



RoboAGVSim AGV Simulator User Manual

V1.0, December 1, 2017

Visit www.roboteq.com to download the latest revision of this manual

©Copyright 2017 Roboteq, Inc

Table of Contents

1. Introduction.....	2
2. Installation	3
2.1 Requirements.....	3
2.2 Installation Steps.....	3
2.3 Directory Structure	6
3. How to Use	7
3.1 Main Screen	7
3.2 MicroBasic Code Editor.....	8
3.3 Track Editor	9
3.4 Controller Configuration Dialog.....	11
4. Sample in Action	14
5. Simulator Special Commands	16

1. Introduction

RoboAGVSim, a simulator that lets AGV (Automatic Guided Vehicle) designers program, test and fine tune their Automated Guided Vehicles in the comfort of their PC. A Magnetic track guided AGV can be built using only two Roboteq components: An MGS1600 magnetic guide sensor and any Roboteq Dual Channel motor controller. The sensor measures the distance to the center of the track. Using its built-in programming language, the motor controller takes this data and computes the power to apply to the left and right motor needed for the robot to stay on track.

The simulator opens with the image of a mobile robot on a blank screen on which users can draw or import a drawing of the track. A source code editor is available for writing the program that will be simulated. A robot icon can then be placed anywhere on the track and made to run. The AGV will move according the user program and uses the Robot's real physical characteristics, such as wheel base, gear ratio, wheel diameter and sensor distance from the pivot point.

Once the program has been verified in the simulator, it can be loaded into the Roboteq controller that is on the real AGV.

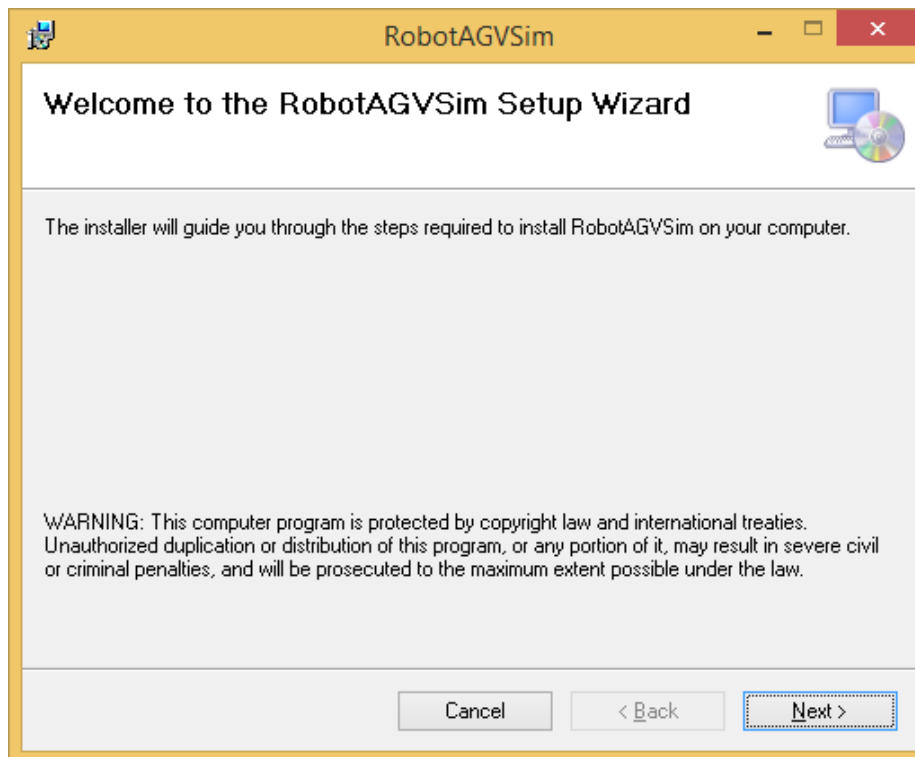
2. Installation

2.1 Requirements

- Microsoft Windows 7 or higher.
- Microsoft .NET Framework 2.0.
- Microsoft .NET Framework 4.0.

2.2 Installation Steps

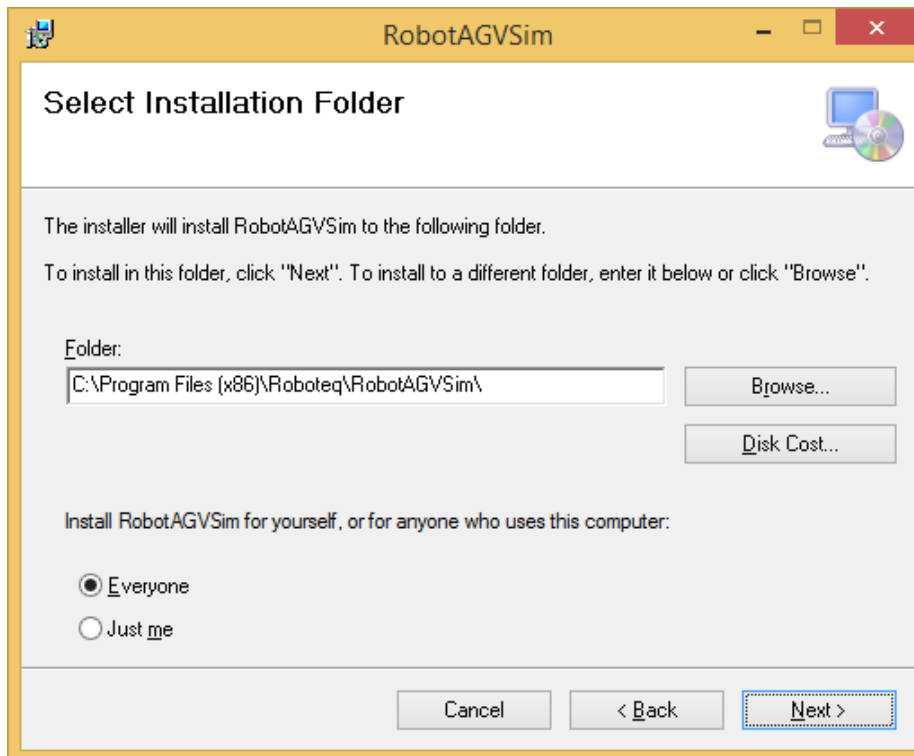
- Run **setup.exe**, on the welcome screen press **Next** button.



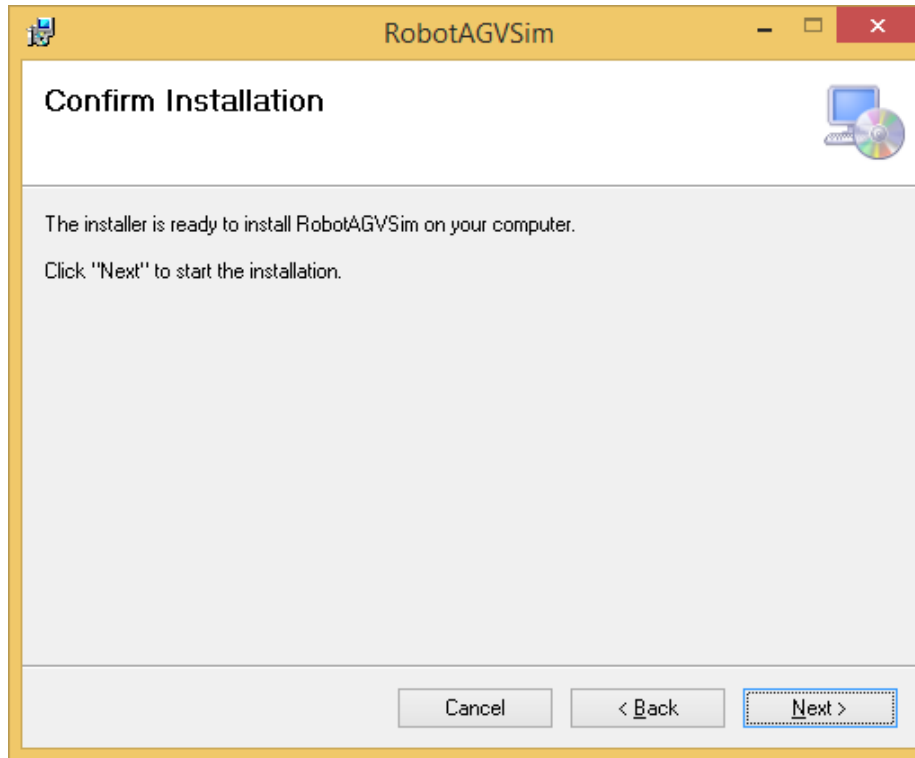
- In License Agreement screen check **"I Agree"** then press **Next** button.



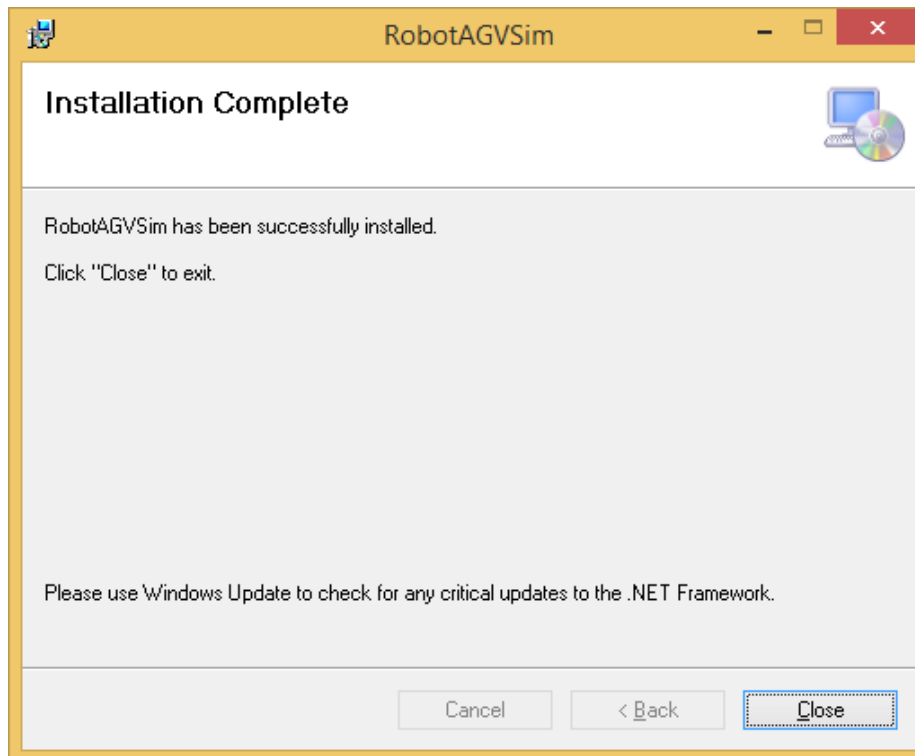
- Choose where you would like to install the application then press **Next** button.



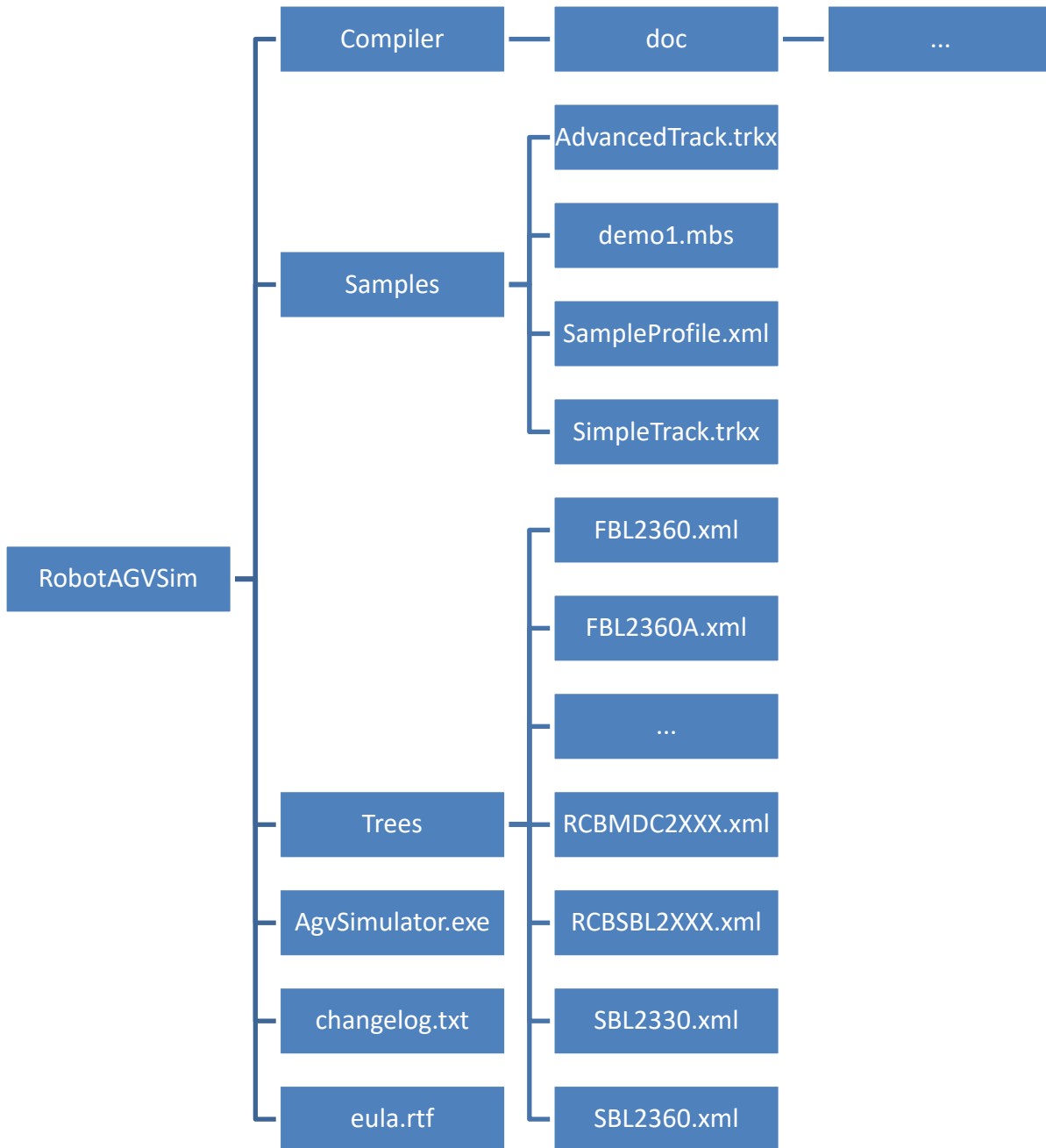
- In the Confirm Installation screen, press **Next** button to proceed with the installation process.



- After installation completes, press **Close** button.



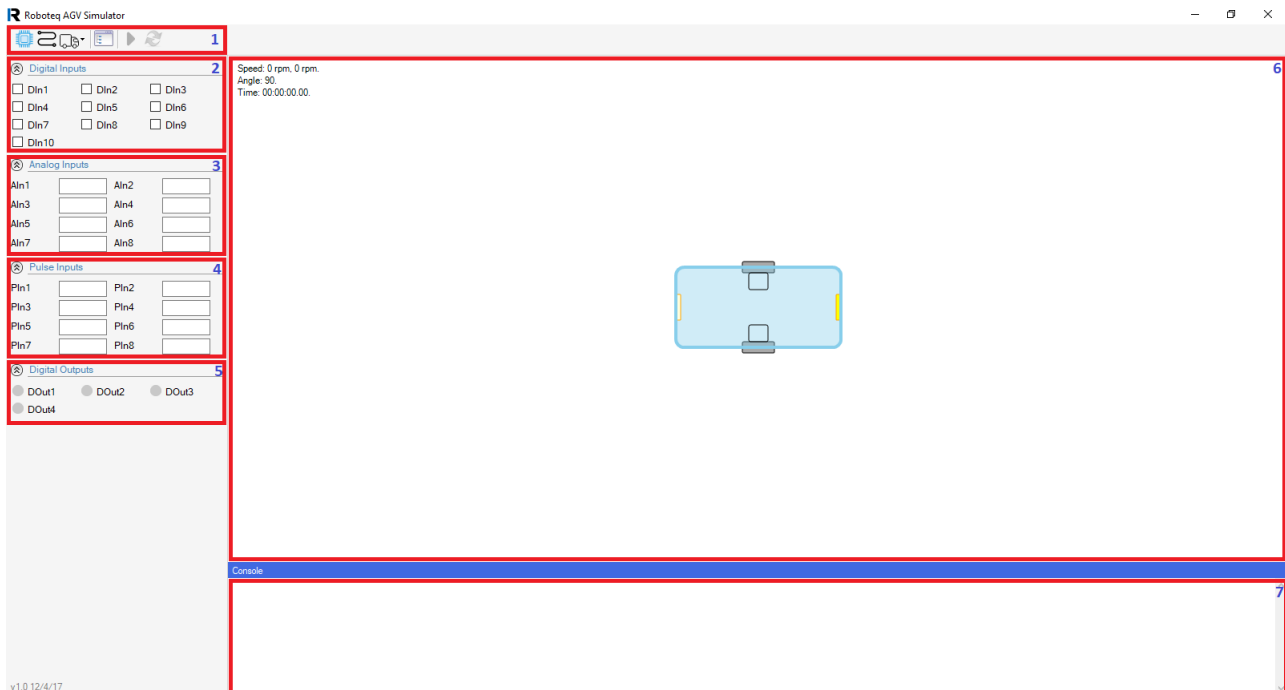
2.3 Directory Structure









- **Compiler/doc:** Contains the documentation for MicroBasic scripting.
- **Samples:** Contains samples files shipped with the simulator.
- **Trees:** Contains configuration for the supported controller models.
- **AgvSimulator.exe:** The simulator executable.
- **changelog.txt:** Contains the changes across all versions.
- **eula.rtf:** The end user license agreement file.

3. How to Use

3.1 Main Screen



1. **Main Toolbar:** Contains the following buttons:

	Open Code	Displays the MicroBasic code editor.
	Edit Track	Displays the track editor dialog.
	Vehicle	Configure vehicle position and orientation.
	Configure Tree	Displays controller configuration dialog.
	Play/Pause	Play/Pause MicroBasic code execution.
	Reset	Reset MicroBasic code execution.

2. **Digital Inputs:** Simulates the digital inputs that feed into the controller.
3. **Analog Inputs:** Simulates the analog inputs that feed into the controller.
4. **Pulse Inputs:** Simulates the pulse inputs that feed into the controller.
5. **Digital Outputs:** Simulates the digital outputs that results from the controller.
6. **Simulation Window:** Displays the simulation process in addition to simulation information.

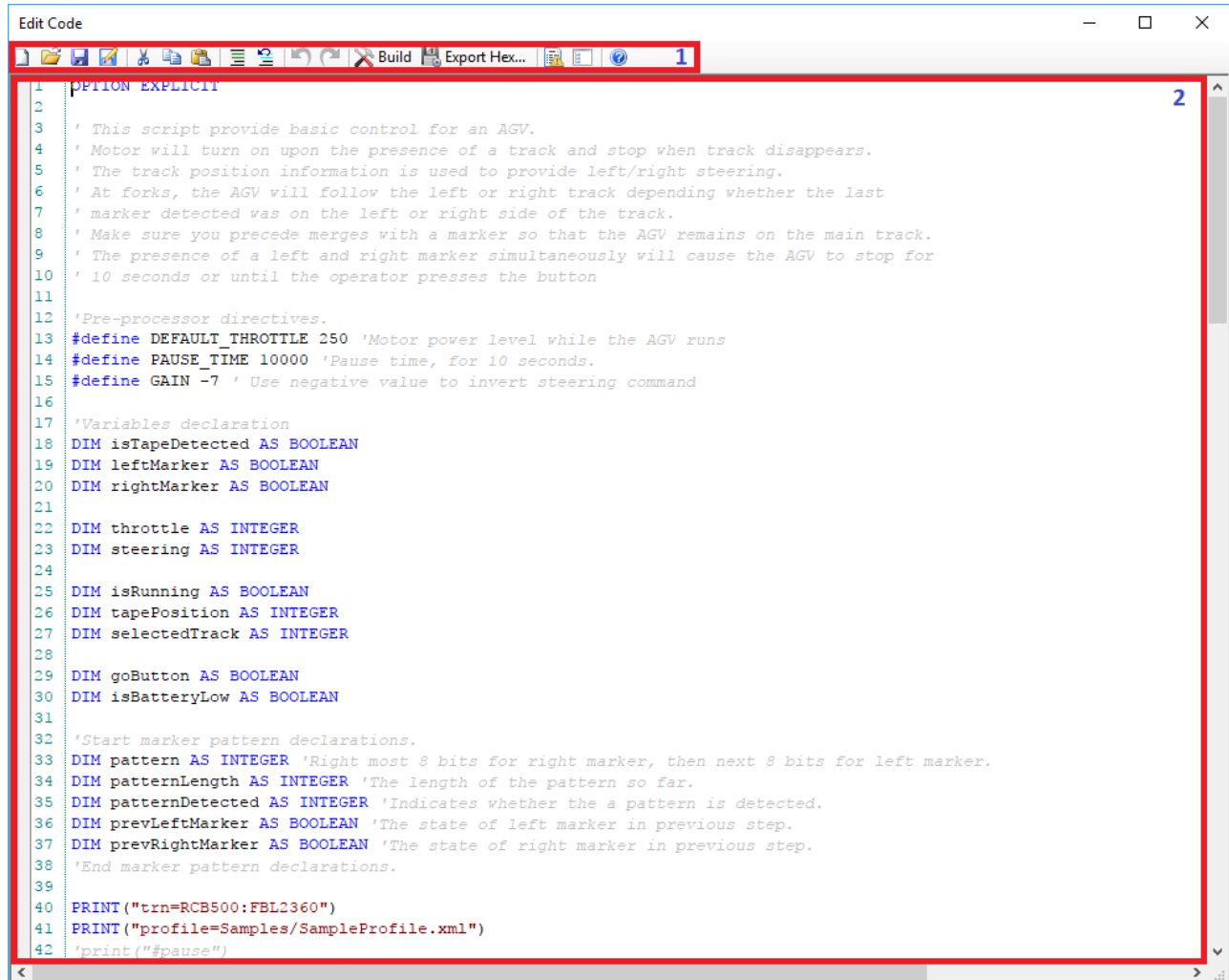
In the simulation window you can use the following:

- Drag with left mouse button causes panning.
- Use mouse scroll to zoom in/out.
- Double click with left mouse button to center the objects.

7. **Console:** Displays the console output from the MicroBasic code.

3.2 MicroBasic Code Editor












The MicroBasic code editor allows users to write and compile scripts with its code highlighting feature and undo/redo capability.








```

1  OPTION EXPLICIT
2
3  ' This script provide basic control for an AGV.
4  ' Motor will turn on upon the presence of a track and stop when track disappears.
5  ' The track position information is used to provide left/right steering.
6  ' At forks, the AGV will follow the left or right track depending whether the last
7  ' marker detected was on the left or right side of the track.
8  ' Make sure you precede merges with a marker so that the AGV remains on the main track.
9  ' The presence of a left and right marker simultaneously will cause the AGV to stop for
10 ' 10 seconds or until the operator presses the button
11
12 'Pre-processor directives.
13 #define DEFAULT_THROTTLE 250 'Motor power level while the AGV runs
14 #define PAUSE_TIME 10000 'Pause time, for 10 seconds.
15 #define GAIN -7 ' Use negative value to invert steering command
16
17 'Variables declaration
18 DIM isTapeDetected AS BOOLEAN
19 DIM leftMarker AS BOOLEAN
20 DIM rightMarker AS BOOLEAN
21
22 DIM throttle AS INTEGER
23 DIM steering AS INTEGER
24
25 DIM isRunning AS BOOLEAN
26 DIM tapePosition AS INTEGER
27 DIM selectedTrack AS INTEGER
28
29 DIM goButton AS BOOLEAN
30 DIM isBatteryLow AS BOOLEAN
31
32 'Start marker pattern declarations.
33 DIM pattern AS INTEGER 'Right most 8 bits for right marker, then next 8 bits for left marker.
34 DIM patternLength AS INTEGER 'The length of the pattern so far.
35 DIM patternDetected AS INTEGER 'Indicates whether the a pattern is detected.
36 DIM prevLeftMarker AS BOOLEAN 'The state of left marker in previous step.
37 DIM prevRightMarker AS BOOLEAN 'The state of right marker in previous step.
38 'End marker pattern declarations.
39
40 PRINT ("trn=RCB500:FBL2360")
41 PRINT ("profile=Samples/SampleProfile.xml")
42 'print("#pause")
  
```

1. **Toolbar:** Contains the following buttons:

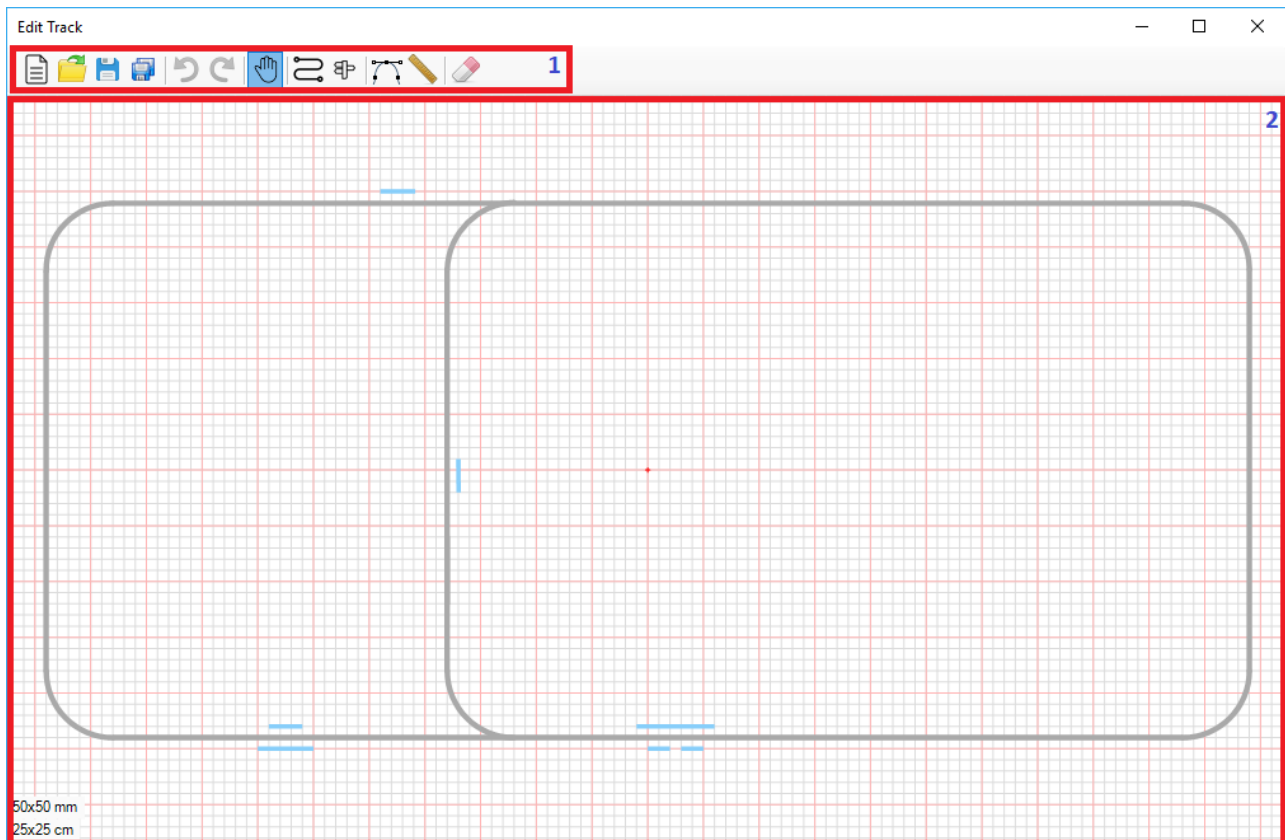
	New	Create new script file.
	Open	Open existing script file.
	Save	Save the current script file.
	Save As	Save a copy of the current script file.
	Cut	Cut the selected code into clipboard.
	Copy	Copy the selected code into clipboard.
	Paste	Paste the code from the clipboard.
	Comment	Comment the selected lines.
	Uncomment	Uncomment the selected lines.
	Undo	Undo last action.
	Redo	Redo last undo.

	Build	Build the code.
	Export Hex	Export the byte code into hex file.
	Errors	Show/Hide errors window.
	Syntax	Show/Hide abstract syntax tree of the latest build.
	Help	Open the help for MicroBasic script.






2. **Code View:** The place where user can type the MicroBasic script with the ability of code highlighting.

3.3 Track Editor

The track editor allows users to draw tracks to simulate real track with the ability to draw markers in real scale.



1. **Toolbar:** Contains the following buttons:

	New	Create new track.
	Open	Open existing track file.
	Save	Save the current track.
	Save As	Save a copy of the current track.
	Undo	Undo last action.



Redo Redo last undo.



Panning Panning mode:

- Use mouse wheel to zoom in/out.
- Drag with left mouse button to pan the drawing.
- Double click with left mouse button to fit the drawing in window.



Track Draw track mode:

- Use mouse wheel to zoom in/out.
- Drag with middle mouse button to pan the drawing.
- Drag with left mouse button while holding ALT key to pan the drawing.
- Double click with middle mouse button to fit the drawing in window.
- Double click with left mouse button while holding ALT key to fit the drawing in window.
- Use subsequent left clicks to draw track segments. While in segment drawing right click or ESC key exits the current segment drawing.
- Drag with right mouse button to erase drawings.
- Pressing ESC key to set panning mode.



Marker Draw marker mode:

- Use mouse wheel to zoom in/out.
- Drag with middle mouse button to pan the drawing.
- Drag with left mouse button while holding ALT key to pan the drawing.
- Double click with middle mouse button to fit the drawing in window.
- Double click with left mouse button while holding ALT key to fit the drawing in window.
- Use subsequent two left clicks to draw marker segment. While in segment drawing right click or ESC key exits the current segment drawing.
- Drag with right mouse button to erase drawings.
- Pressing ESC key to set draw track mode.



Curve Curvature mode:

- Pressing curve button turns on curvature mode, where each segment/arc middle point is drawn in red with yellow border.
- Use mouse wheel to zoom in/out.
- Drag with middle mouse button to pan the drawing.
- Drag with left mouse button while holding ALT key to pan the drawing.
- Double click with middle mouse button to fit the drawing in window.
- Double click with left mouse button while holding ALT key to fit the drawing in window.
- Drag segment/arc middle point to create arc/segment from segment/arc. While in dragging, right click or ESC key cancels.
- Pressing ESC key to set draw track mode.



Measure Dimension Mode:

- Use mouse wheel to zoom in/out.
- Drag with middle mouse button to pan the drawing.
- Drag with left mouse button while holding ALT key to pan the drawing.
- Double click with middle mouse button to fit the drawing in window.
- Double click with left mouse button while holding ALT key to fit the drawing in window.
- Move over segment/arc to display its dimensions in the top left corner of the drawing area.
- Pressing ESC key to set draw track mode.



Erase Erasing Mode:

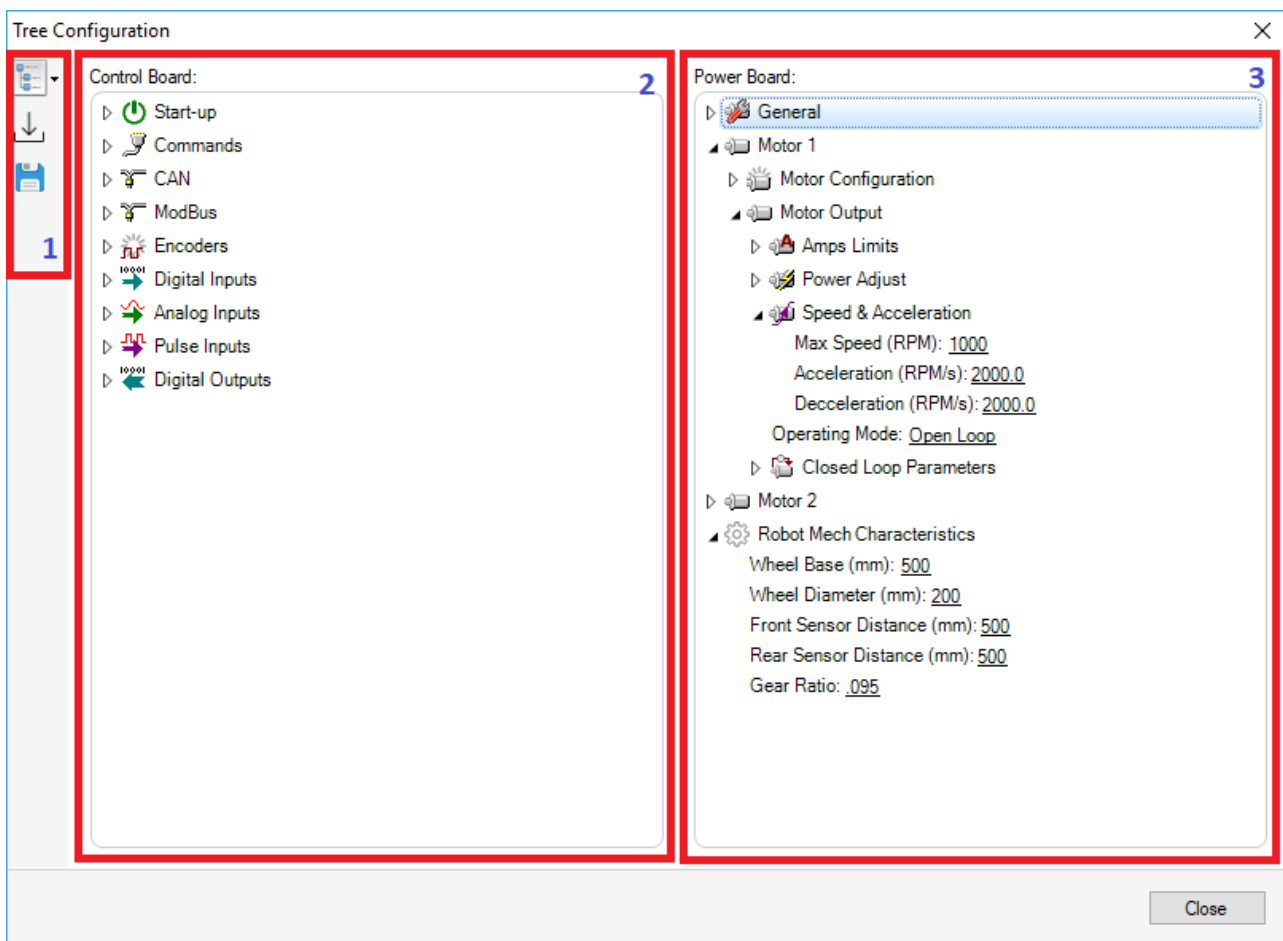
- Use mouse wheel to zoom in/out.
- Drag with middle mouse button to pan the drawing.
- Drag with left mouse button while holding ALT key to pan the drawing.
- Double click with middle mouse button to fit the drawing in window.
- Double click with left mouse button while holding ALT key to fit the drawing in window.
- Drag with left mouse button to erase drawings. While in dragging, right click or ESC key cancels.
- Pressing ESC key to set draw track mode.

2. **Drawing Area:** The place where track drawing takes place.

- In the top left corner, information about the current segment is displayed while drawing. Also, the information about segment/arc under mouse is displayed there in dimension mode.
- In the bottom right corner, information about the grid scale is displayed. The first line is the small grid scale. The second line is the large grid scale.

3.4 Controller Configuration Dialog

The controller configuration dialog allows users to set the configuration for the controller they using in the simulation process.



1. **Toolbar:** Contains the following buttons:



Select Board Displays a list of supported boards to select from. The following are the list of supported models:

- FBL2360, FBL2360A
- GBL2360, GBL2372, GBL2396
- MDC2230, MDC2250, MDC2260
- MDC2450, MDC2460
- SBL2330, SBL2360



Load Profile Load the controller profile from xml file. The profile xml file can be obtained from Roborun+ utility or from previously saved profile.

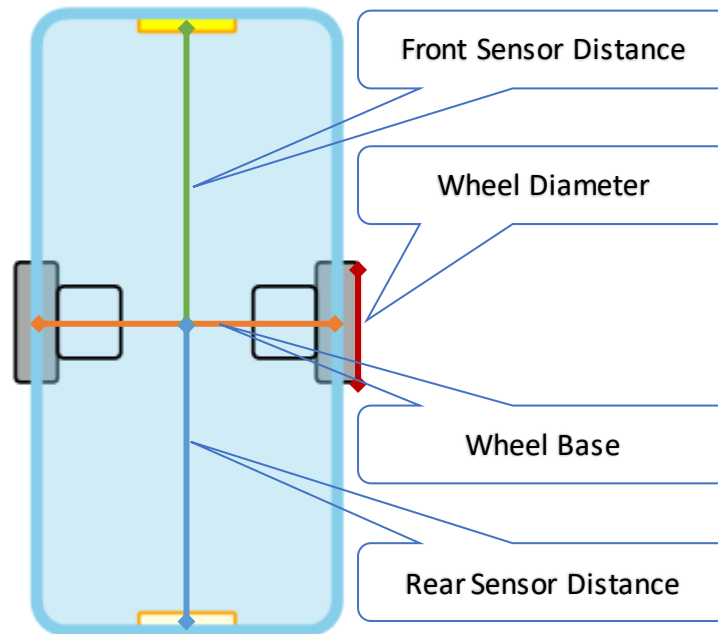


Save Profile Save the current profile into xml file.

2. **Control Board Configuration:** Contains the configuration parameters for the control board model.
3. **Power Board Configuration:** Contains the configuration parameters for the power board model. These configurations determine the characteristics of the motors such as Speed and Acceleration and also the characteristics of the Robot:

- **Wheel base:** The length of the wheel base in millimeters.
- **Wheel diameter:** The diameter of the wheels in millimeters.
- **Front sensor distance:** The distance of the front sensor from the middle of wheel base.

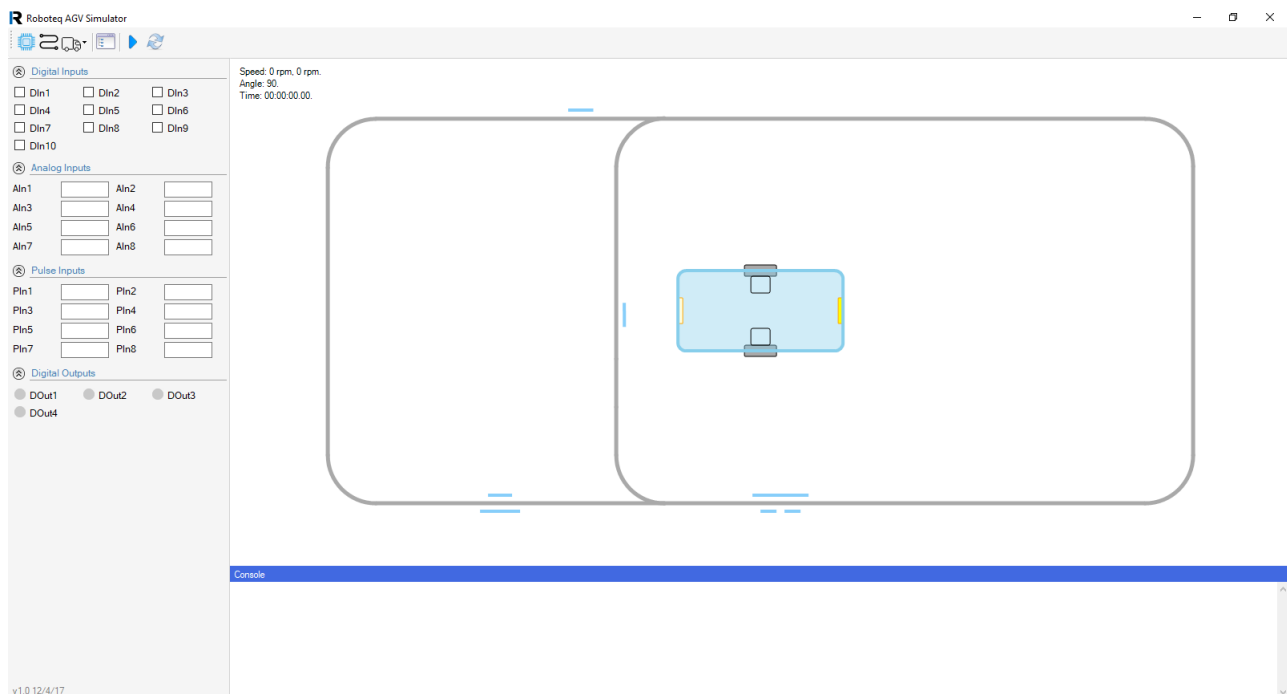
- **Rear sensor distance:** The distance of the rear sensor from the middle of the wheel base. Putting zero in this field means that there is no rear sensor.
- **Gear Ratio:** The ratio between motor cycles and wheel cycles.



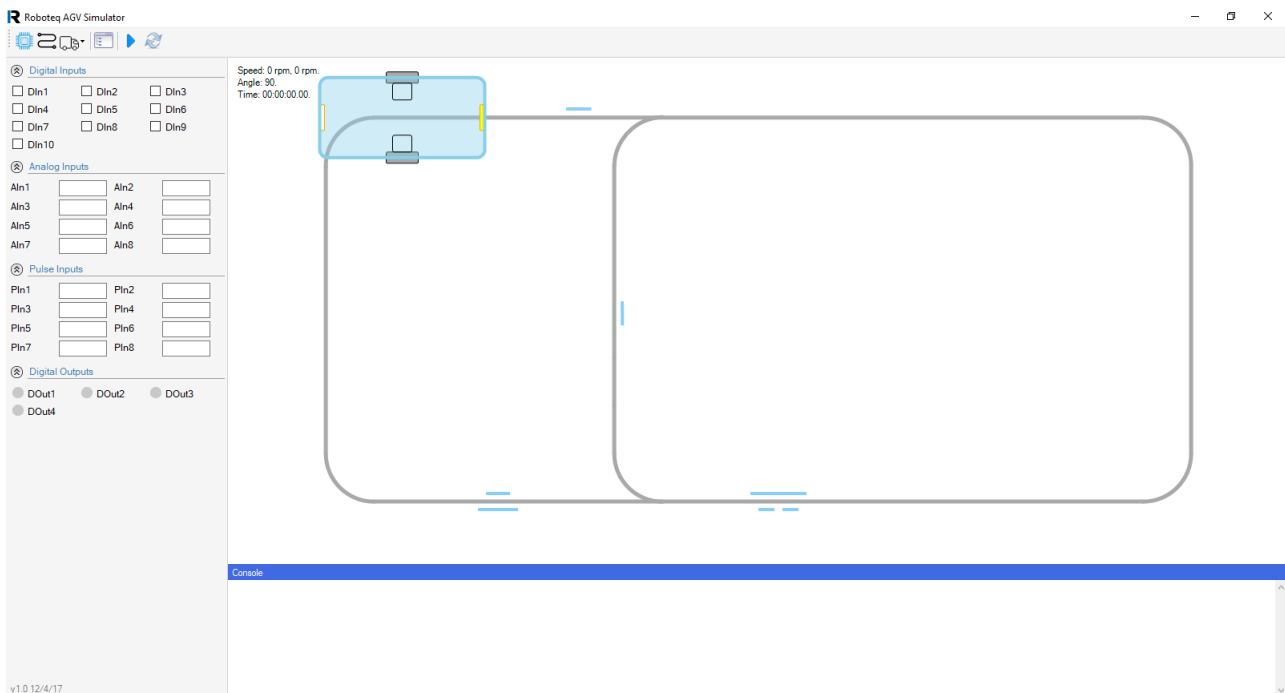
4. Sample in Action

In this section we'll use the **SimpleTrack.trkx** and **demo1.mbs** to show how the simulation is done using the AGV Simulator. These sample files can be found in **Samples** folder.

1. Run AGV Simulator.
2. From the toolbar in **Main Screen**, press **Open Code** button to show the **Code Editor**.
3. From the **Code Editor**, press **Open** button. Navigate to **Samples** folder and choose **demo1.mbs** then close the **Code Editor**.
4. From the toolbar in **Main Screen**, press **Edit Track** button to show the **Track Editor**.
5. From the **Track Editor**, press **Open** button. Navigate to **Samples** folder and choose **SimpleTrack.trkx** then close the **Track Editor**.



6. We need to place the vehicle in the correct position. From the toolbar choose **Vehicle** then press **Position** menu item.
7. In the **Simulation Window**, move the vehicle to the correct place then press the left mouse button.



8. From the toolbar, press **Play** button to start the simulation process, you will notice the vehicle is moving following the track.
9. In simple track we are using the following markers:
 - Single marker on left of the track: Causes the vehicle to branch left on the following forks.
 - Single marker on right of the track: Causes the vehicle to branch right on the following forks.
 - Large marker on right of the track and two short markers on left: Cause the vehicle to branch left if the battery is low (DIN3 set) otherwise it will branch right.
 - Large marker on left of the track and a shorter one on right: Causes the vehicle to stop if the battery is low (DIN3 set) otherwise it will continue moving.
10. You will notice red point is lightening when the sensor detects a track and a green point is lightening when the sensor detects a marker.
11. In MicroBasic code the DOUT1 indicates branch left and DOUT2 indicates branch right.

5. Simulator Special Commands

Print statements in MicroBasic code cause the provided parameters to be printed in the **Console** window. Some **Print** statements are interpreted in different way:

Example	Description
<code>PRINT("trn=RCB500:FBL2360")</code>	Causes the simulator to load the configuration of the given controller model.
<code>PRINT("profile=Samples/SampleProfile.xml")</code>	Causes the simulator to load the profile from given path. Absolute and relative paths are allowed in this command. Note: The relative path is computed from the folder where the simulator executable file resides.
<code>PRINT("#pause")</code>	Causes the simulator to pause the execution of the script at this point.