AE698A

INTRODUCTION TO VIRTUAL INSTRUMENTATION

TERM PROJECT REPORT

ON

''<u>OBJECT TRACKING AND TRAVERSING USING MOTOR</u>''

Course Instructor

Dr. K. Poddar

SUBMITTED BY-

Samiksha Nagrare (16101038)

Rahul Ranjan (16101034)

Acknowledgment

We are thankful to Dr. Kamal Poddar for explaining us the concepts of virtual instrumentation in a very lucid manner. We are very thankful to the lab assistants, Mr. Nitin and Mr. Rajesh, to make the lab available at almost all the times and helping us in any way possible to help us complete the project. The TA's, without whom the project would never have taken shape as it is now. We thank them for their continuous support in explaining us the details regarding the system and its working.

Abstract

This project deals with object tracking of an object using the image acquisition module in the following three parts. For the first part, integration of two cameras helped us capturing an object in three dimensional space. Position of the chosen object has been captured in multiple frames in such way that its position and velocity could be found in a particular space frame. The second part gave us approximate coordinates to form a image using the frames captured. These coordinates are utilized to move a traverse through a motor, thus forming the third section to fulfil the objective.

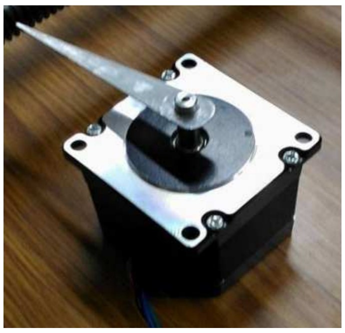
Introduction

Image Acquisition and Processing with LabVIEW combines the general theory of image acquisition and processing. The first step towards any kind of image processing is the acquisition of the actual images to be processed which can be obtained using a camera. It can be chosen between two main types of cameras - analog and digital. National Instruments high-speed image acquisition (IMAQ) devices provide up to 80 MB of onboard memory and they work with motion control and data acquisition hardware.

Some of the instruments used for this project are :

(a) Stepper Motor:

Stepper motors move a known interval for each pulse of power. These pulses of power are provided by a stepper motor driver and is referred to as a step. When 360° of a circle is divided by the step angle gives the number of steps per revolution. Here, the 360° of the circle is divided into 20000 steps i.e. with one input pulse the motor will rotate 0.018°.



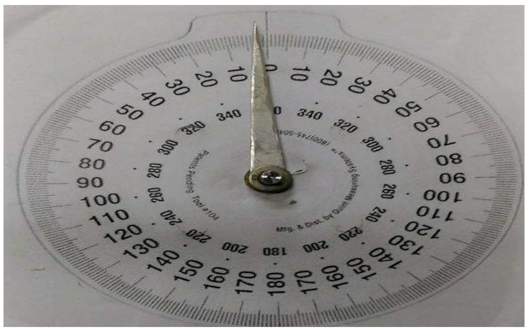


fig. Stepper motor (L) and its needle pointer that rotates (R)

(b) Stepper Motor Driver:

The stepper motor driver used in this program is named as RMCS-1102, developed by Rhino Motion Controls. It designed for smooth and quiet operation without compromising on torque and control at higher speeds. It achieves micro-stepping using a synchronous PWM output drive.

The first 6 terminals of 12 are used for pulse and direction combination and the first 4 out of 8 switches are used for specifying different step resolutions. Tables (1) and (2) shows these terminals and switches respectively.

Terminal no.	Terminal name	Description
Terminal 1	ENA-	Enable(motor free) -ve optically isolated input
Terminal 2	ENA+	Enable(motor free) +ve optically isolated input
Terminal 3	DIR-	Direction -ve optically isolated input
Terminal 4	DIR+	Direction +ve optically isolated input
Terminal 5	PUL-	Pulse -ve optically isolated input
Terminal 6	PUL+	Pulse +ve optically isolated input

table (1): First 6 Terminals

Steps/rev	SW1	SW2	SW3	SW4
200	ON	ON	ON	ON
400	OFF	ON	ON	ON
800	ON	OFF	ON	ON
1000	OFF	OFF	ON	ON
2000	ON	ON	OFF	ON
3200	OFF	ON	OFF	ON
4000	ON	OFF	OFF	ON
8000	OFF	OFF	OFF	ON
1600	ON	ON	ON	OFF
6400	OFF	ON	ON	OFF
10000	ON	OFF	ON	OFF
12000	OFF	OFF	ON	OFF
12500	ON	ON	OFF	OFF
12800	OFF	ON	OFF	OFF
16000	ON	OFF	OFF	OFF
20000	OFF	OFF	OFF	OFF

table (2): First 4 Switches

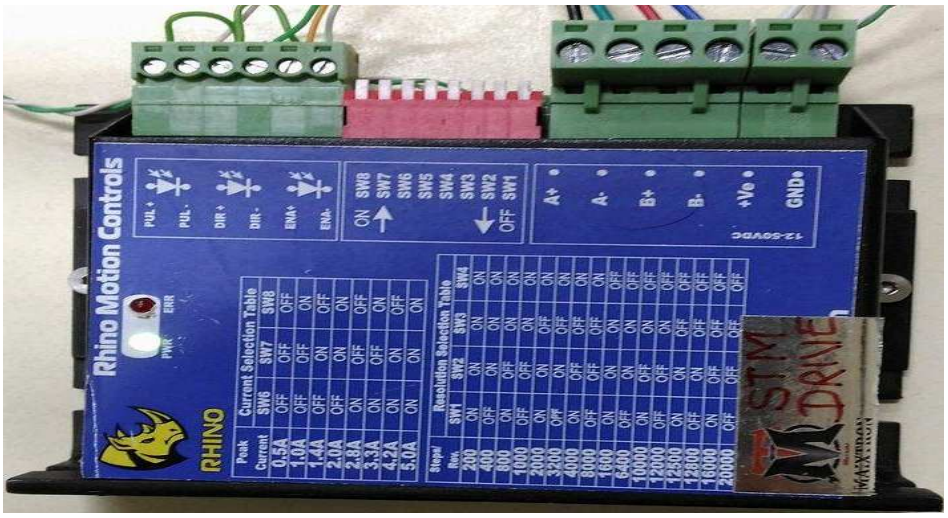
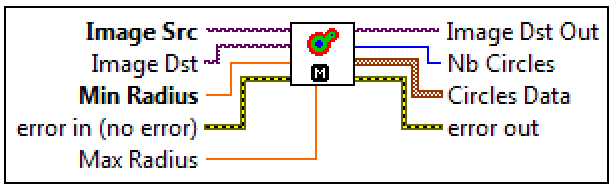
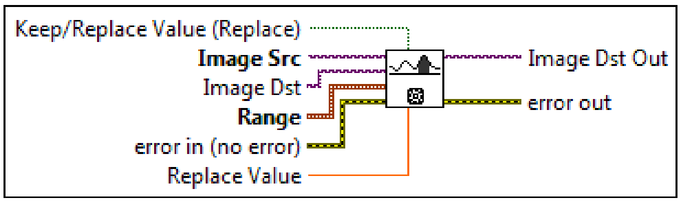
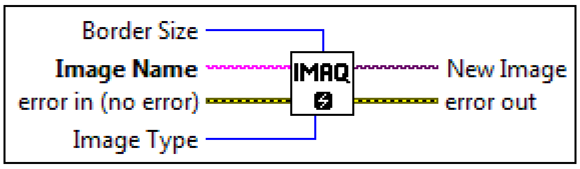
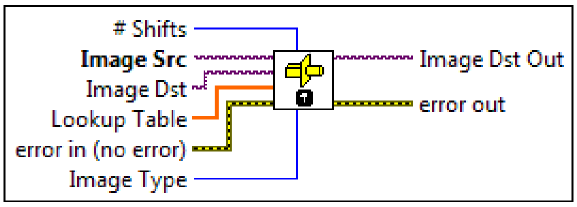
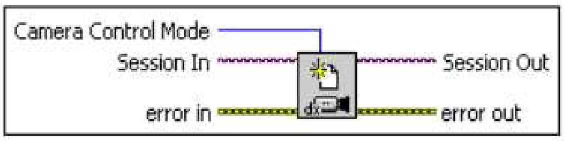
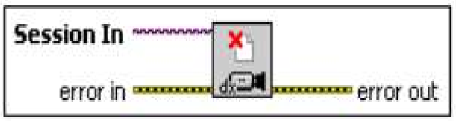
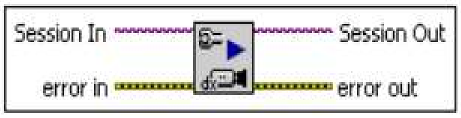
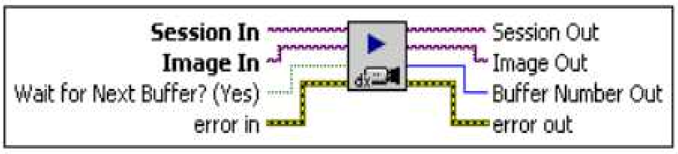
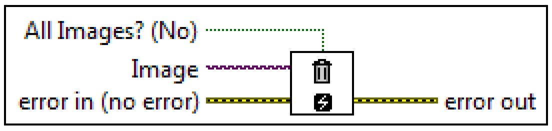


Fig. Stepper Motor Driver

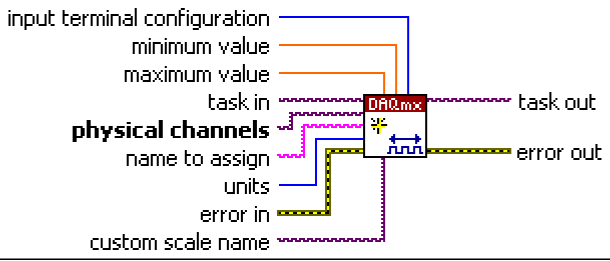
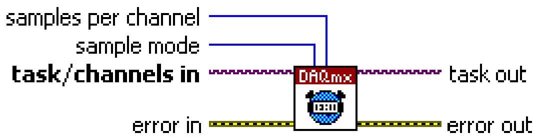
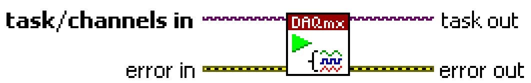
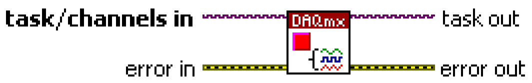
VIs and Functions Used

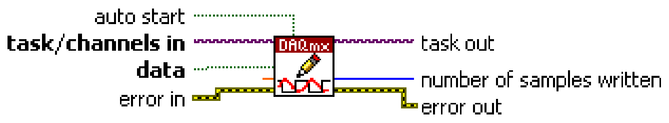
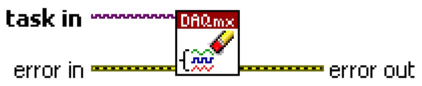
(A) IMAQ functions used:

Sr. No.	**Name**	**Block**	**Function**
1	IMAQ find circles VI	 Image Src Image Dst Min Radius error in (no error) Max Radius Image Dst Out Nb Circles Circles Data error out	Separates overlapping circular particles and classify them based on their surface area, radius and perimeter.
2	IMAQ Threshold VI	 Keep/Replace Value (Replace) Image Src Image Dst Range error in (no error) Replace Value Image Dst Out error out	Applies threshold to a VI. Converts grayscale image to binary image.
3	IMAQ create VI	 Border Size Image Name error in (no error) Image Type IMAQ New Image error out	Creates a temporary memory location for image.
4	IMAQ Cast Image VI	 # Shifts Image Src Image Dst Lookup Table error in (no error) Image Type Image Dst Out error out	Converts the current image type to the image type specified by Image Type.
5	IMAQdx Open Camera VI	 Camera Control Mode Session In error in Session Out error out	Opens a camera, queries the camera for its capabilities, loads a camera configuration file, and creates a unique reference to the camera.
6	IMAQdx Close Camera		Stops an acquisition in progress, releases resources associated with an acquisition, and closes the specified Camera Session.

7	IMAQdx Configure Grab VI		Configures and starts a grab acquisition. Use the grab VI for high-speed image acquisition and to copy an image out of the buffer.
8	IMAQdx Grab VI	 Session In Image In Wait for Next Buffer? (Yes) error in Session Out Image Out Buffer Number Out error out	Configures and starts a grab acquisition. Use the grab VI for high-speed image acquisition and to copy an image out of the buffer.
9	IMAQ Dispose VI	 All Images? (No) Image error in (no error) error out	Destroys an image and frees the space it occupied in memory.

(B) Data Acquisition blocks:

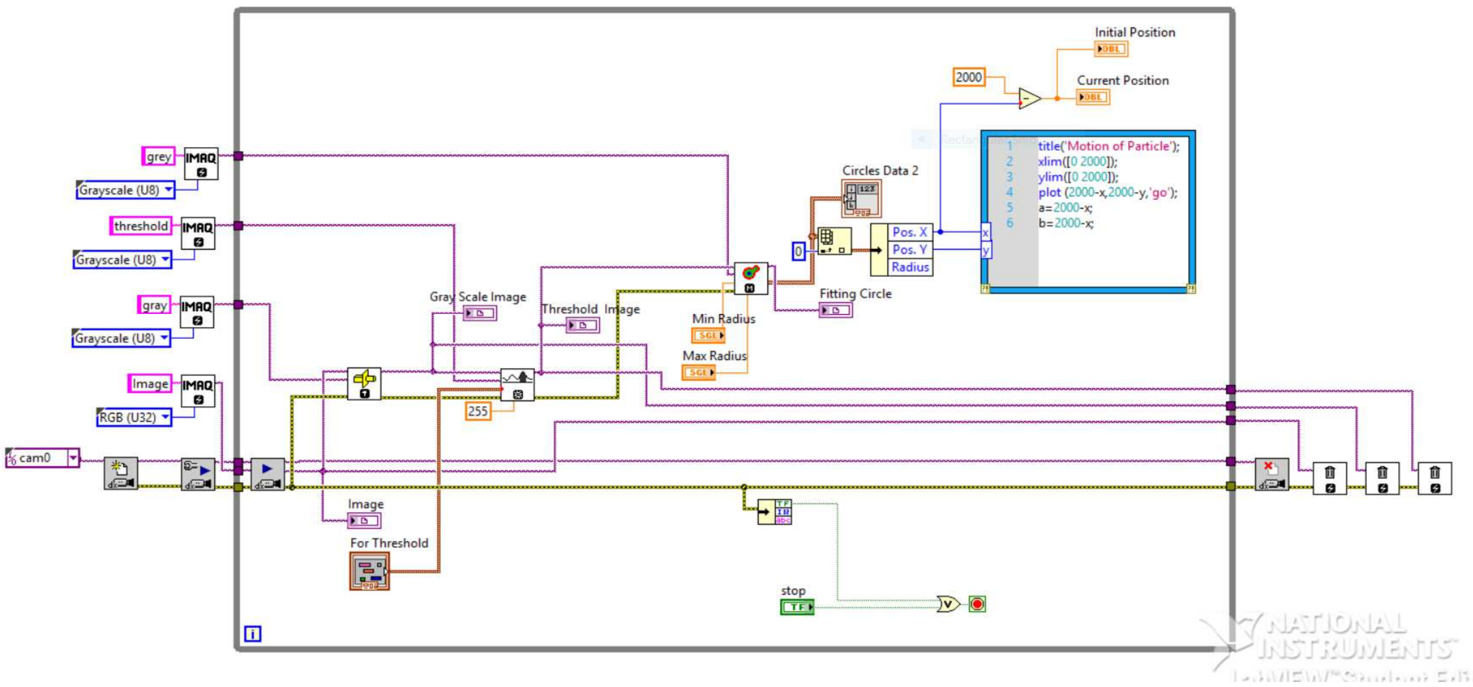
Sr. No.	***Name***	***Block***	***Function***
1	DAQmx Create Channel	 input terminal configuration minimum value maximum value task in physical channels name to assign units error in custom scale name DAQmx task out error out	Create virtual channel or set of virtual channels and adds them to a task.
2	DAQmx Timing	 samples per channel sample mode task/channels in error in DAQmx task out error out	Configures the number of samples to acquire or generate and creates a buffer when needed.
3	DAQmx Start Task	 task/channels in error in DAQmx task out error out	Transitions the task to the running state to begin the measurement or generation.
4	DAQmx Stop Task	 task/channels in error in DAQmx task out error out	Stops the task and returns it to the state the task was in before the DAQmx Start Task VI ran or the DAQmx Write VI ran with the auto start input set to TRUE.

5	DAQmx Write	 auto start task/channels in data error in DAQmx task out number of samples written error out	Writes samples to the task or virtual channels you specify.
6	DAQmx Clear		Clears the task.

<u>Block Diagrams</u>

1) Image Acquiring VI

- Figure 1 shows the block diagram for image acquiring.
- This VI uses image acquisition functions to acquire, transmit, etc. the photographs of the object to be tracked.
- Using configure grab and grab, we acquire images continuously instead of using 'snap' directly because images would suffer from delayed input.
- We acquire coloured images and then using cast image we convert them from 32 bit to a 8 bit image.
- Now using threshold, we convert this to binary image in such a way that there would be at least a minimum and maximum threshold of 0 and 255 pixels respectively.
- To make things simpler, we have considered a black coloured background to ignore the filter usage.
- Using IMAQ find circle, this program identifies the image as a circular object of which we get an array of cluster containing its x and y coordinate, radius and its surface area.
- Using these coordinates, we plot the motion of the particle, viz., tracking.
- Using data socketing, the value get passed to another VI for motor control and traversing purpose.



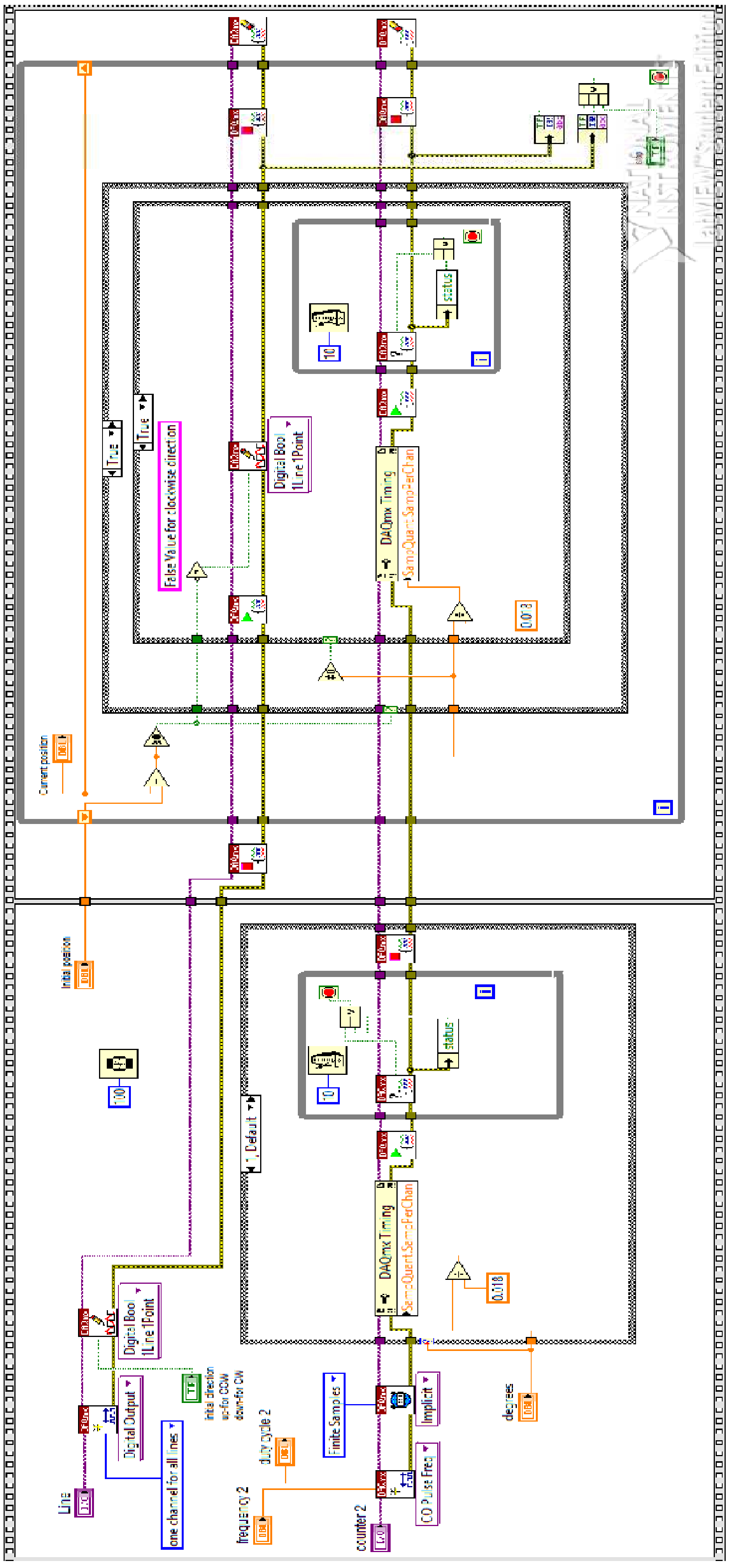
```
title('Motion of Particle');
xlim([0 2000]);
ylim([0 2000]);
plot (2000-x,2000-y,'go');
a=2000-x;
b=2000-x;
```

Figure 1: Block Diagram for acquiring continuous images and plotting its (x-y) motion in graph

2) Stepper VI

- Figures 2,3 and 4 shows the block diagram for stepper VI.
- This VI uses a stack sequence, three case structure and two while loops to initialize and process to obtain required results.
- Here, we initialized a digital line so as to give initial instructions to motor on how to rotate. It's clockwise when the switch is OFF and counter clockwise if it is ON.
- The stepper motor used has been given a setting of 20,000 steps per rotation. Hence, each step gives an accuracy of 0.018 degrees.
- Also, we initialize the stepper motor and bring it on zero position if it is not and we check whether the task has been completed. This is done in the first frame of the stacked sequence.
- Now the position of the object obtained via data socketing is compared with the initial position and if the difference is obtained, the motor starts traversing the object. It is to be noted that if positive difference is obtained, then the motor will run clockwise and if it is negative, it will run counter clockwise.
- The shift register will pass value of the previous object position obtained and it is subtracted from the new object's position.
- This process will continue till the difference becomes zero.
- Zero difference indicates the position that we were focused to achieve is reached, i.e., it has traversed the x position.

Figure 2: Block Diagram for Controlling the direction of motion of the Stepper Motor



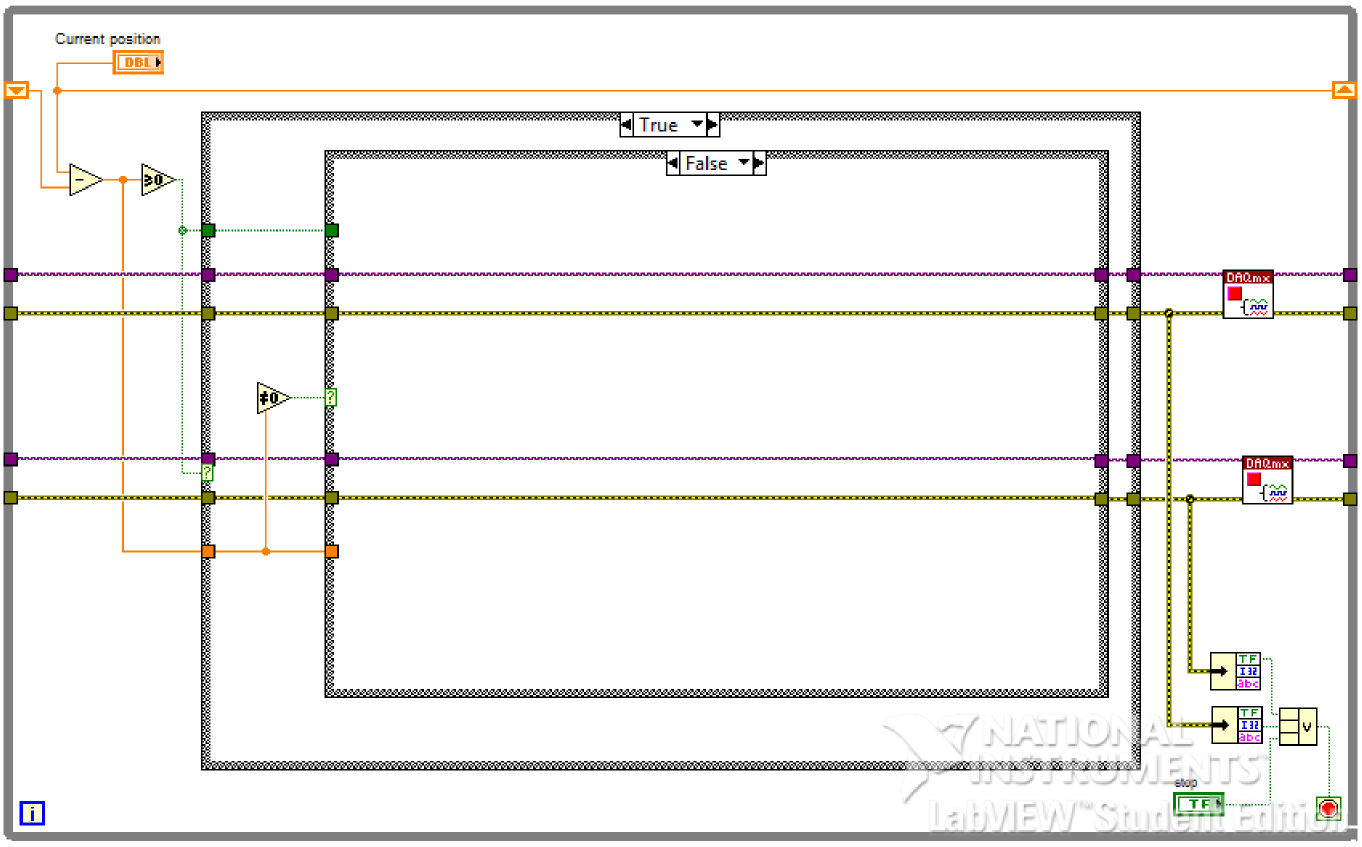


Figure 3: False Case of the second-case structure

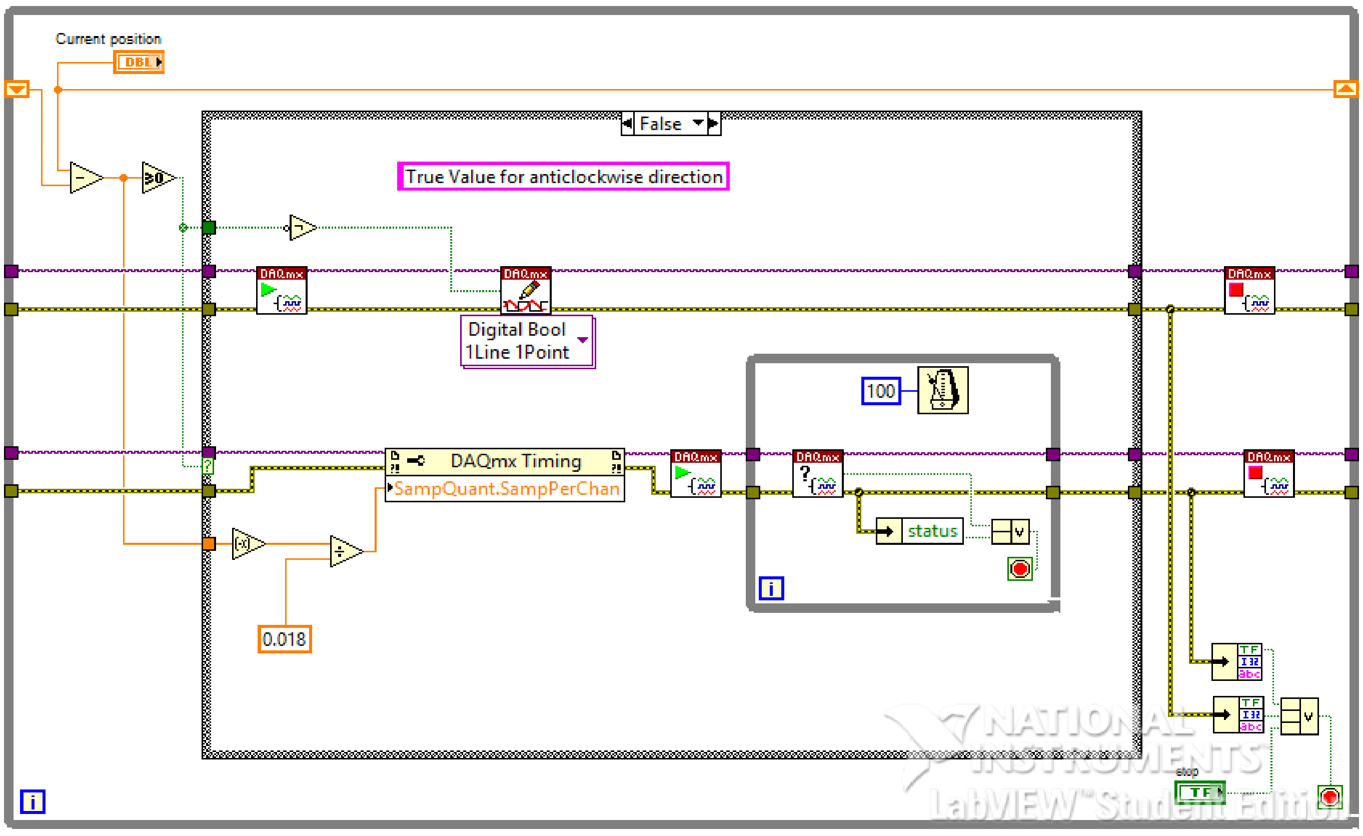

Current position
DBL
False
True Value for anticlockwise direction
Digital Bool 1Line 1Point
DAQmx Timing
SampQuant.SampPerChan
100
status
0.018
stop
NATIONAL INSTRUMENTS LabVIEW Student Edition

Figure 4: False Case of the First Case Structure

Future scope

- This principle could be extended to three dimensional axes so that we can obtain any objects' mechanical action. This could then be used to obtain the size of the object and control the motion of any autonomous vehicle.
- Frequency of the motor which is given in revolutions per seconds can also be controlled depending upon the speed of the moving object.