

<h2>Laboratory for Instrumentation and Measurement</h2>		
Group No:	EEL2-4	L1: in charge of the report
Date:		Assistant A2:
Professor:		Assistant A3:
<h3>Introduction to Measurement Automation (Part 2)</h3>		

Report History - Please leave all original papers in the report

Report 1	Date			Remarks	
	received				
	checked				
	result	o.k.			
		n.g.	→ 1. Correction		→ Term.....
1. Correction	Date			Last chance!!	
	received				
	checked				
	result	o.k.			
		n.g.	→ 2. Correction		→ Term.....
2. Correction	Date			Remarks	
	received				
	checked				
	result	o.k.			
		n.g.	→ not passed		→ back to L1.....

Final decision:

o.k.

not passed

Prof.

Objectives

- Experience the sensor and control interaction
- Gain experience in the graphical programming based on LabVIEW

Preparation

Solve the following programming task on paper:

The sensor head has to be moved so that it turns around in an angular range of +/- 45 degrees. The zero position will be set by the operator by pressing a button "position reset" at a specific time.

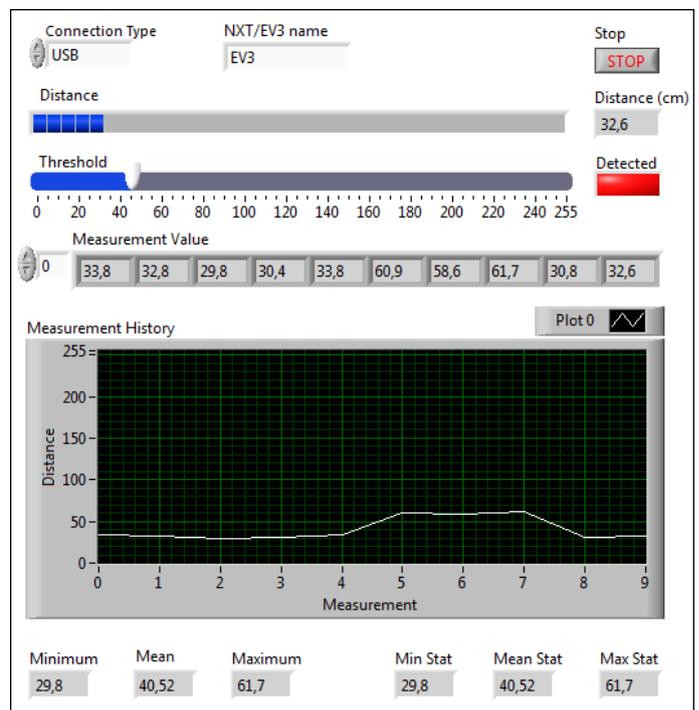
The following function blocks are available:

- | | |
|----------------------|--|
| motor unlimited | - starts the motor until a stop function is called |
| motor stop | - stops the motor |
| get output value | - read the position |
| reset motor position | - resets the position to zero |

Task 6: Arrays, graphs, statistics and saving data

Hardware and software setup as described in the previous lab session.

1. Load the program `task4.vi` and save it as `task6.vi`.
2. Limit the number of measurements by adding an additional stop condition if the number of measurements (see `i` tag in the lower left corner of the loop) exceeds a constant value of 10.
3. Save the program and test it.
4. Reduce the measurement rate by placing an isolated `Wait (ms)` block into the while loop and set the time delay to 0.5 s.
5. Add a `Waveform Graph` to the front panel, size it appropriately and call it `measurement history`. Label the axes appropriately and set the y axis to a fixed scale between 0 and 260.
6. In the block diagram, place the graph outside the while loop and connect it to the sensor output in the while loop. Click at the intersection between the while loop and the data flow and select `indexed`. Now the single measurements are automatically converted into an indexed array.
7. Right-click at the sensor data flow outside the while loop and select `create indicator`. Place the array indicator on the front panel.
8. Save and test the program. Check if you have captured exactly 10 measurement points.
9. Display mean, minimum and maximum value of the indexed array. There are several ways to do so. You may use `Statistics.vi` of the express palette or calculate the values manually by using `Array Size`, `Array Max & Min` and `Add Array Elements`.
10. Create the corresponding indicators on the front panel by right-clicking at the vi's output.
11. Save and test the program.
12. Save the data automatically in a file after the measurements are done. To do so, add the block `Write To Measurement File` from the `Output Palette` (→ `Express`) or `Write Delimited Spreadsheet` and configure it appropriately.
13. Save and test the program. Investigate the file.



Self test:

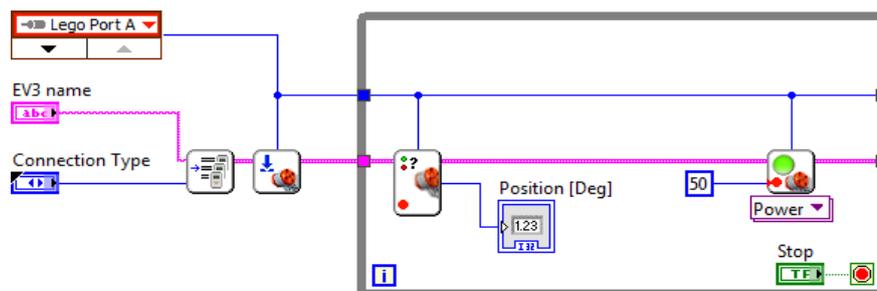
Try to explain to each other how you generate an array in LabVIEW and how to access the data.

Task 7: Rotation and position measurement

Scope of this task is to rotate the sensor head and to read the angular position. The starting point of this exercise is the `helloWorld.vi`.

Make sure that the motor is connected to port A. To control the EV3 use the MINDSTORMS Robotics → I/O palette.

1. Open the `helloWorld.vi` and save it as `task7.vi`.
2. Enhance the program so that the motor starts (unlimited) with a power setting of 50% and that it stops as soon as the user presses the Stop button. Find the appropriate function and use a `while` loop.



3. Add a control that enables changing the sign of the power setting, hence the direction of the motor.
4. The position of the motor itself is measured by an incremental encoder. Reset this encoder at the beginning of your program. Get the position value and display it in the front panel.
5. Save your program and test it.

Task 8: Scan

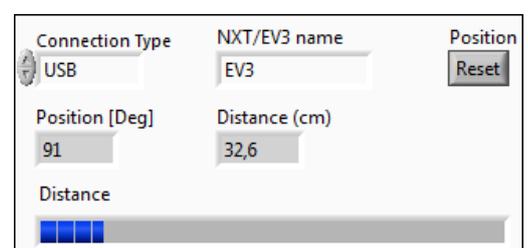
Enhance the program so that the sensor head turns 90 degrees and finally returns to its start position.

1. Save the program as `task8.vi`.
2. Replace the stop condition so that the sensor turns 90 degrees.
3. Add a second `while` loop to let the sensor head return to the zero position. The speed can be set to a power of 50.
4. Remove the position reset at the beginning of the program and activate it only if a reset button is pressed during the scan.

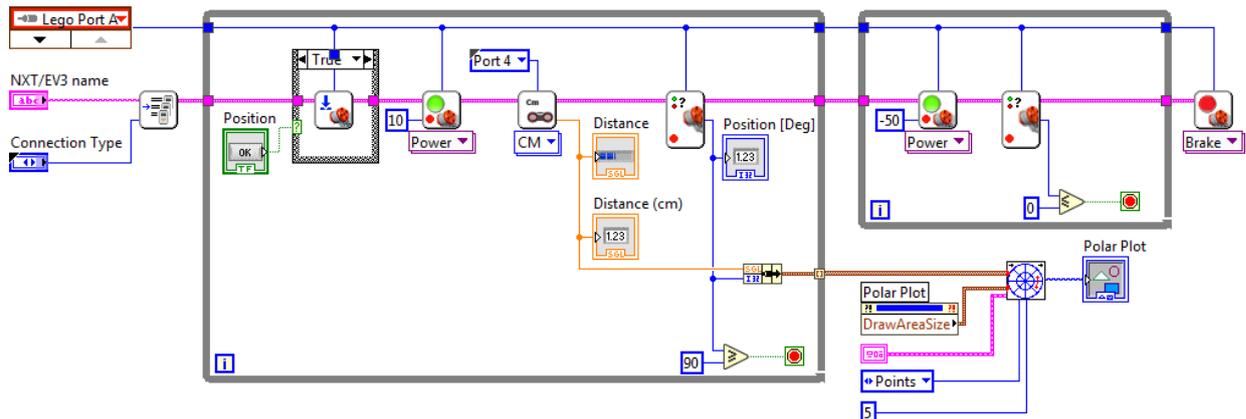
Task 9: Integration of ultrasonic function to scan function

Add the ultrasonic measurement to the scan function.

1. Save the virtual instrument as `task9.vi`.
2. Open the virtual instrument `task3.vi`. Copy and paste the appropriate code to carry out ultrasonic measurements during the scan procedure into `task9.vi` including the horizontal graduated bar.
3. Save and test the file.



Task 10: Radar plot



Display the results in a polar diagram as radar plot.

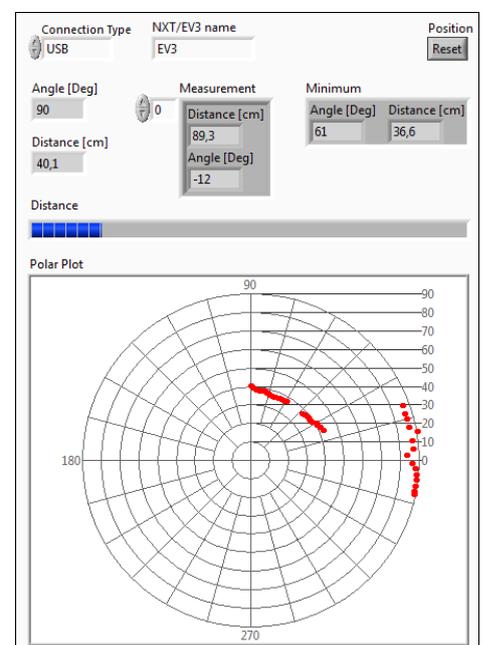
1. Save the file as `task10.vi`.
2. Add a polar plot indicator to the front panel.
3. Place the polar plot indicator outside the while loop.
4. Bundle the angle and the sensor reading to a cluster (search for bundle) and connect it to the polar plot indicator.
5. Right-click at the intersection of the bundle data flow and the while loop. Activate indexing.
6. Save and test the file. Check if the orientation of the display corresponds to the real measurement.
7. Set up 3 different scenarios with objects and capture the radar plots. Compare the radar plot with the real scene. Describe how a box is displayed in the radar plot and explain why. If you had boxes with different sizes, would you see the difference in size in the plot?

Task 11: Display measurements and closest object

1. Save the file as `task11.vi`.
2. Create an indicator for the measurement results outside the while loop. Place this indicator in the front panel.
3. Save the file and test how the array indicator is working.
4. Enhance the block diagram to find the measurement and the angle with the lowest distance. This information will provide the location and distance to the closest object.

To do so, use the array functions `Array Max & Min` and `Index Array` outside the while loop and use a separate (indexed) data flow for angle and distance. Bundle the minimum distance and the related measurement angle to a cluster and create an indicator.

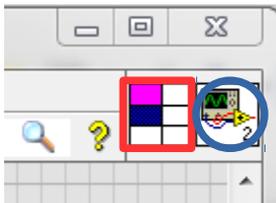
5. Save and test the file.



Task 12: Conversion into a virtual radar instrument

The functionality of the radar sensor can be easily integrated into other programs after converting it into a stand-alone virtual instrument.

1. Save the program as `task12.vi`.
2. Delete the motor constant and the Specify NXT/EV3 vi.
3. Create control blocks at the two unconnected wires and an indicator at the Stop Motors vi.
4. Go to the Front panel view and place the three new blocks at the bottom of the window.
5. Right-click at the connector icon in the upper-right corner (red square), select Patterns and choose the pattern with six terminals (two columns with three terminals each).



6. Click at the upper left corner of the connector icon and then on the NXTObject control.
7. Click at the center box of the icon's left terminal column and then on the Motor control.
8. Click at the lower left corner of the connector icon and then position reset control.
9. Click at the upper right corner of the icon and then the NXTObject indicator.
10. Click at the center box of the icon's right terminal column and then on the Minimum indicator.
11. Click at the lower right corner of the icon and then the Measurement array indicator.
12. Right-click at the icon in the upper-right corner and select Edit Icon.
13. Right-click at the icon (blue circle) and select Edit icon. Design the icon distinctly.
14. Save the file as `radar.vi`.
15. Open the `helloWorld.vi` and save it as `task12.vi`.
16. Replace the Tone vi by the radar.vi. Therefore right-click at the Tone vi, select Replace → All Palettes → Select a VI → select the file `radar.vi`.
17. Remove the Duration control and the Tone constant.
18. Create an indicator for the minimum distance to an object and its angle.
19. Save and test the program.