sequences are entered and validated, the programs opens a dialog box informing the user to wire the controller as pictured in the diagram. The user should only press the “OK” button once all wiring is finished.

In this case, only the NI USB 6009 micro-controller is programmed and configured to be used, but more boards can later be added to the program. As a way of making the controller wiring easier to the user and reducing the chance of possible connection error due to changing cylinder actions from one circuit to another, the controller schematic was created with each cylinder having the same connections and every pin is fixed to a certain cylinder. To create the schematic, a maximum number of cylinders must be known in order to limit them. This quantity can be found out by looking at the controller’s number of input and output pins and relating to the maximum cylinders being able to control along with being able to acquire all sensor signals. As a simplification, all cylinders must be considered as being of double action so that every cylinder has two pins assigned to it from the controller. Also all cylinders need two end position sensors in which each need one input to the controller (two inputs per cylinder). So in the case of the USB NI 6009, there are 8 analog inputs, 2 analog outputs and 12 digital pins which can each be configures as inputs or outputs. Say the user needs 5 cylinders, the 5 cylinders need 2 output pins each making it 10 output pins and 10 input pins are also needed for all end position sensors from all cylinders. This means that the controller USB NI 6009 is only able to control 5 cylinders with 2 digital pins to spare. One of the digital pins will be used to connect the start button while the other could in the future be connected to a physical emergency stop button.

The start button should be a normally closed button connected to GND or ground, all digital inputs on the controller board are normally high, so if a digital input pin is ever not needed, for example, if the user only utilizes two cylinders (A and B) the other input sensor pins must be connected to GND. A single action cylinder should only connect to the cylinder+ pin while a double action cylinder should connect to both the cylinder+ and cylinder- pins.

Once the user clicks he “OK” button on the dialog box to wire the controller, the user is automatically taken to the tab “Dados Enviados/Recebidos” or sent/received data. Here the user is able to visualize graphically what cylinders are being activated and view what sensors inputs are supposed to indicate and the current sensor values shown in real time. The user has an indicator LED showing the start button status, while the start LED is off the program will not start. After starting, the start button is not needed unless the user would like to execute the sequence in a loop. The user also has two buttons, one in the top right corner which is used to terminate the program, by clicking this, the program terminates but only after executing all sequences. The second button, bottom center, is the emergency stop button. If at any point in the program this button is pressed, the program terminates immediately.

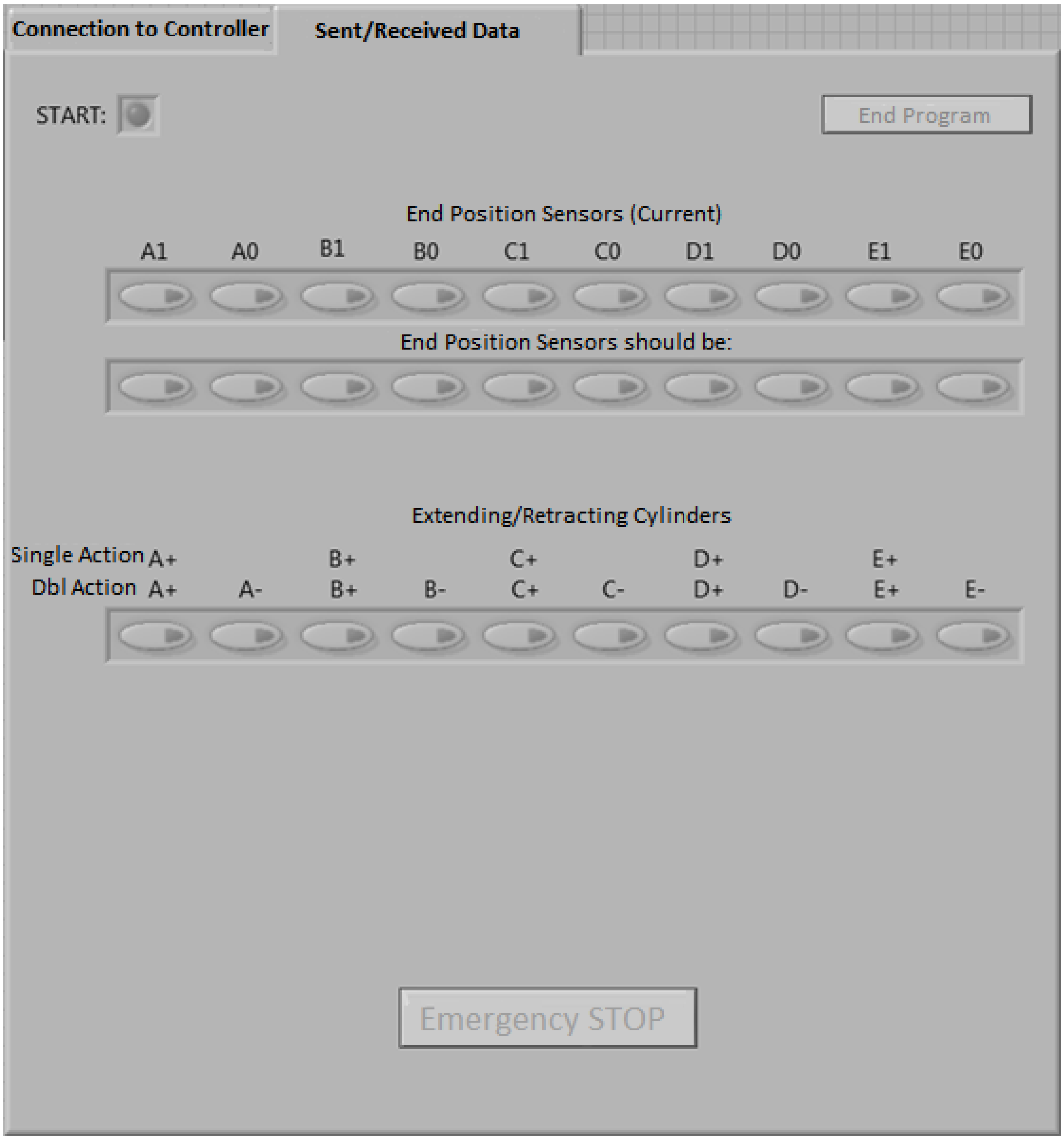


Figure 6-Sent and received data to/from controller

While the sequences are executing, the indicator LEDs next to the each sequence indicator and group indicator should light up when that sequence or group is being executed. Below is an image of these LEDs.

3.4 Encountered Problems

Throughout the project development, several problems were encountered that reduced the project's progress. The two main problems encountered were the lack of knowledge and experience with respect to the LabVIEW platform and towards the controller communication code. After a few tutorials, example code completions and a few teacher questions and tips, program and controller familiarization was quickly increased and the project growth continued to expand seamlessly. (Instruments, Labview TM User Manual, 2003; Instruments, USER GUIDE AND SPECIFICATIONS NI USB-6008/6009; Aprenda Sobre O LabVIEW: Introdução à Programação Gráfica No LabVIEW)

3.5 Possible Improvements

Like every work, there is always room for improvements. Various possible improvements for this work include being able to save and import a pneumatic sequence into the program and being able to edit the entered pneumatic sequence. Other improvements could also include, the insertion of a physical stop button that could stop the program at any time, the addition of more microcontroller options like other NI microcontrollers and Arduino base microcontrollers. The addition on other methods of resolution of the sequence like SFC method and the addition of a higher quantity of cylinders for the user to choose from when using a controller with many input and output pins. The last suggested improvement is to add the ability for the user to input the pneumatic sequence into the program and control pneumatic controllers over the network while watching everything happen over a camera filming the pneumatic cylinders and broadcasting over the network. (Kumar, Singh, & Kulkarni, 2014) (Cooper & D., 1996) (Takamiya, et al., 2008)

4 Conclusion

The assigned project was most constructive, all the objectives and tasks were accomplished as previously stated. The end result is a program constructed in the LabVIEW platform that allows the user to input electro-pneumatic sequences to control single or double action cylinders through the NI USB 6009 Microcontroller from National Instruments. The user input electro-pneumatic sequences are processed and manipulated using the waterfall method which allows for a safer and more compact code and execution. As with all projects, there is always room for improvement and numerous improvement ideas have been suggested for future implementation.

Being such a successful project, it is an amazing experience to construct a useful program that works and is able to help future mechanical engineering students understand how to command and operate electro-pneumatic cylinders (Zhang, Jiang, Wang, & Peng, 2002) as well as help investigators complete automation circuits using electro-pneumatic cylinders (Dedrick & Erie, 1978). The completion of this project also helped the understanding and comprehension of how block programming works using the LabVIEW platform and how controller communication is accomplished using the NI USB 6009 microcontroller from National Instruments.

5 References

Aprenda Sobre O LabVIEW: Introdução à Programação Gráfica No LabVIEW. (n.d.). *Introdução Ao LabVIEW.*

Brown, M. (1977, January 4). *United States Patent No. US4000718 A.*

Cooper, & D., M. (1996). *Design of Library Automation Systems: File Structures, Data Structures, and Tools.* New York, NY, USA: John Wiley & Sons, Inc.

Dedrick, A. R., & Erie, L. J. (1978). Automation of On-Farm Irrigation Turnouts Utilizing Jack-Gates. *Transactions of the ASAE, 21*(1), 92-96.

Groover, & P., M. (2007). *Automation, Production Systems, and Computer-Integrated Manufacturing* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.

Huber, J. E., Fleck, N. A., & Ashby, M. F. (1997, October 8). The selection of mechanical actuators based on performance indices. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 453*(1965), 2185-2205.

Instruments, N. (2003). Labview TM User Manual. National Instruments.

Instruments, N. (n.d.). USER GUIDE AND SPECIFICATIONS NI USB-6008/6009. National Instruments.

Jacareí, S. P. (2001). Tecnologia Eletropneumática Industrial. Brasil: Parker Hannifin Ind. Com. Ltda.

John, K.-H., & Tiegelkamp, M. (2010). *IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids.* Springer Science & Business Media.

Kumar, N., Singh, N., & Kulkarni, A. (2014). LabVIEW based online monitoring and control of PMDC motor speed. *2014 6th IEEE Power India International Conference (PIICON).*

Li, L., Liu, H., & Sun, Y. (2007). *A Method for the Simulation of Hydraulic System Controlled by PLC Based on LabVIEW and FluidSIM.* Shenyang, China: Northeastern University. Retrieved from http://en.cnki.com.cn/Article_en/CJFDTOTAL-JCYY200710061.htm

Love, J. (2007). *Process Automation Handbook: A Guide to Theory and Practice.* Springer Science & Business Media.

Machado, J. (n.d.). AUTOMAÇÃO DE SISTEMAS A EVENTOS DISCRETOS: CONSIDERAÇÕES GERAIS. Guimaraes, Portugal: University of Minho.

Sheridan, T. B. (2002). *Humans and Automation: System Design and Research Issues.* New York, NY, USA: John Wiley & Sons, Inc.

Takamiya, K., Okumura, R., Abe, N., Nakano, Y., Miyata, K., Fukutani, S., . . . Yamana, H. (2008, November 28). Development of a new control system for pneumatic transportation facility in KUR. *Journal of Radioanalytical and Nuclear Chemistry, 278*(3), 719-721.

Zhang, B., Jiang, Z., Wang, T., & Peng, G. (2002). *Application of Virtual Instruments in Testing of Pneumatic Components and Systems.* Beijing, China: Beijing Institute of Technology.