Final Exam: MarioBot 01 May 2015

Joey Weate (jfw6fp)

------------------------------------------------------------------------------------------------------------------------------------------

*Goal:* Implement a new component to the myRio iRobot in order to expand our knowledge of applying something unknown and to create ideas for future myRio classes.

-------------------------------------------------------------------------------------------------------------

*End State:* An iRobot's reaction to sound input was un-debug-able. When the myRio detected a short clap, it would drive forward. When a loud clap was detected, it would stop. This will be discussed in detail in the Audio Input section of the report.

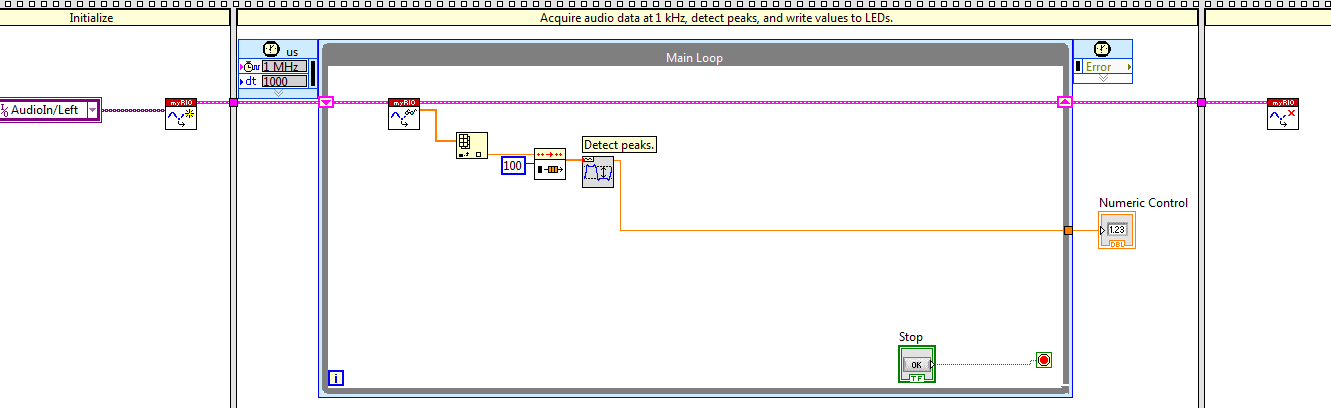
This resulted in switching to an external component to the iRobot- A relay attached to two strands of Christmas tree lights. When the iRobot moved forward, one strand of lights would come on. When the iRobot moved backwards, the second pair of lights would come on.

------------------------------------------------------------------------------------------------------------------------------------------

*Audio Input:*

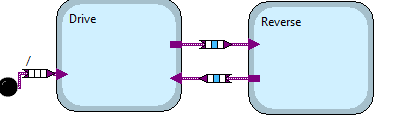
The goal of Audio Input was to utilize a microphone attached to the myRio to determine the state of the iRobot. This would practically serve as a clap-on clap-off car, but could be altered by future labs to input voice audio, applying more intense manipulations of signals to determine a state. For example, the iRobot could *potentially* be voice controlled. This would give students a chance to apply sampling and signal manipulation techniques learned in Signals and Systems.

To take in input, the following VI was built:



This Vi has three main subVis: Open, Read, and Close. The audio in signal is opened in the Initialize stage later to be looked at in the main loop. Within the main loop, 100 audio values are sampled by putting them in an array. The peak of the 100 values is then found and saved as a number. This number serves as a good estimate for the amplitude of the audio signal.

Within the iRobot state chart, the following exists:



In the drive state, the iRobot moves forward and in the Reverse the iRobot moves backwards. The guards between the states is a Numeric Control threshold. For example, if the amplitude of the audio signal is greater than 5, the reverse state is entered. From the reverse state, if the amplitude is greater than 2, but less than 5, the Drive State is entered. This would allow a loud clap to implement the Reverse motion while a soft clap would cause the robot to drive forward. The reason the drive guard is greater than 2 would be so that if no clapping is occurring, the drive state is not automatically entered. If this guard was instead just an amplitude less than 5, the only way to reverse the iRobot would be to clap very fast and loud continuously so the amplitude sampled is always above 5. If you stopped clapping, the input would be around 0, which would cause the drive state to enter. having a guard from 2 to 5 instead gives the user/driver the chance to determine easier what state they would like to be in.

When running both the audio input VI and the iRobot state chart in parallel, an error kept occurring while trying to load the software to the myRio. It was an unknown error with the SmartOpen VI. Upon countless debugging efforts, this error could not be resolved.

When running independent of each other, each part worked. The audio VI was hooked up instead to the LED's on the myRio. Based on the intensity of the claps, the LED's would turn on. If a soft clap occurred, 1 LED would turn on. If a loud clap occurred, 4 LED's would turn on. This proved that our Audio Input, software was working. Trying to debug it further, we altered the iRobot state charge to have guards of LED's being on- since we could properly turn on LED's. If 1 LED turned on, drive forward. If 4 LEDs turned on, reverse. This guard was implemented by comparing an array of the LED status to the desired guard input. However, this resulted in the same end state: Unknown Error, SmartOpen Vi. Deleting piece by piece of both parallel software's, the bug was always in existence.

Within the audio input VI, an open vi is being used. After some research, this is not the same as SmartOpen. We never *purposely* used the SmartOpen vi, nor could we find it as an option to implement. This leads me to believe that either the SmartOpen vi was being used to read the data of the myRio code in parallel with the iRobot navigation state, or that we were supposed to use the SmartOpen vi but for some reason our version of Labview could not access it.

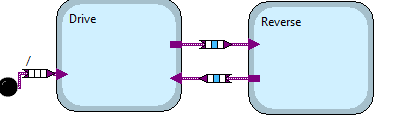
Sampling signals is a huge part of everyday life. Getting hand on experience with these processes are very education and worthwhile for a Computer Engineer. Audio Input to a self-driving vehicle is an essential. What happens if the car is being honked at? (aka cars/drivers are trying to communicate with each other). What should a self-driving car if an ambulance, fire truck, or police vehicle is being it, making a ruckus? Audio Input could better the myRio iRobot safety procedures.

------------------------------------------------------------------------------------------------------------------------------------------

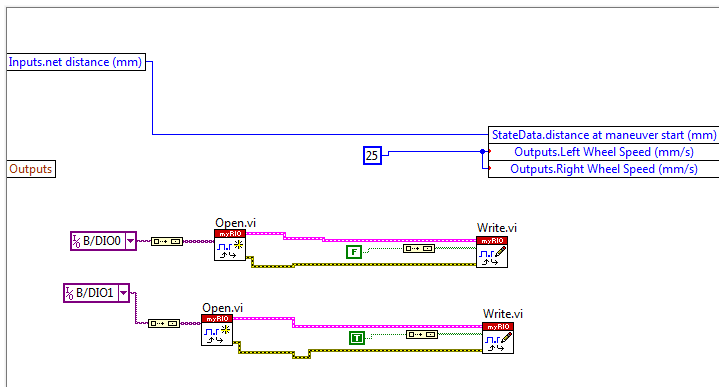
*Relay Lights:*

After failing to debug the Audio-input communications, relay lights were instead attached to the myRio. If the car was going forward, one strand of lights would be turned on. If the car was in reverse, a different strand of lights would turn on. This would serve the purpose for vehicle lights such as reverse lights and could easily be altered to serve as left/right turn signals. This is important in a self-driving car in order to communicate with other cars or drivers what actions the self-driving vehicle intends to do.

The same iRobot navigation state chart was used for this portion, seen below:



What was altered was the guards between the states:



From each transition, which LED was being lit would change. This was done by sending pins on the myRio either high or low. This worked the first time through, turning on a strand of lights on the first state transition, but after that the communication seemed to fail.

What appeared to be the error was that once the data pins were read and written to the first time, they were no longer accessible. So on the first time through the transition, the pins were opened and written to properly, however the output pins/signals were not closed out of the right way. As seen in the previous section, the Close subVI is used at the very end of the Audio Vi when the signals are done being manipulated. Perhaps the pins need to be closed somewhere before the next transition open occurs. More hands on experience with these subVIs would prove to be very useful in debugging.

While the idea of having reverse and turn signals to strengthen communication between iRobots is good, the relay seems to not be a useful method of doing so. The relay is a switch for electronics that requires a lot of power, which consequently turns off lamp, fans, and other potential essential safety critical features.

While both attempts seemed to not produce a run-able iRobot, we learned more about the difficulty and more of what it takes to react to an inputted signal successfully.

-------------------------------------------------------------------------------------------------------------------------------------