

## Position, Displacement, Distance

### Objectives:

- Observe and understand motion as a change of position;
- Measure the position of an object using the Time-of-Flight VL53L0X Motion Sensor connected at the Raspberry Pi;
- Use the PhyPiDAQ-Software to display motion as a change of position in real-time in a graphical form;
- Employ spreadsheets like LibreOffice or Excel to process the measurements stored in .csv files to analyse different motions;
- Gain an understanding of the difference between position, distance, and displacement;
- Interpret and compare position-time graphs for different types of motions such as:
  - moving in the positive direction versus moving in the negative direction;
  - moving fast versus moving slow at constant speed;
  - a constant speed motion versus an accelerated motion;
  - combinations of the above;
- Determine the Cart position at different given instants by reading graphs, charts and table of values;
- Calculate and compare displacements and distances for different time intervals.

### Materials and Equipment:

Raspberry Pi 3B+ or 4B	Variable Speed Motorized Cart
PhyPiDAQ-Software for Data collection	Reflector of metal, white or light colour, for reflecting the Laser beam towards the sensor's receiver;
VL53L0X Motion Sensor	Jumpers to connect the sensor to the Raspberry Pi
Breadboard	Clamps
Tripod Stand	Rod

### Set-up:

Connect the VL53L0X Motion Sensor to the Raspberry Pi according to the circuit shown in Figure 1. The students used the [Fritzing open-source](#) program to build this circuit.

Connect monitor, mouse and keyboard to the Raspberry Pi and fasten the breadboard with the sensor, as shown in the Figure 2 or in the Figure 3. Alternatively, you can use any other materials from your lab to fix the sensor into the right position to run the measurements.

Place the Variable Speed Motorized Cart about 3cm in front of the sensor. Stick the reflector on the side facing the sensor, perpendicularly oriented to the Laser beam. According to the datasheet of the VL53L0X Motion Sensor, a light-coloured reflector provides a better quality of the measurements.

A Variable Speed Motorized Cart from the [Phywe Company](#) with three-stage switch for forward, off, and reverse motion is used in this experiment. The wheels are geared to operate at the constant speed, which is adjusted through the speed control slider. Therefore this Motorized Cart is used to experience uniform motions with constant speed. Alternatively, one can use other Motorized Cart from other company.

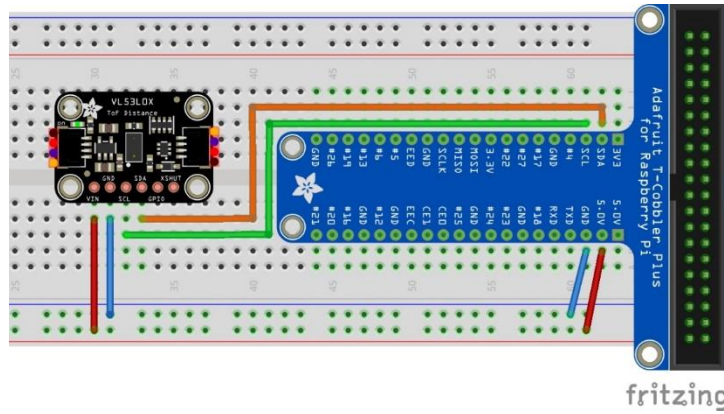


Figure 1 Connecting the VL53L0X Motion Sensor to the Raspberry Pi. The VL53L0X communicate to the Raspberry Pi via I<sup>2</sup>C-Interface.

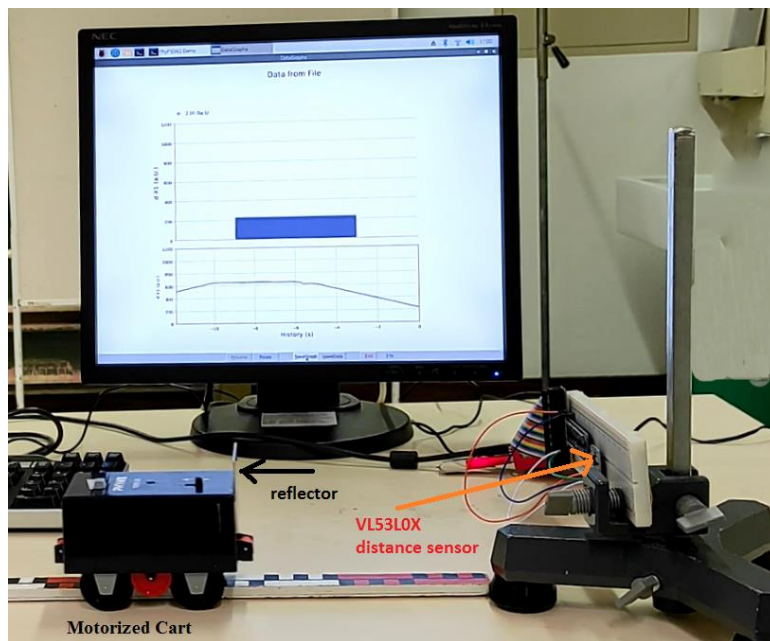


Figure 2 Set-up of the experiment with VL53L0X Motion Sensor and Variable Speed Motorized Cart

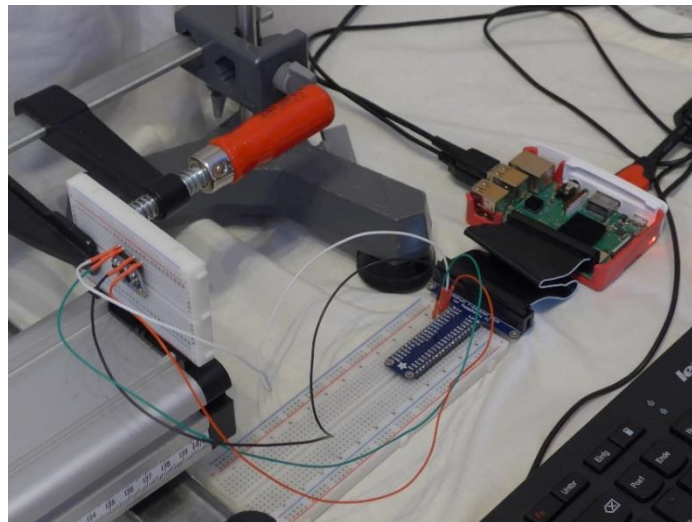


Figure 3 Connecting the VL53L0X Motion Sensor to the Raspberry Pi

## Procedure:

In order to visualise the Cart's position in real time one has to go through the following steps. Watch this video <https://youtu.be/Vd5KpSkuN2w> first to get an insight into the procedure of configuring the experiment.

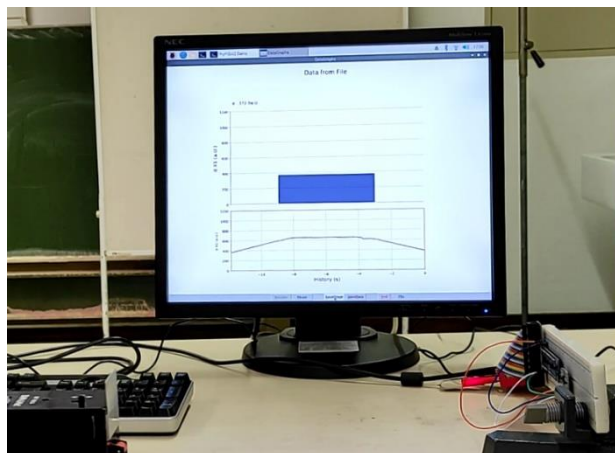
On the Graphical Interface of the PhyPiDAQ Software open the Work directory where the configuration files are going to be saved. In the `DAQ config`: text box introduce the pre-configured file `Distance.daq` from the directory `/home/pi/git/PhyPiDAQ/examples`.

Alternatively, choose the `default.daq` file and edit this according to the `VL53L0x_position_vs_time.daq` saved as .pdf file and uploaded in the **Activity-Based Physics** web page. To activate the VL53L0x Motion Sensor remove the #-Sign from the command `DeviceFile: config/VL53LxConfig.yaml` and press the `reload device config` button at the bottom of the `PhyPi Config` window. Click on the `Device Config` Button at the top of the window to change into the Sensor's configuration window. Once there one can edit the .yaml configuration of the VL53L0x Motion Sensor according to the `VL53LxConfig.yaml` saved as .pdf file and uploaded in the **Activity-Based Physics** web page. Here one can choose between the two sensor types and enter the desired measuring range. Note to click on `EditMode` at the top of `PhyPi Config` window to change or modify the commands in the configuration files.

After all the stages are finished, one can start the measurement by pressing the `StartRun` Button on the main graphical interface of the PhyPiDAQ. The measurement window opens. This video shows the experimental run with the display of position of the Motorized Cart in real-time: <https://youtu.be/E0Rt2T9bhUE>

Observe the measured Position versus Time in real-time on the measurement window while the Variable Speed Motorized Cart is set to move forwards, to stand still at certain location to the sensor, and then it is set to move backwards.

In order to read position's values at different instants one can activate the `DisplayModule: DataGraphs` in the .daq configuration of the experiment. On the measurement window the bar chart gives the Cart position at a certain instant, as shown in the Figure 5.



*Figure 5 Position vs. Time graph and bar chart for instant position magnitudes displayed with the `DataGraphs` Module.*

Save the Position vs. Time graph on the measurement window by pressing the `SaveGraph` Button on the taskbar. The picture called `DataGraph.png` is saved in the same folder containing `.daq` and `.yaml` configuration as well as the recorded measurements in the `.csv` file.

Press the `End` Button to complete the measurements. To repeat the measurements, choose to don't exist the `PhyPiDAQ` in the coming out menu window. The same `.daq` configuration is meant to be used to run again the experiment, for instance by setting the Motorized Cart at a higher speed.

## Data Collection:

The `PhyPiDAQ`-Software enables measurements' collection in `.csv` files. In the `Distance.daq` configuration one has to activate `DataFile: testfile.csv` as well as `CSVseparator: ','` by removing the `#`-Sign. Testfile can be renamed to the purpose of the experiment. Add the `#`-Sign to disable the `DataFile: null` command. On the Raspberry Pi's Terminal the name of the `.csv` file comes up.

Another option is to press the `SaveData` Button on the task bar of the measurement window, as shown in the Figure 4.

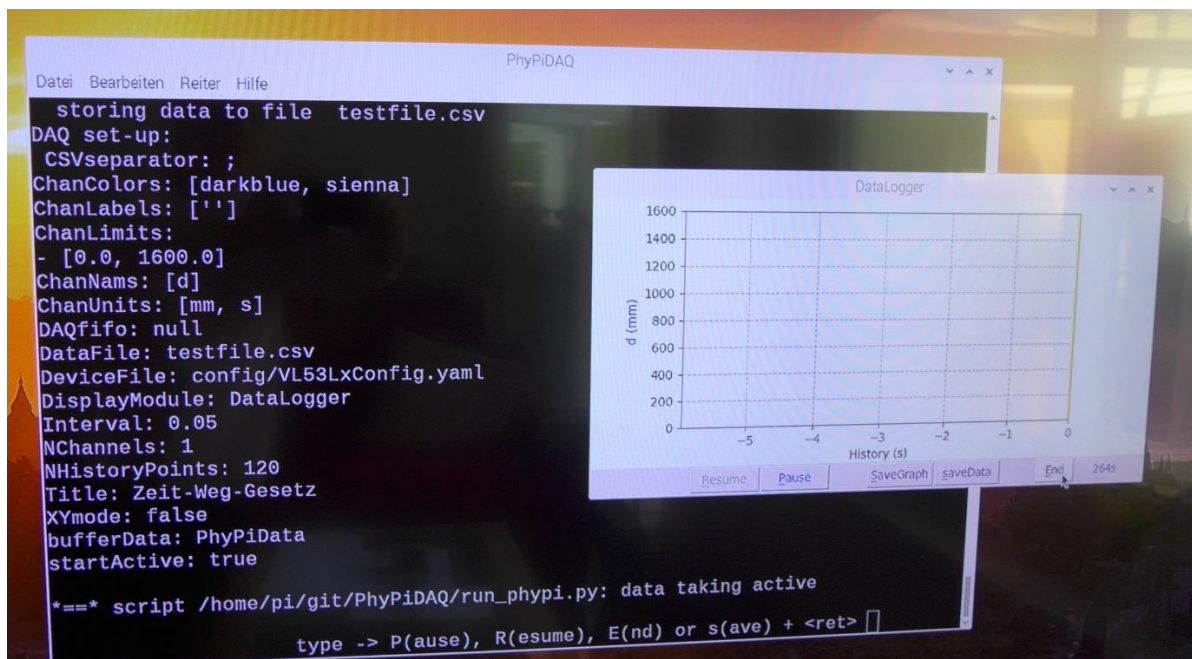


Figure 4. Raspberry Pi's Terminal and `PhyPiDAQ` measurement window. The recorded data are stored in `.csv` file. Alternatively, one can save data in `.dat` file by pressing the `SaveData` Button on the task bar of the measurement window.

## Data Analysis:

Observe the position vs. time graph in real-time and contrast it to the real motion of the Motorized Cart. Make predictions about the changes in the graph, if the Cart moves faster, slower, in the positive or negative direction.

Use the position vs. time graph saved as `DataGraph.png` file by pressing the `SaveGraph` Button in the task bar of the measurement window to read position values at different instants. An example is

shown in the Figure 6. Assign positive time values on the horizontal axis starting from 0s on the left corner of the coordinate system.

Annotate your position vs. time graph with descriptions of motion at each phase of data collection.

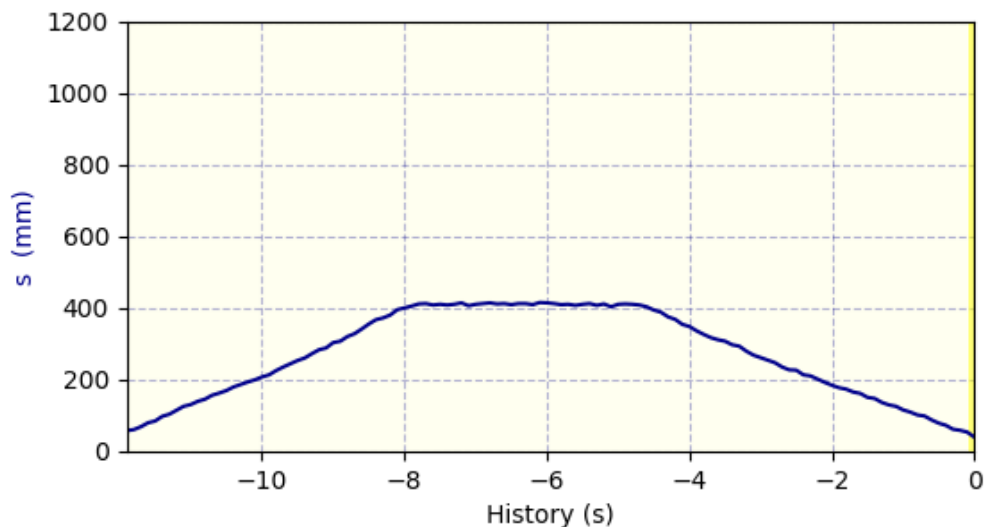


Figure 6. Position vs. Time graph saved in the measurement window.

Find initial and final positions and the corresponding instants at each phase of the motion and label them as  $s_1, t_1$ ,  $s_2, t_2$ ,  $s_3, t_3$  and  $s_4, t_4$  on the graph.

Knowing the Cart position as location relative to the motion sensor, identify the position's magnitudes at given instants. Read out the time values at different given positions.

The distance is the total distance the Cart travelled. It is always increasing as the Cart moves in any direction. Pick out two different instants from the graph. What was the distance travelled between the two instants? Choose two different instants at the same magnitude of the position. What was the distance travelled in this case?

What was the total distance travelled from the initial to the final position in this example? Is the distance a vector or scalar quantity?

The displacement,  $\Delta s$ , is the difference between the final position,  $s_2$ , and the initial position,  $s_1$ . Using the equation for displacement  $\Delta s = s_2 - s_1$ , find the displacement for two different instants on the graph.

What was the displacement for the initial and final positions of the Cart in this experiment? Compare this value to the magnitude of distance.

Is the displacement a vector or scalar quantity?

The Figure 7 shows the Motorized Cart at three different instants in a different run. Sketch the distance and the displacement from  $t_1$  to  $t_3$ . Compare the magnitudes of displacement and distance.



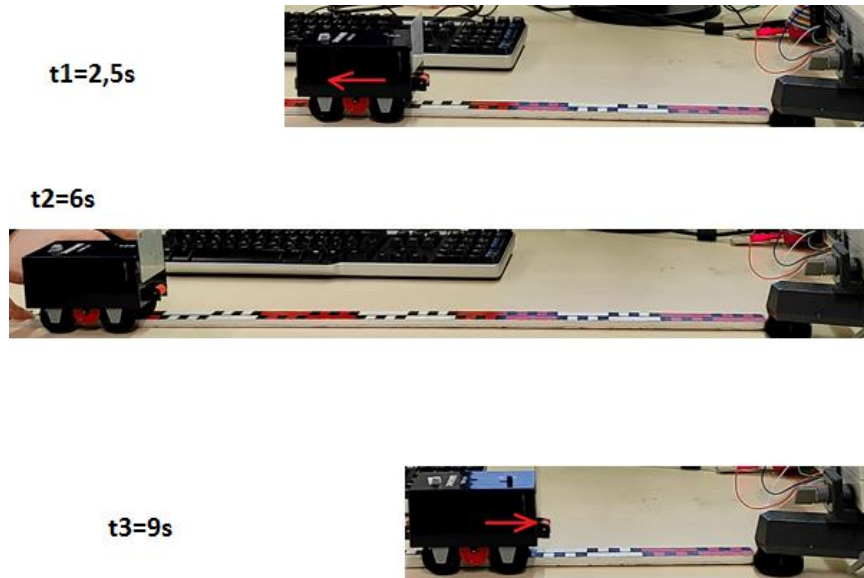


Figure 7. Motorized Cart at three different instants

The measurements stored in [position\\_vs\\_time\\_Motorized\\_Cart.csv](#) file allow a more accurate analysis of the recorded position at different instants. In the Distance.daq configuration the logging interval has been set at 0.1 seconds, meaning that Cart position was recorded every 0.1s. The logging interval can be improved by setting the logging interval at 0.05s.

Now open the [position\\_vs\\_time\\_Motorized\\_Cart.csv](#) file in Excel or in LibreOffice installed on the Raspberry Pi. Plot the recorded position and time in a scatter chart. Compare your result with the chart shown in the Figure 8.

Go through the same questions by reading position and time magnitude from the table of values or from the scatter chart.

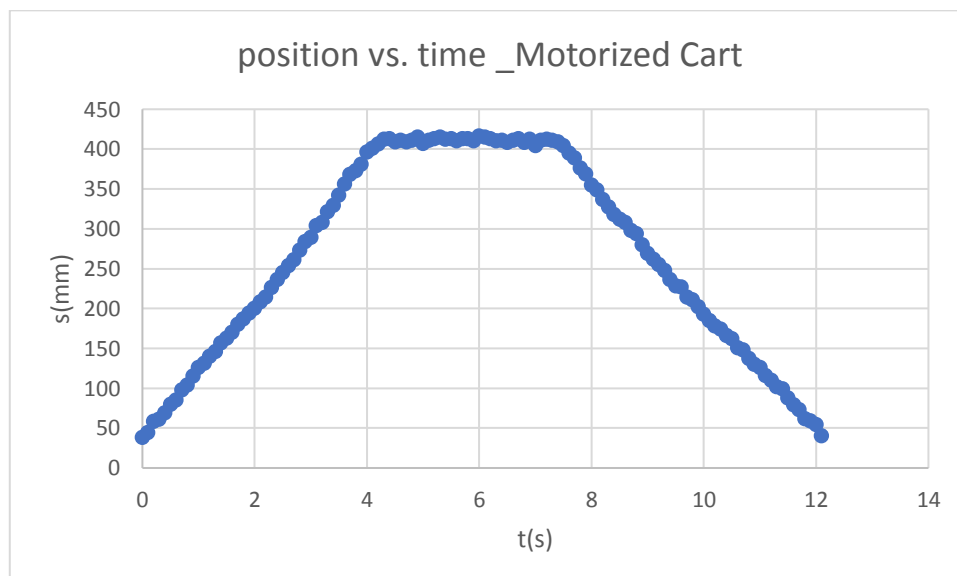


Figure 8. Position vs. time of the Motorized Cart displayed in an Excel-scatter chart based on the measurements stored in the [position\\_vs\\_time\\_Motorized\\_Cart.csv](#) file.