<div align="center">

Labview Assignment #4, due before November 12th at 3:00pm.

</div>

Note: There are a lot of small blocks in the LABView code I give you below for Exercise 1 and 2. I have found if you open the Word document, you can zoom into the code pictures a bit better than the PDF file.

Exercise 1:

First watch these two Youtube videos about shift registers (or find other tutorials online).

https://www.youtube.com/watch?v=wRvpb9jtRlE

https://www.youtube.com/watch?v=4ivYARn2fy8

For this assignment we are going to start working with the 2D Picture block in LABView. After performing this exercise, you could for example use what you learn to create 2D pictured environment for your robot and draw where the robot is located in this environment. Below is the LABView program I would like you to reproduce for this assignment. It shows you how to draw lines in the 2D Picture. Each time through the while loop it takes a random number between 0 and 1 and draws a 25 pixel by 30 pixel rectangle. (Of course this could have been done easier by using the Draw Rectangle block but I wanted to show how to combine multiple draw commands.) Then a shift register is used to keep track of all the lines printed in the 2D Picture as long as the Boolean LED is true. If you click on the Boolean LED making it false, the lines drawn in the previous loops should not be printed to the 2D Picture.

If your screen is big enough make the 2D Picture 800X800 pixels.

Location of the blocks that may be new to you for this assignment:

From the Front Panel:

Modern->Graph->Controls->2D Picture

Modern->Boolean->Round LED

From the block diagram:

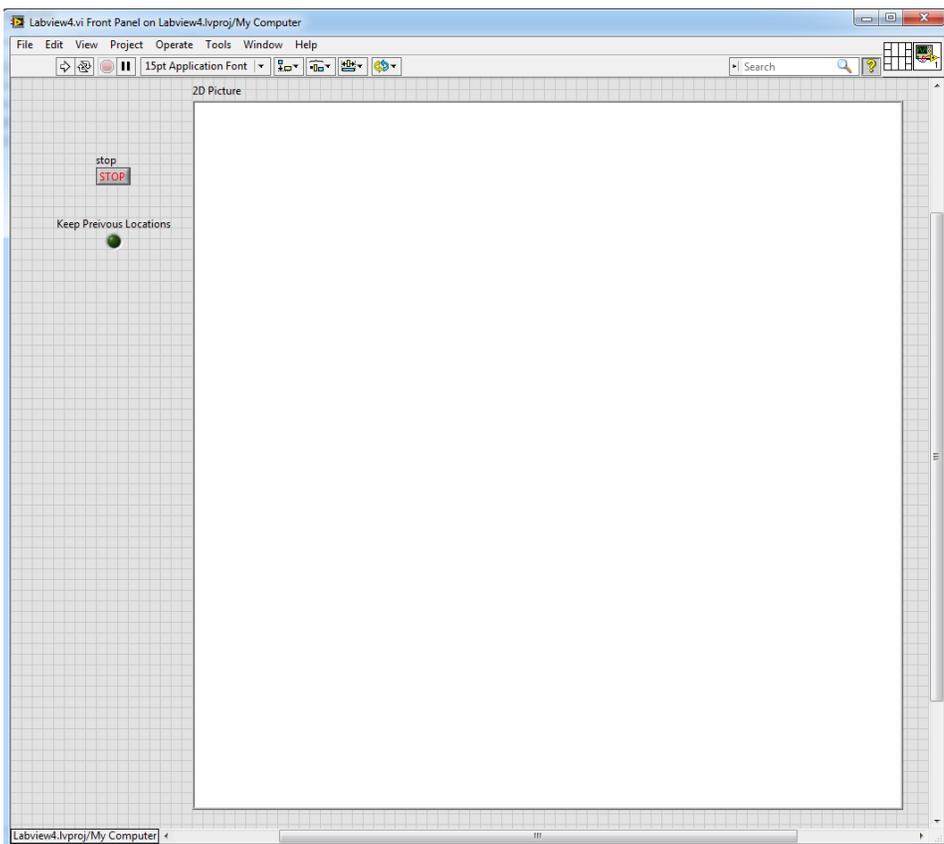Programming->Graphics & Sound->Picture Functions->Move Pen

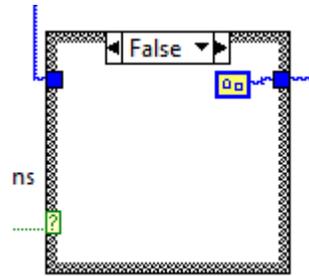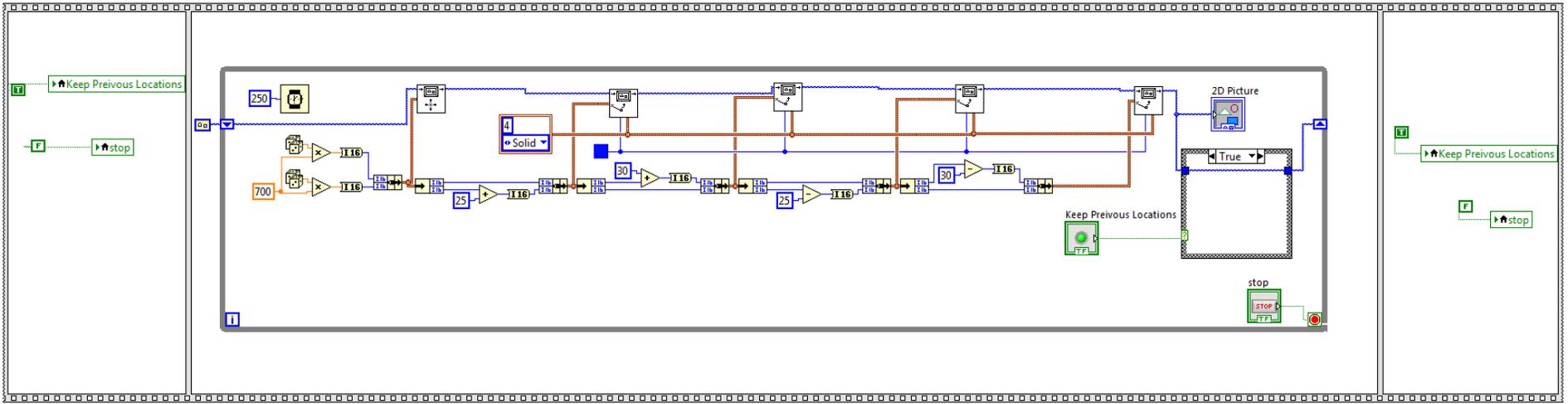Programming->Graphics & Sound->Picture Functions->Draw Line

Programming->Graphics & Sound->Picture Functions->Empty Picture

Programming->Numeric->Random Number (0-1)

Programming->Numeric->Conversion->I16

Programming->Cluster, Class & Variant->Unbundle and Bundle

False Case

Exercise 2:

For this exercise, I would like to give you an introduction to communicating to and from LABView using the TCP Ethernet protocol. The goal of the VI is to take two double precision numbers and combine them into a single string. This string is then sent to an Echo Server with the "TCP Write" command. (Matlab has a nice simple Echo Server that we are going to use.) Then the Echo Server echoes back the same string. We receive this string with the "TCP Read" command and imagine that it is different information then we just sent. We use the "Scan From String" command to pull the two real numbers out of the string and display them.

Copy this VI and demonstrate it working. *The first time you try to write data across a TCP socket from an application the Windows Firewall will pop up with dialogs asking for permissions. Make sure to "check" all boxes in those pop-up windows allowing public and private networks.* Because we are using this echo server, you may find it easier to run this LABView application on the lab PCs instead of your laptop even if you have Matlab installed on your laptop. If things to do work on your laptop, give the lab PCs a try. You will need to find the IP of your computer. Easiest way is to run "ipconfig" from a command prompt. Make sure to find the IP of the main Ethernet connection. On the lab PCs it is usually the first one listed. This is the IPv4 address you will type into the text box entering the "TCP Open Connection" block.

Then before running the LABView application, open Matlab and type the command "echotcpip("on",12321)". 12321 is the port the echo server will be listening to. *Later if you want to turn off the echo server type the command "echotcpip("off")" to turn off the server.* Once the echo server is running in Matlab, you can run your LABView application. Type in some real numbers into the two Send Values and then click "SEND". Play with different numbers and see that you are getting back the same numbers. When you are done, you can click the "STOP" button.

Location of blocks you may have not used for the Block Diagram view:

String->Carriage Return Constant   and   String->Line Feed Constant   and   String->Scan From String
String->Number/String Conversion->Number To Fractional String
Structures->Local Variable
Data Communication->Protocols->TCP->"TCP Open Connection" and "TCP Read" and "TCP Write" and "TCP Close Connection"
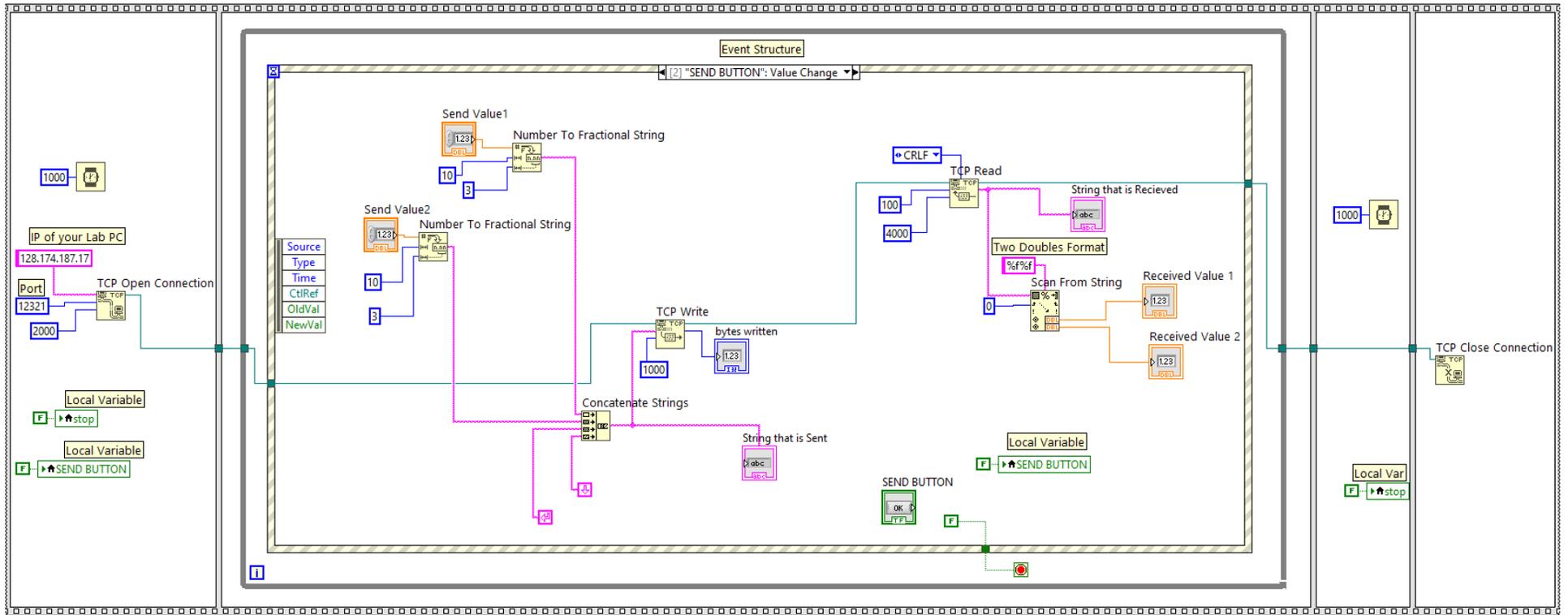
TCP Blocks found under "Data Communication" -> "Protocols" -> "TCP"
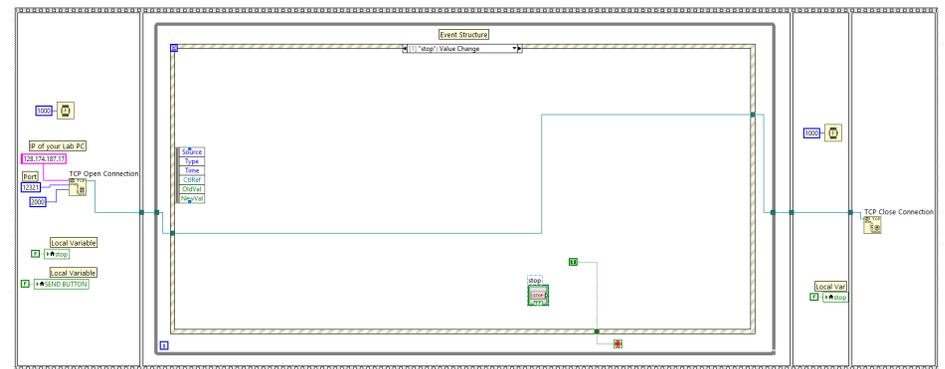
Showing the Timeout Event
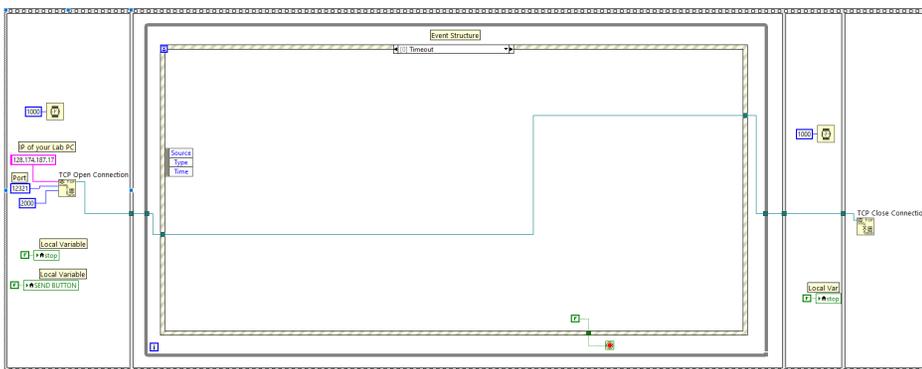
Stop: Value Change Event

TCP Blocks found under "Data Communication" -> "Protocols" -> "TCP"

TCP Blocks found under "Data Communication" -> "Protocols" -> "TCP"