Microsoft Robotics Studio Simulation Environment

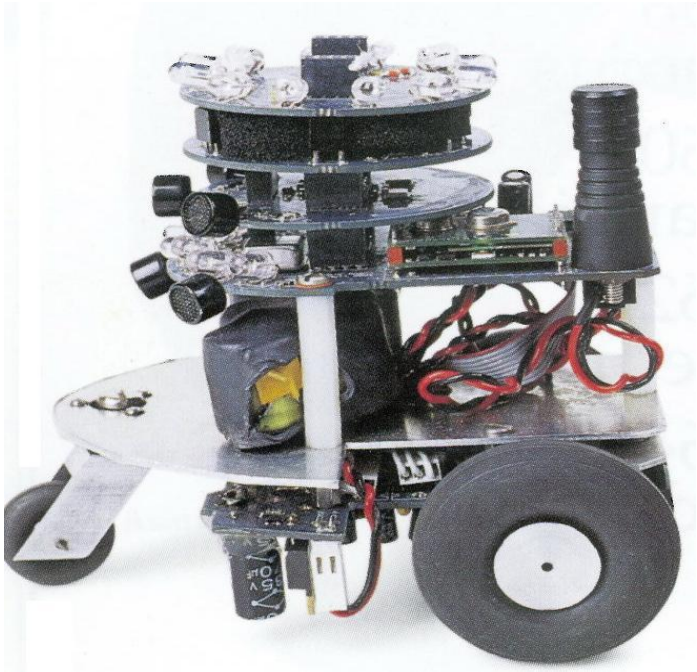
Instructor
Prof. Shih-Chung Kang

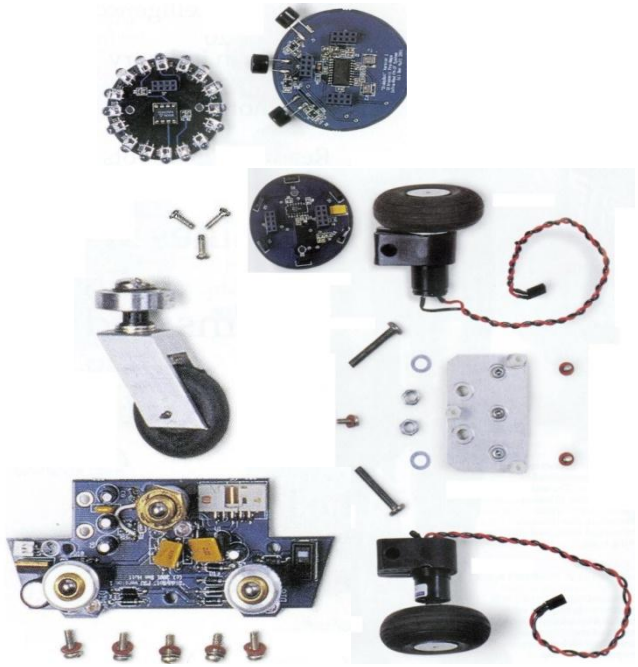
2008 Spring

(I) Overview

Challenges Posed by Robotics Development

- **Difficulty of concurrent use**
- **Difficulty of hardware troubleshooting**
- **Robotics hardware can be expensive and hard to find**

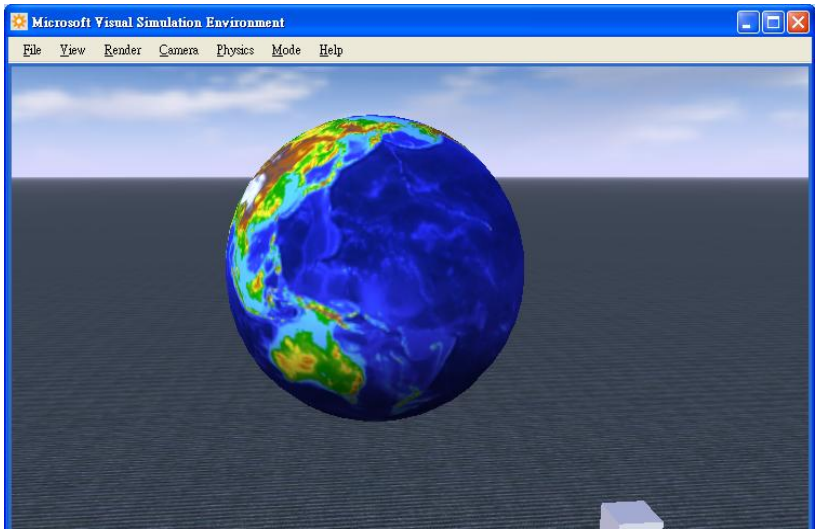


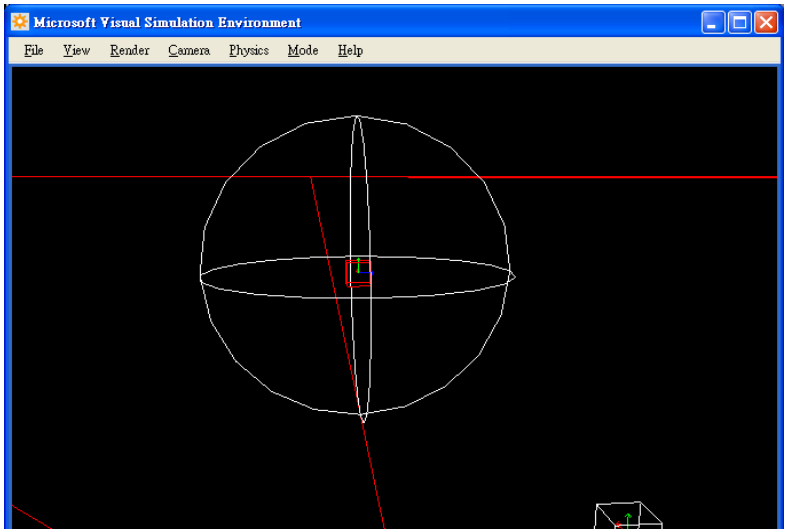


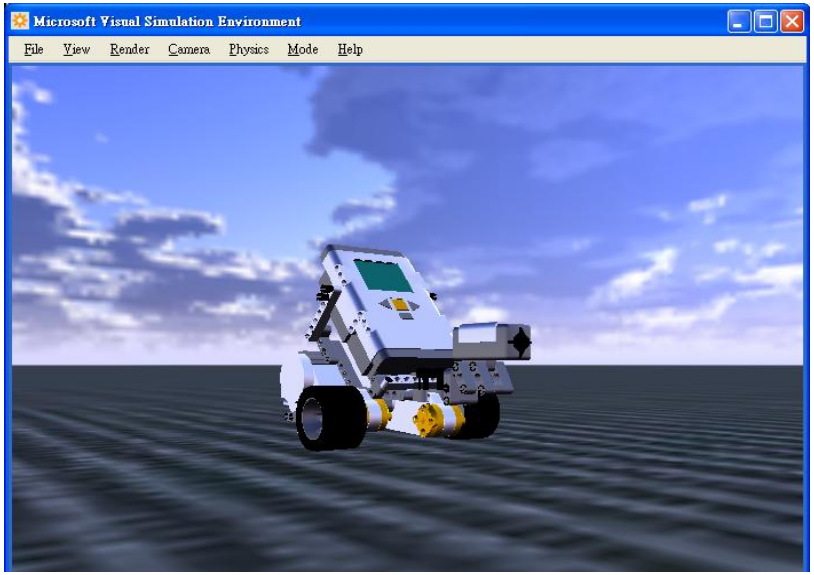
Benefits of Simulation

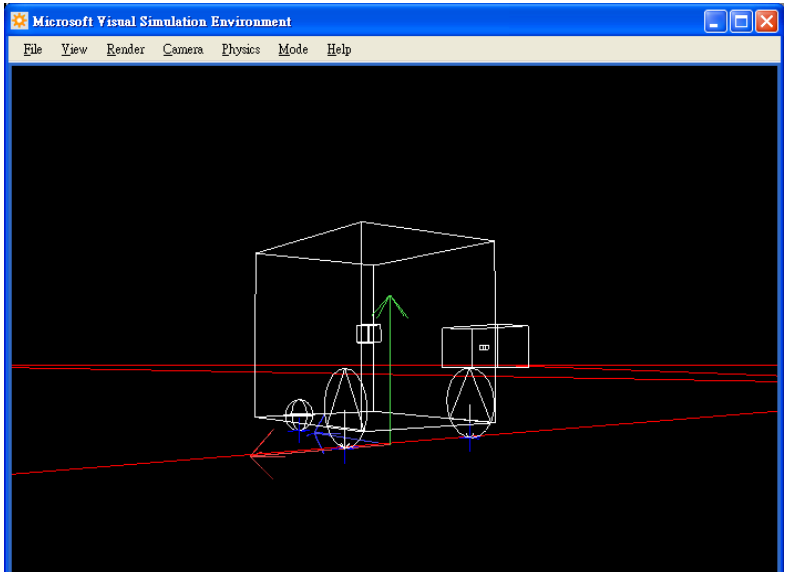
- **Low barrier to entry**
- **Staged approach**
- **Prototyping**
- **Education**
- **Learning system**

Snapshots


Microsoft Visual Simulation Environment
File View Render Camera Physics Mode Help


Microsoft Visual Simulation Environment
File View Render Camera Physics Mode Help


Microsoft Visual Simulation Environment
File View Render Camera Physics Mode Help

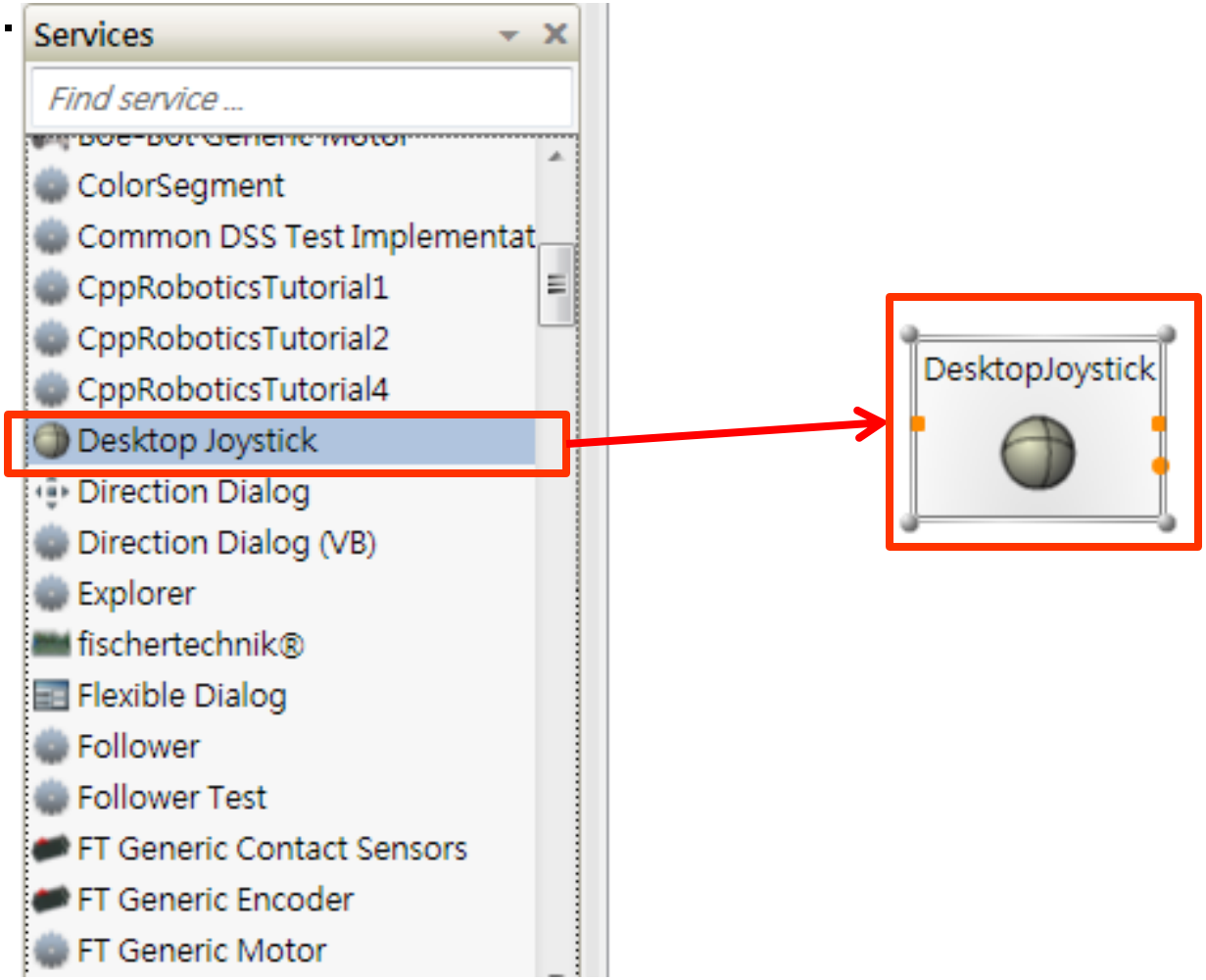

Microsoft Visual Simulation Environment
File View Render Camera Physics Mode Help

Demo

(II) Joystick a Robot in Simulation

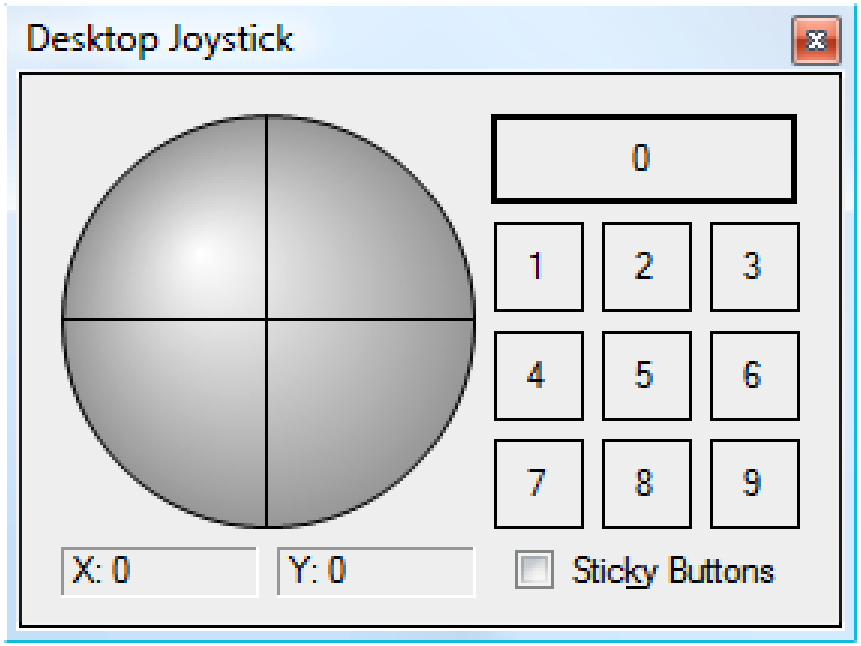
Step 1 Start the Program

- Drag the **Desktop Joystick** service onto your workspace.



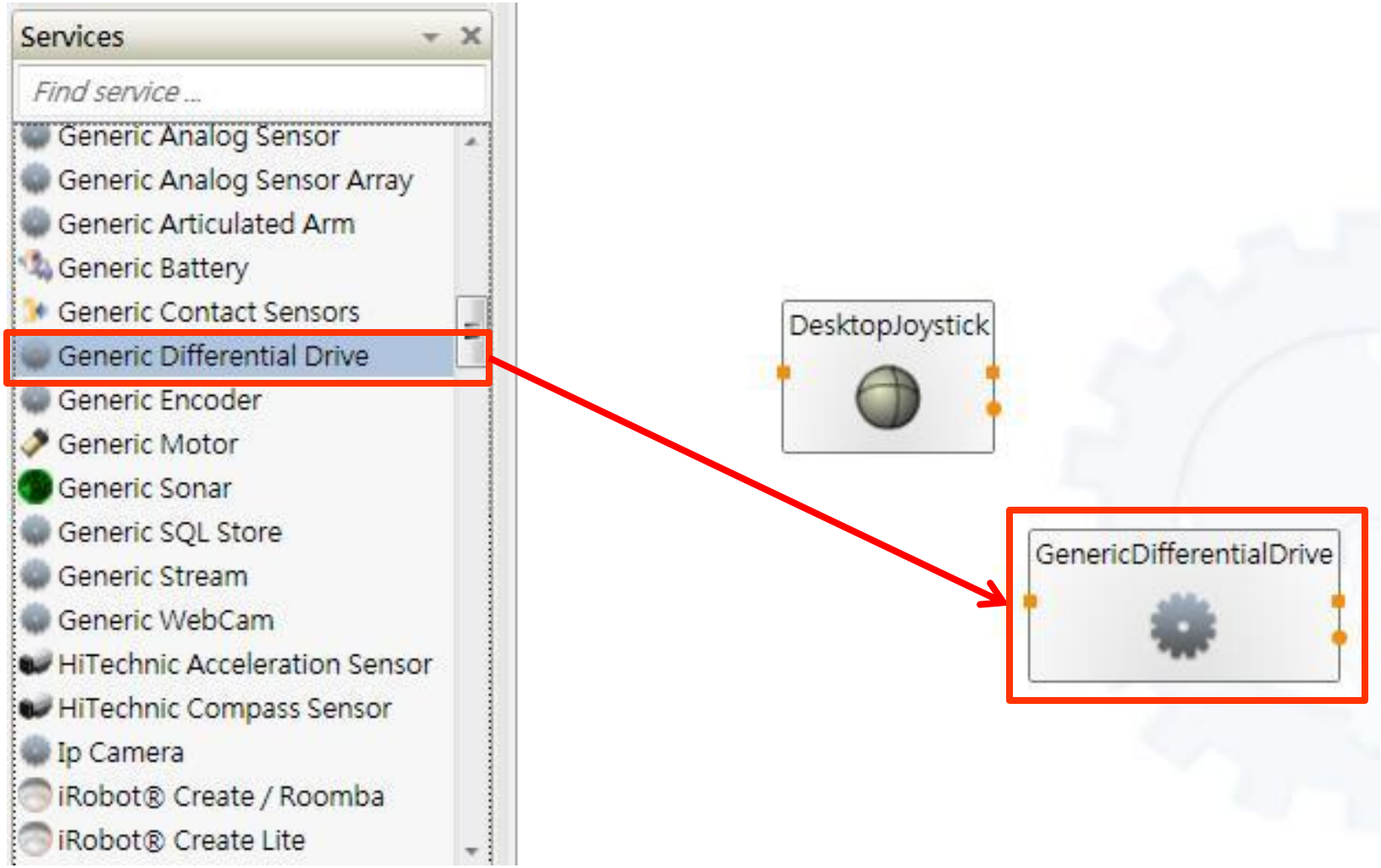
Step 2 Test

- Try it out !



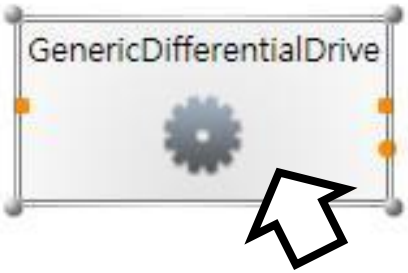
Step 3 Add a Service

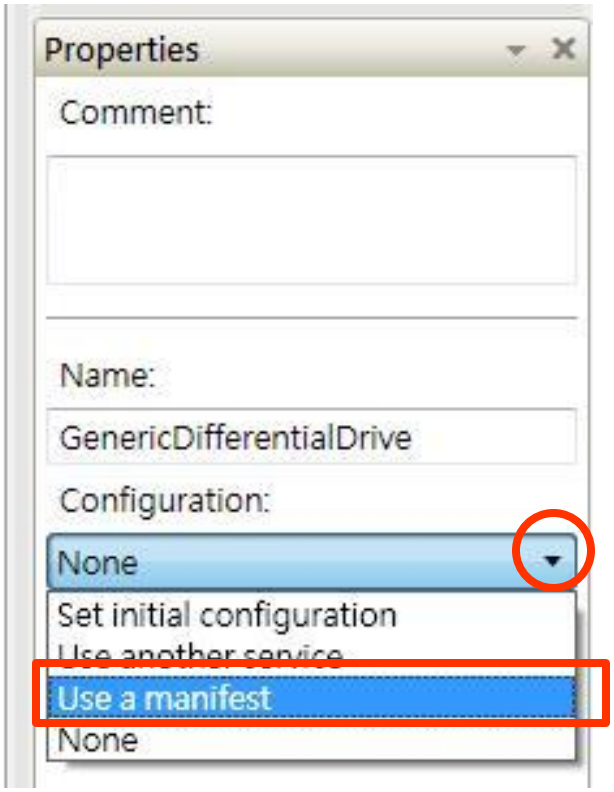
- Add a **Generic Differential Drive** service.



Step 4 Set Up the Property

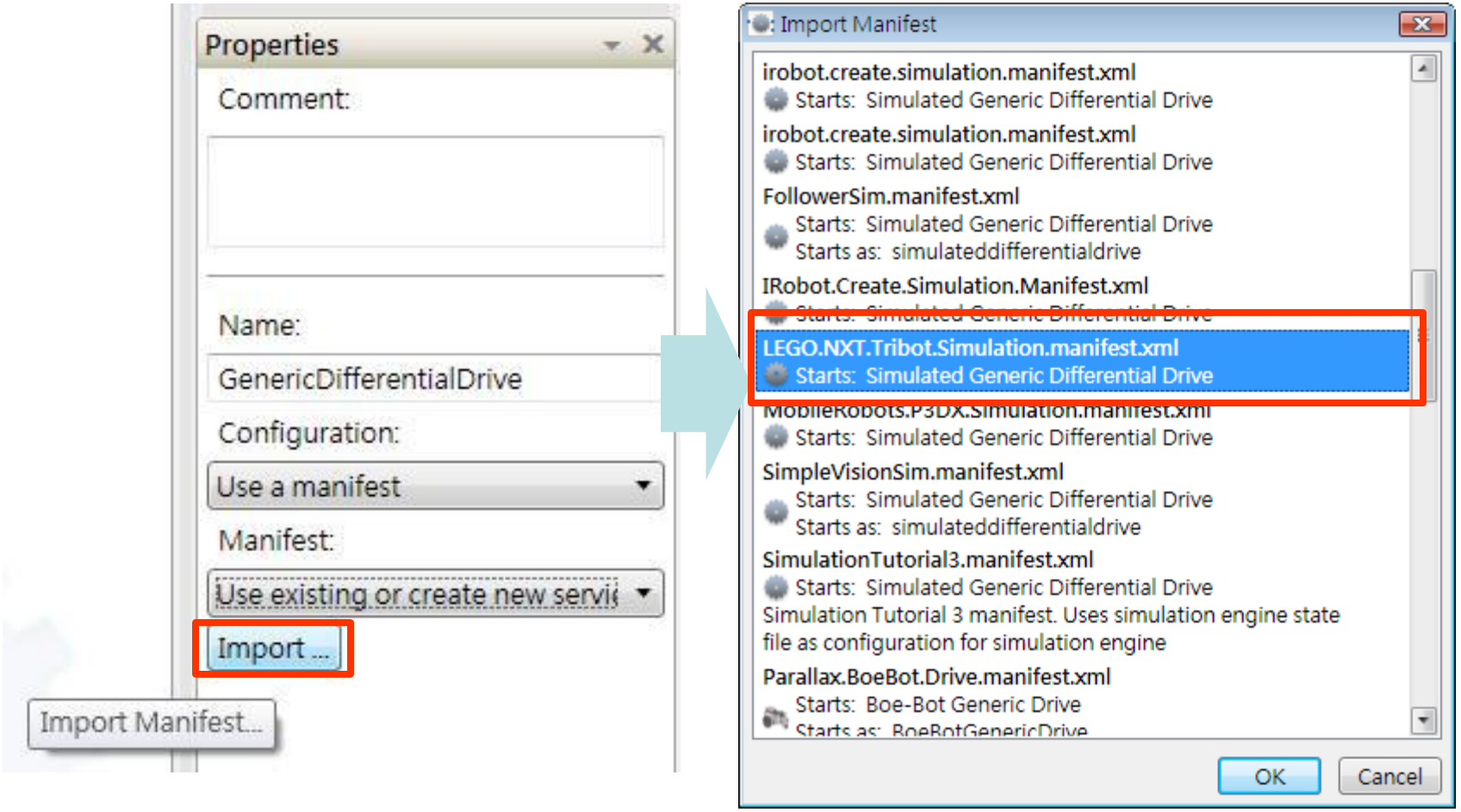
- First click on the **GenericDifferentialDrive** box; Then, in the **Properties** on the right, for **Configuration**, select the **Use a Manifest** entry.





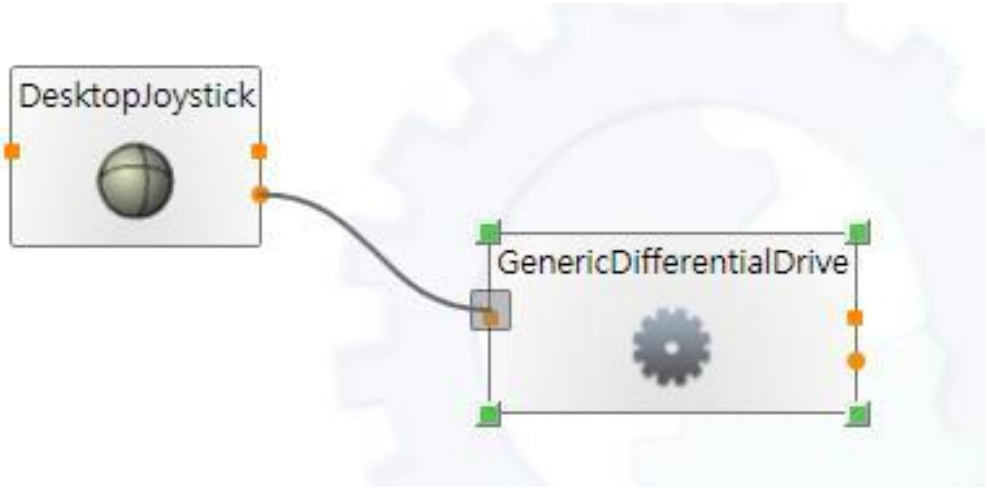
Step 5 Using a Manifest File

- Select the entry for **LEGO.NXT.Tribot.Simulation.Manifest.xml**.



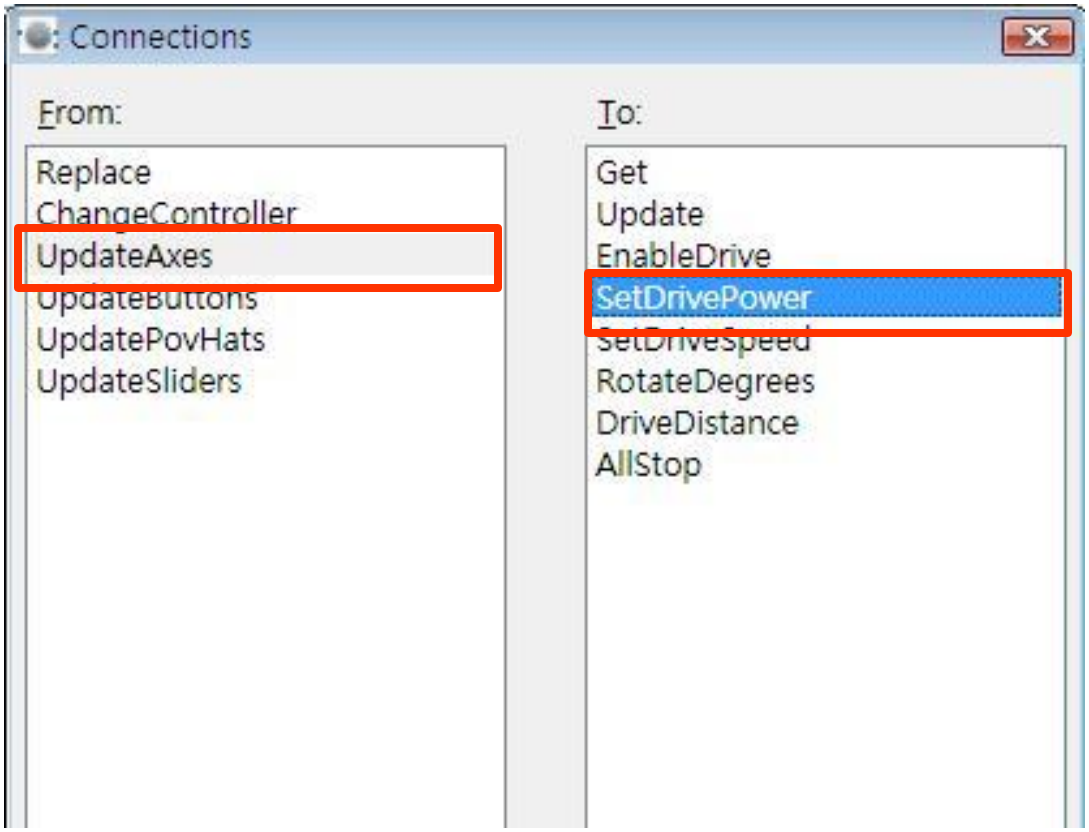
Step 6 Connect

- Connect the notification port (the small red circle on the right) of the **DesktopJoystick** to the input port (the small red square on the left) of the **GenericDifferentialDrive**.



Step 7 Set Up the Connection Type

- In the **Connections** dialog box, select **UpdateAxes** from the left column and **SetDrivePower** in the right column, then click **OK**.



Step 8 (1/3) Transformation Rule

- The **DesktopJoystick** service is sending notifications when the joystick axes change (in this case the X and Y values change), as you may have noticed when trying out the diagram earlier, These values vary between -1000 and 1000.
- The **GenericDifferentialDrive** service is expecting **LeftWheelPower** and **RightWheelPower** values between -1 and 1, where these values are indicative of the amount of power to be fed to the Left and Right wheels respectively.
- You now need to create a data transformation that will convert the X and Y values from the **DesktopJoystick** into **Left-** and **RightWheelPower**.

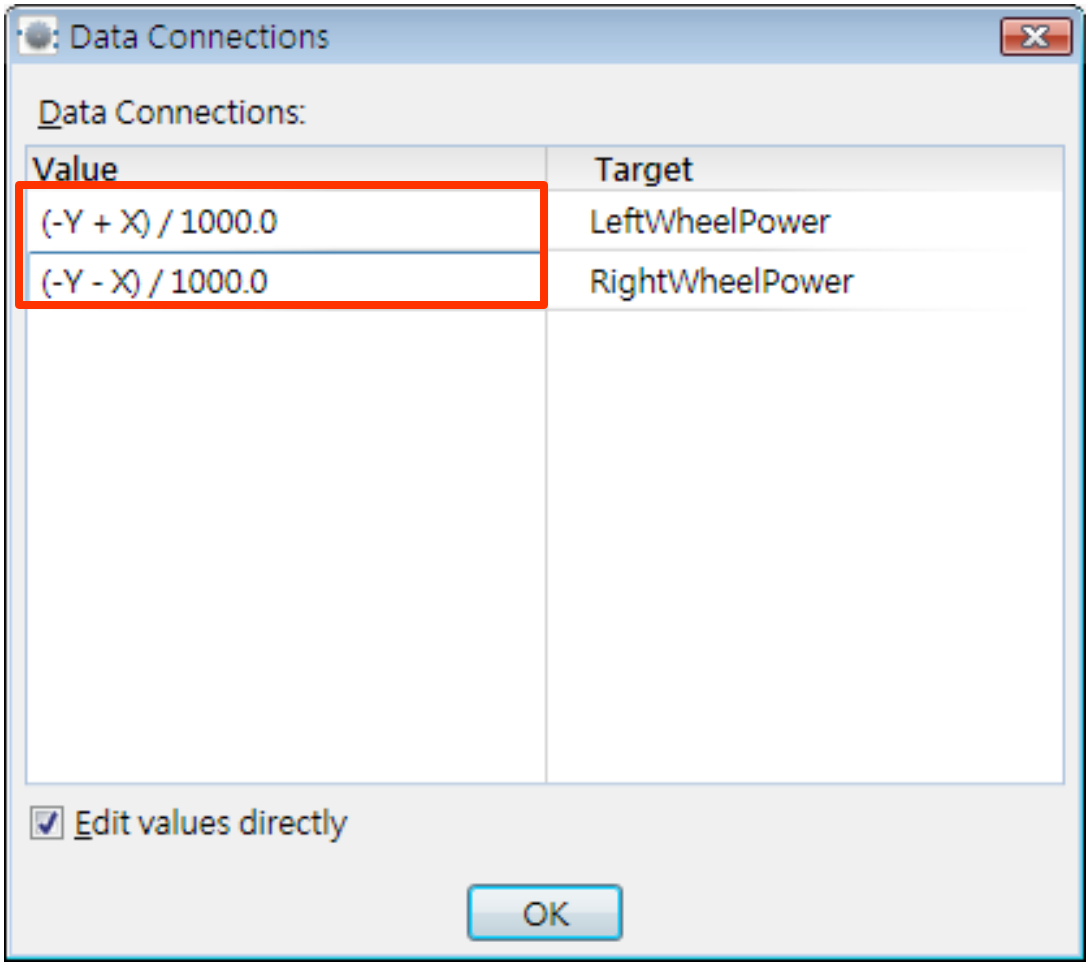
Step 8 (2/3)

- Select the **Edit values directly** check box in the **DataConnections** form. Then enter the following expressions into the value boxes for the **LeftWheelPower** and **RightWheelPower** targets.

(-Y + X) / 1000.0
(-Y – X) / 1000.0

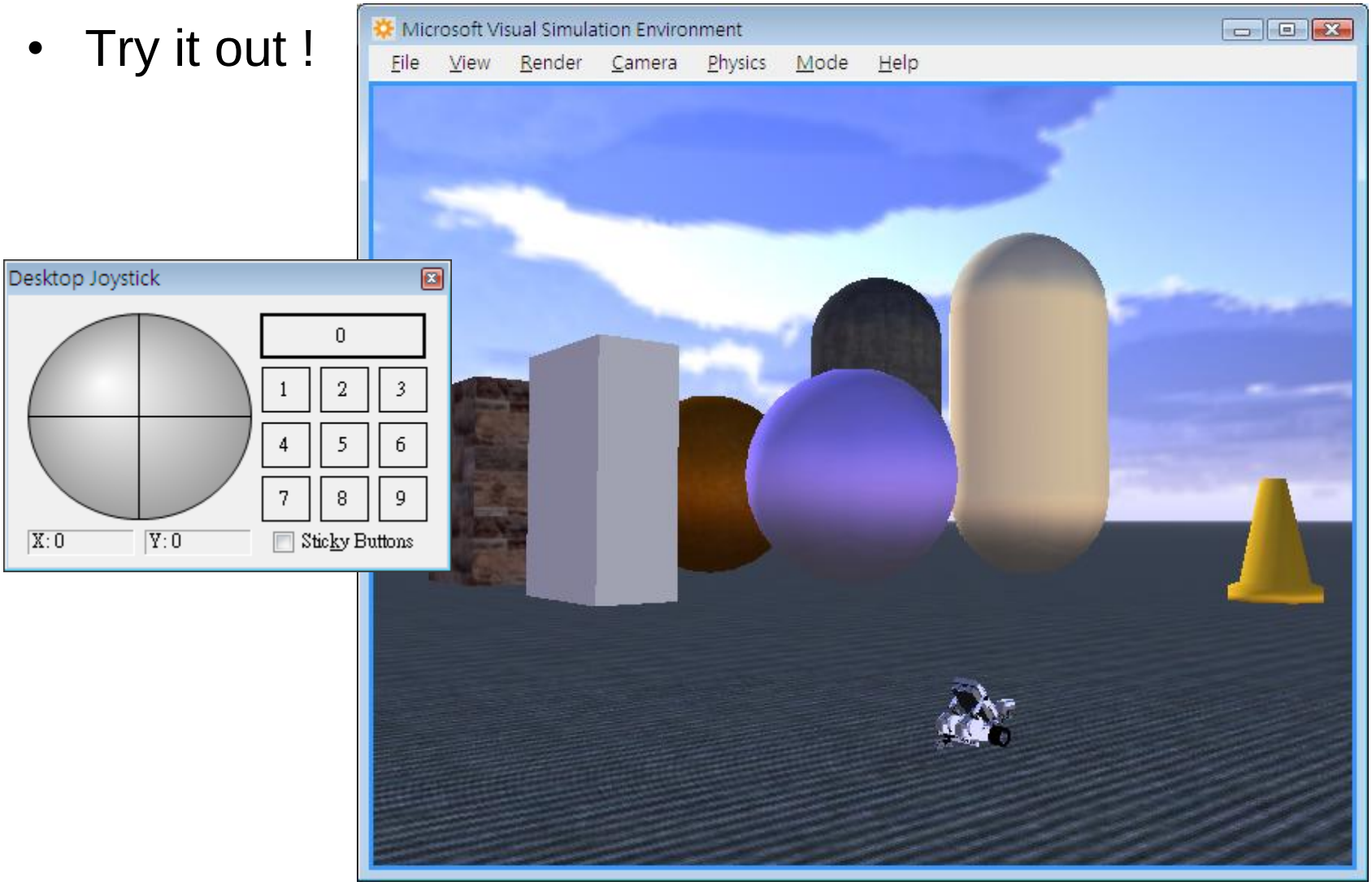
Step 8 (3/3)

- Like this !



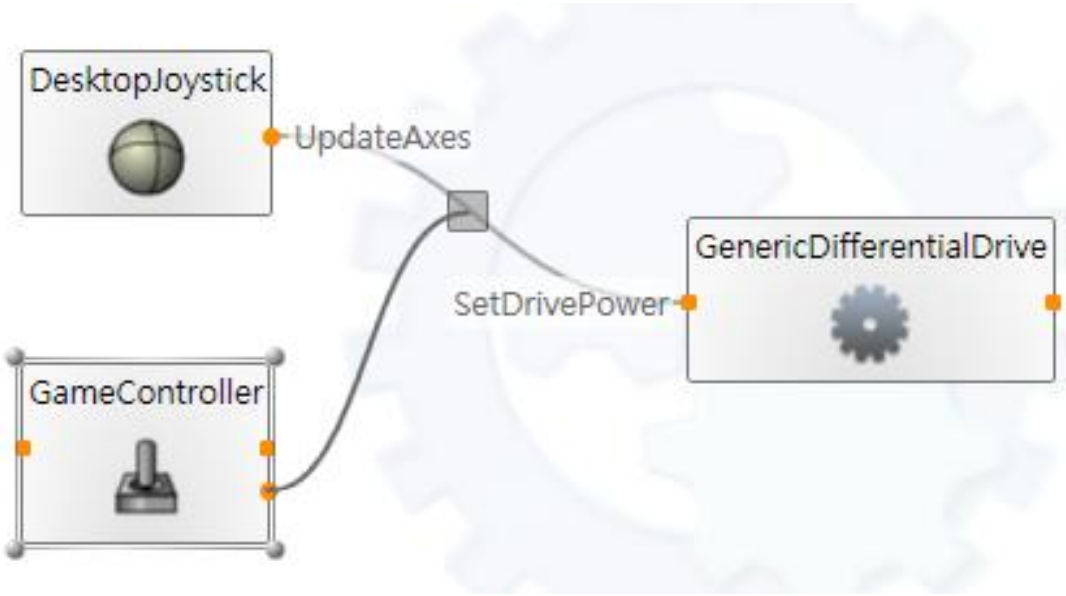
Step 9 Test

- Try it out !



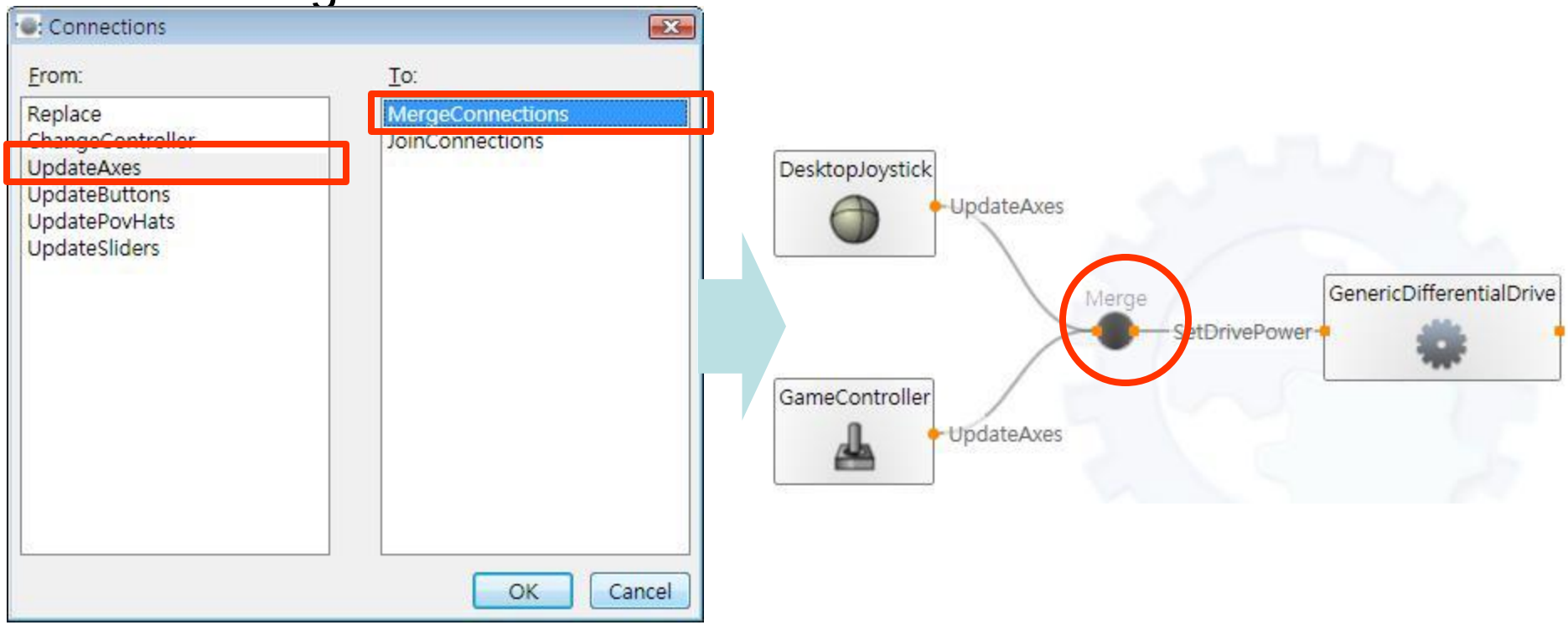
Step 10 Using a Joystick (1/2)

- Drag a **GameController** service from the **Services** toolbox. Then drag from the notification pin of the **GameController** service to the middle of the wire connecting the **DesktopJoystick** and **GenericDifferentialDrive** services.



Step 10 (2/2)

- Release it and on the left side, as before, choose **UpdateAxes**, on the right side choose **MergeConnection**.
- Run again !

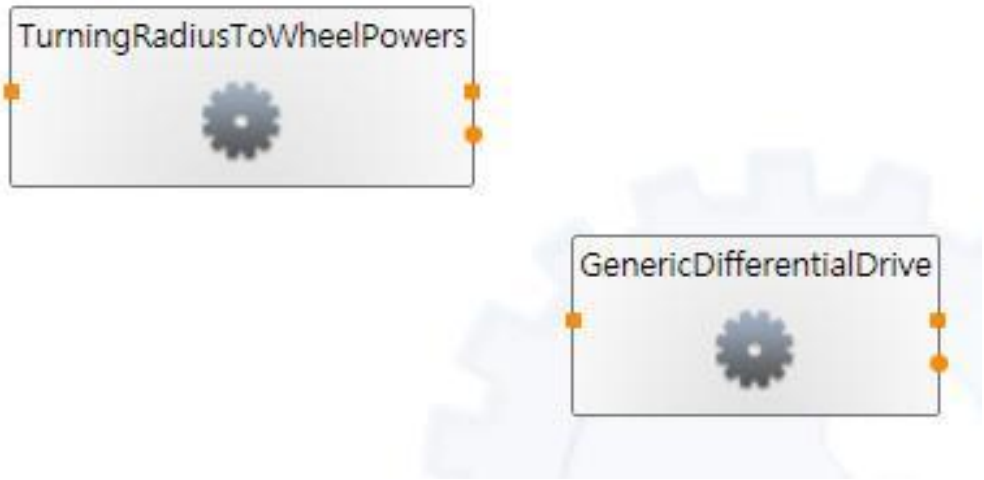


Result

(III) Drive in Circles

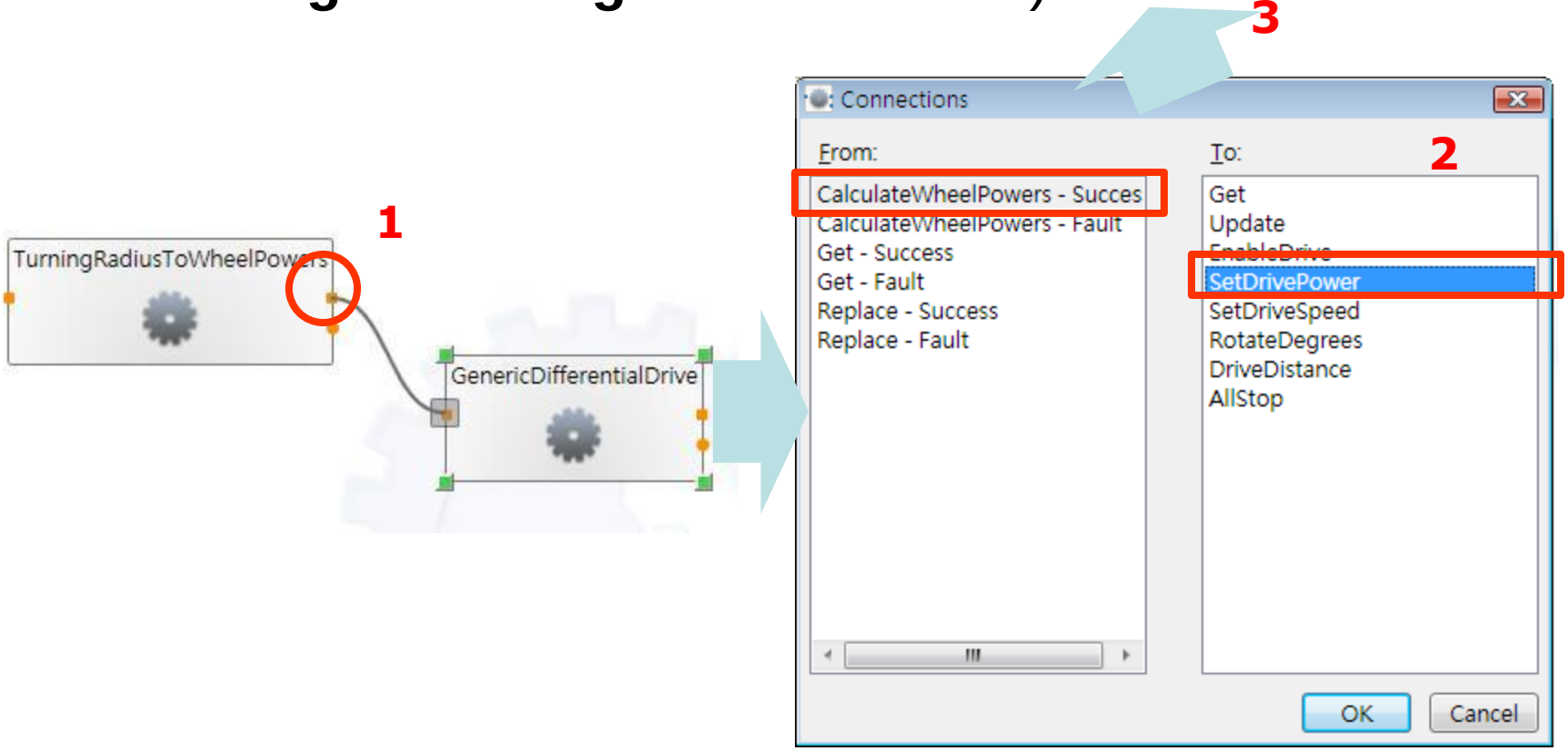
Step 1 Start the Program

- Start by adding a **TurningRadiusToWheelPowers** service and a **GenericDifferentialDrive** service.



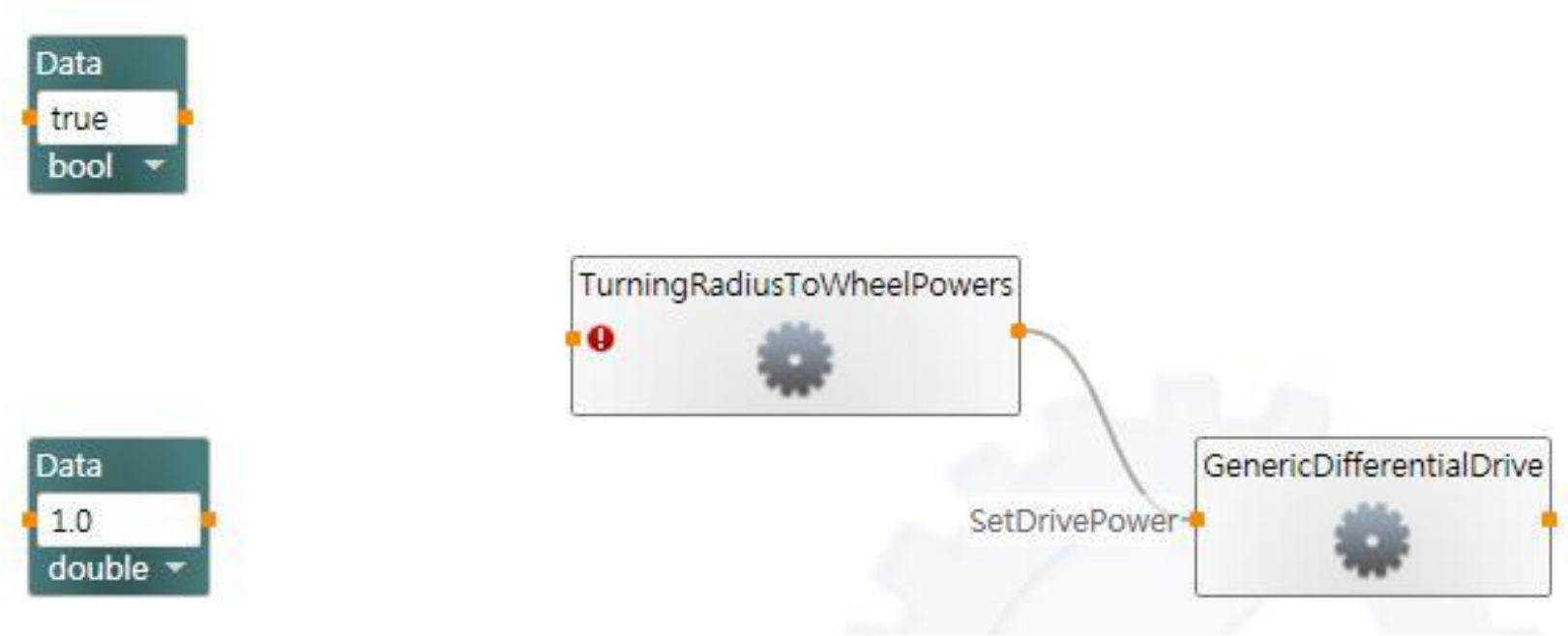
Step 2

- Connect the output port of the former to the input port of the latter (associating **Left** with **LeftWheelPower** and **Right** with **RightWheelPower**).



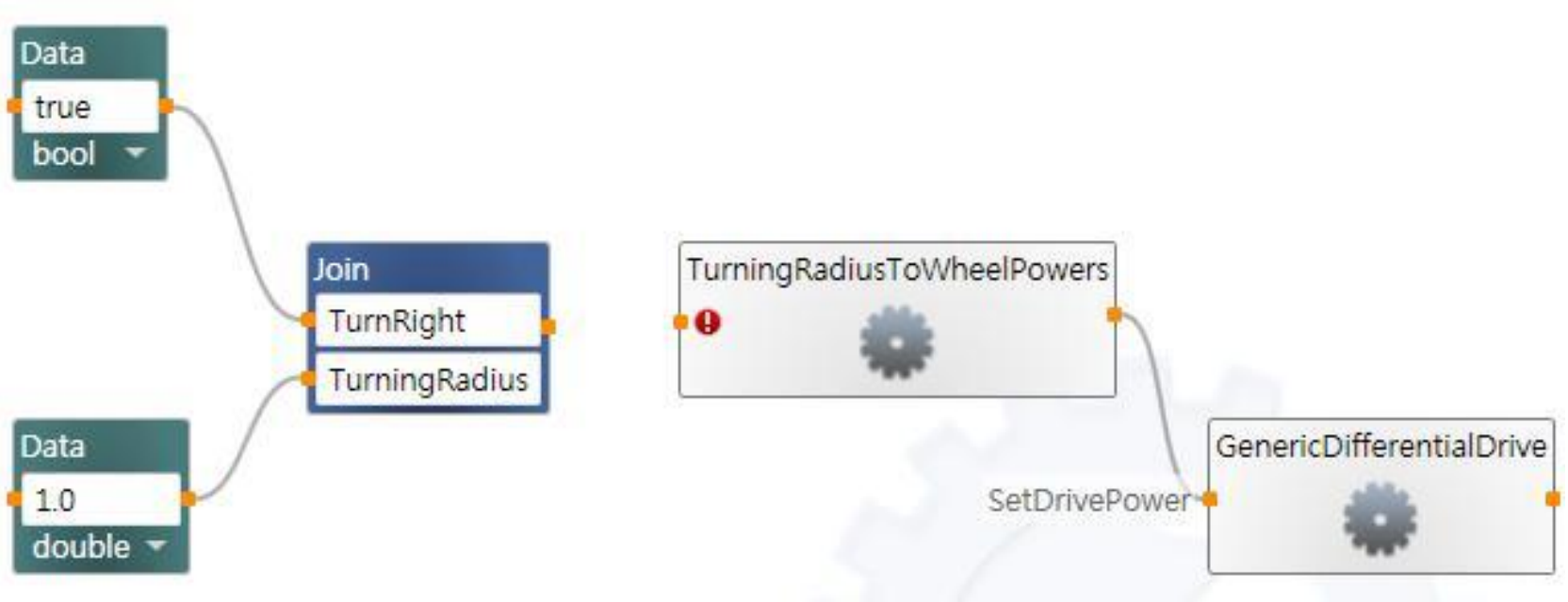
Step 3

- Drag over two **Data** and set one block to be of type **boolean** with value **true**, and the other to be of type **double** with value **1.0**.



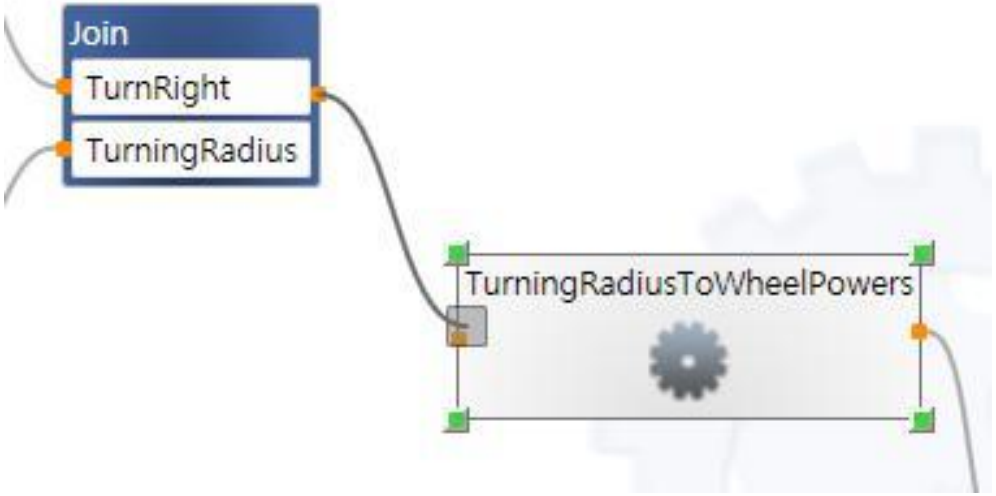
Step 4

- To aggregate the two pieces of data, we must add a **Join** block.
- Connect the **boolean** data to the **Join**, specifying the name **TurnRight**; also connect the **double** data to the **Join**, calling it **TurningRadius**.



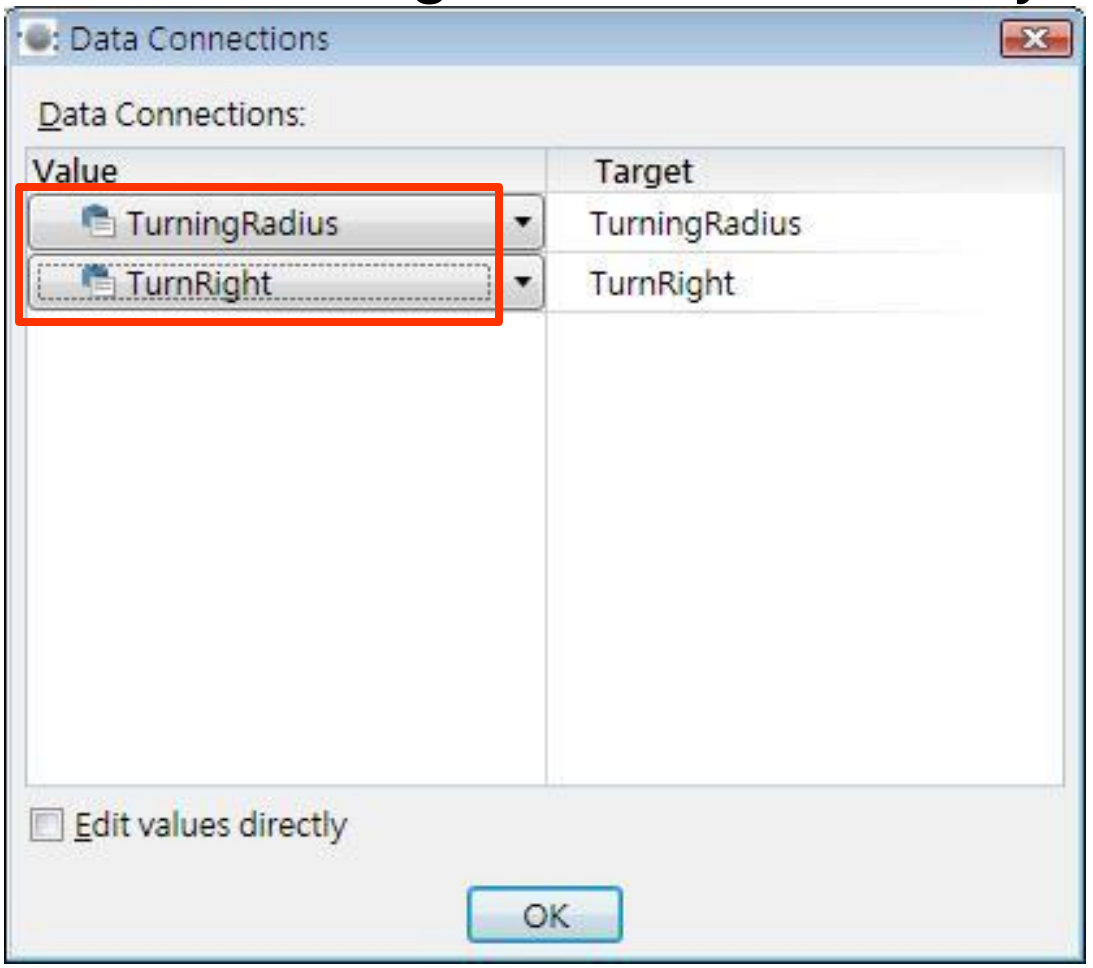
Step 5 (1/2)

- **TurningRadiusToWheelPower** takes in two arguments: the **TurningRadius** (a **double**) and a **boolean**, **TurnRight**, which encodes whether or not to turn to the right (otherwise, turn to the left).
- Connect **Join** and **TurningRadiusToWheelPower**.



Step 5 (2/2)

- Associating **TurningRadius** with **TurningRadius** and **TurnRight** with **TurnRight**. Click **OK**, then try it out !



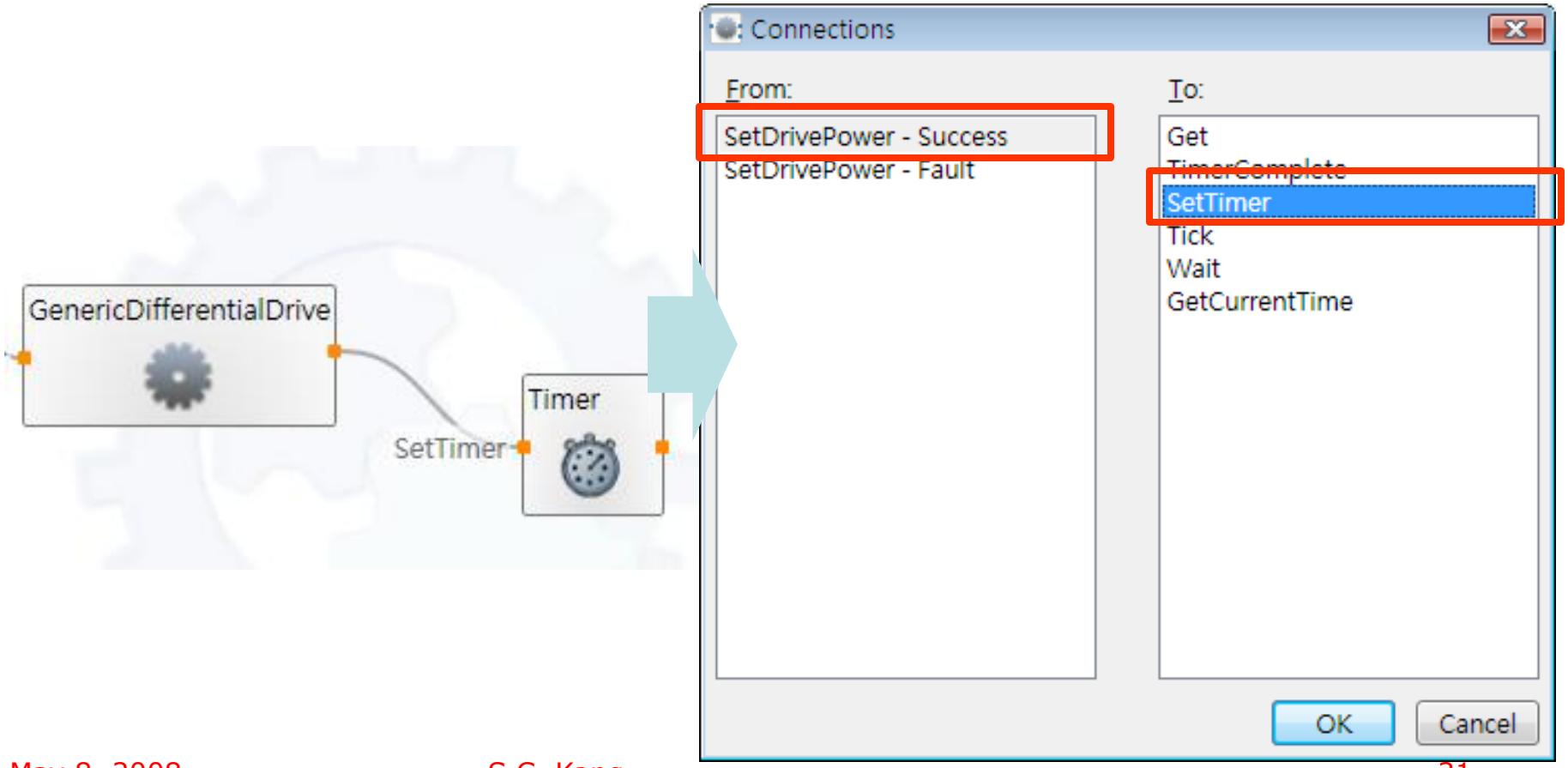
Result

- Your robot can now drive in circles. Vary the hard-coded values and watch its behavior.

(IV) Drive along an Arc

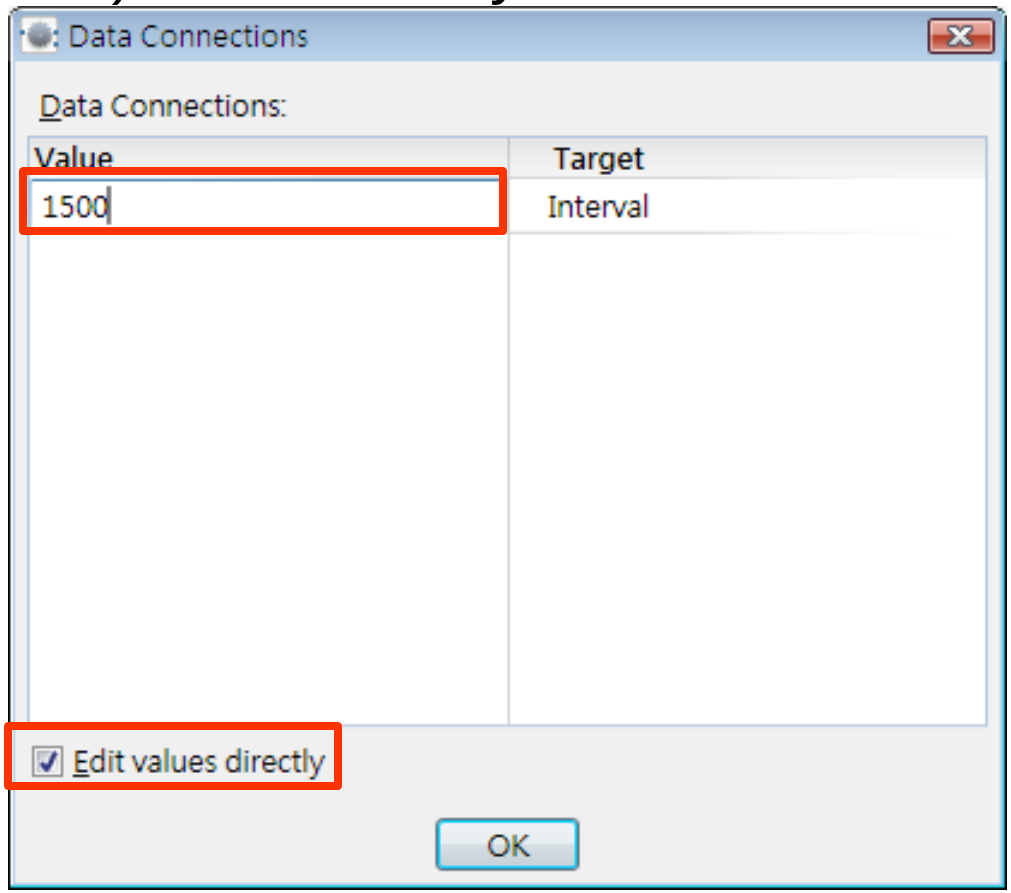
Step 1 Continues the Diagram (III)

- Add **Timer** service to your diagram and connect to **GenericDifferentialDrive** (associating the **SetDrivePowerSuccess** with the **SetTimer**).



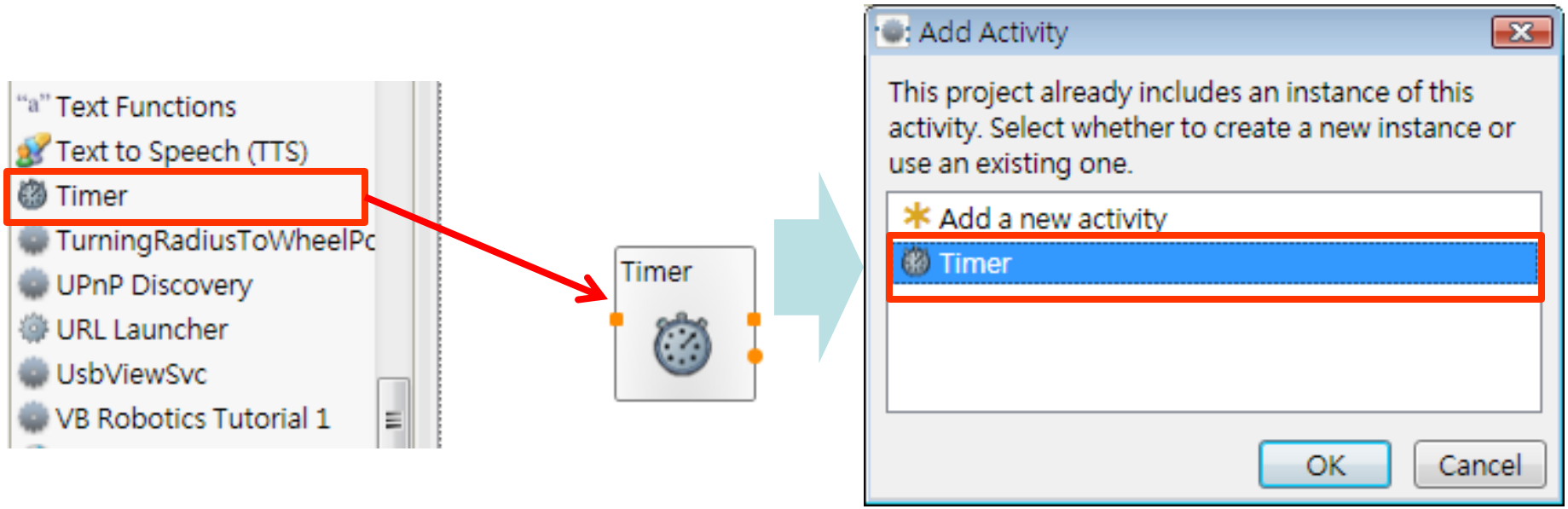
Step 2 Set Up Timer

- In the **Data Connections** window, check **Edit Values Directly** and type in **1500** for the value (the units are milliseconds).This link only indicates execution flow.



Step 3

- Drag in a new **Timer**, and in the **Add Activity** dialog that pops up, select the **Timer** that already exists instead of adding a new one.

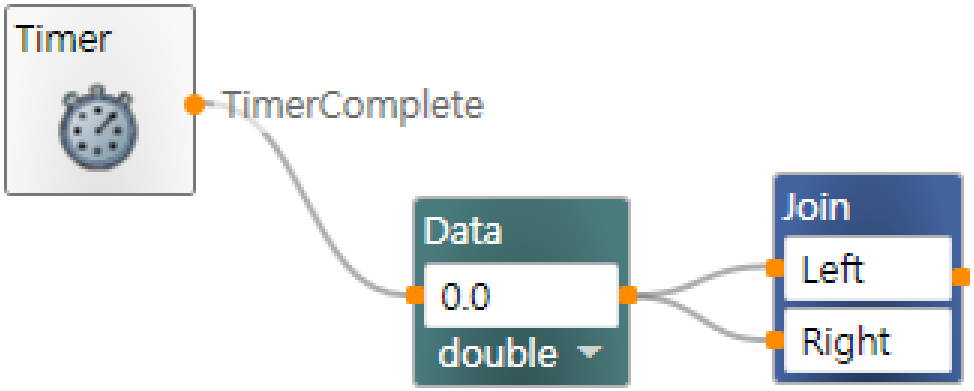


Step 4

- Connect the new **Timer**'s notification port to a new **Data** block with a **double** value of 0.0 (use the **TimerComplete** operation).
- This will set up the execution path so that when the timer completes (that was set on the right), execution will begin.

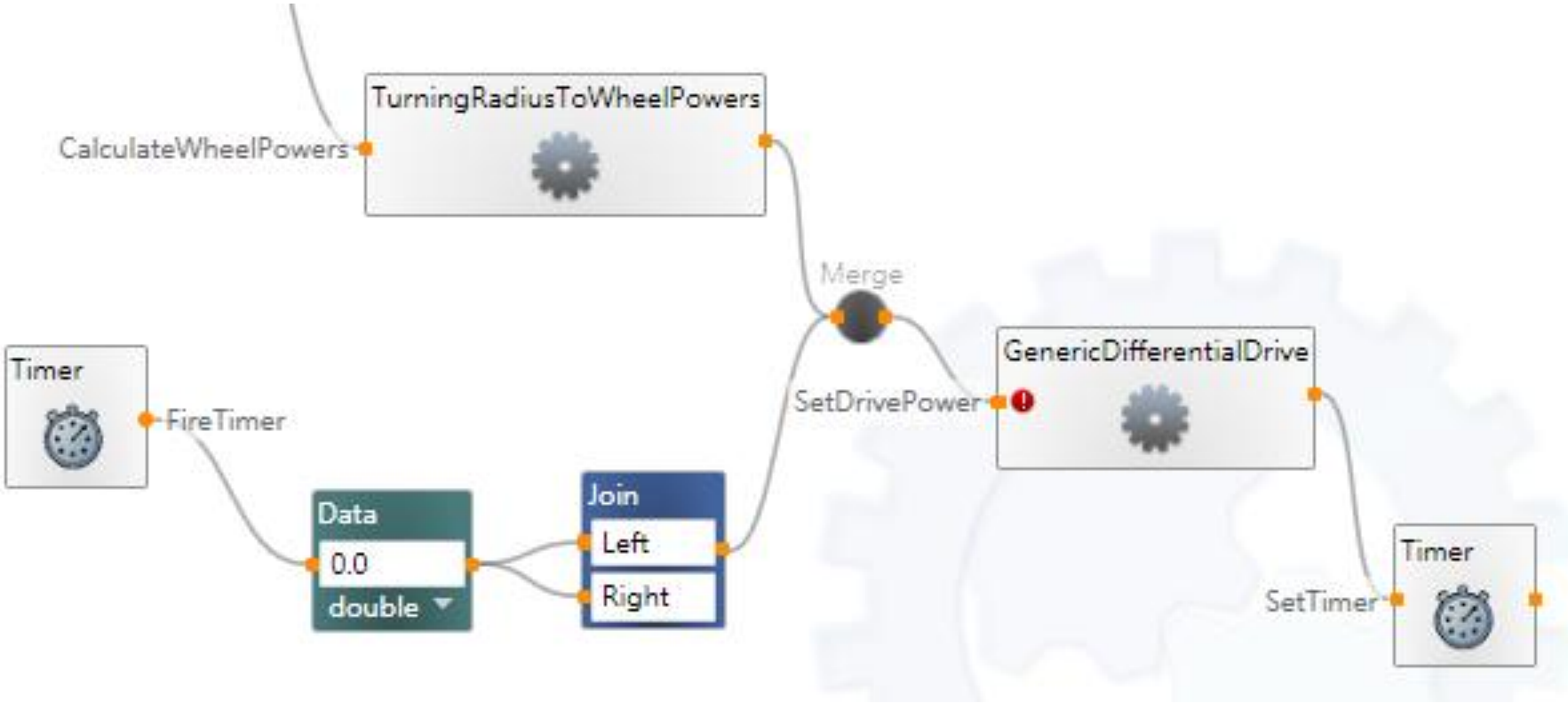
Step 5

- In order to stop, we must set both the left and right wheels to no power. Add a **Join** with names **Left** and **Right** and attach the output of the data value to both of them. They are the same as the names output by **TurningRadiusToWheelPowers**.



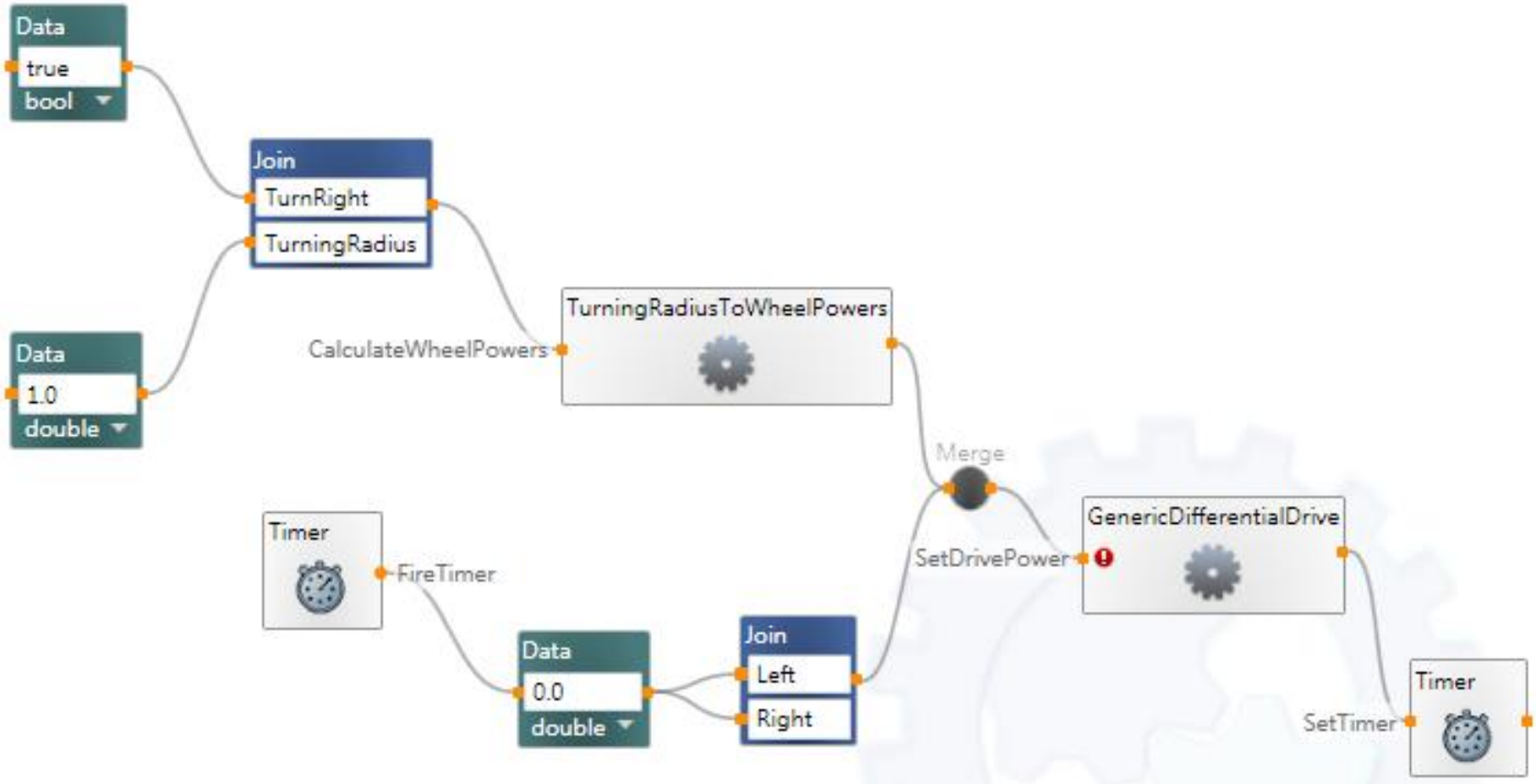
Step 6 Merge Connections

- Connect the output of the **Join** to the drive; This time, the middle man we need is **Merge**. It is necessary, however, for the inputs to **Merge** to match exactly, both in name and in type.



Step 7 Finish

- The overall diagram should be look like:



Result

- Your robot can now drive along an arc; again, vary the hard-coded values and watch its behavior.

Practice

- How about driving along the path like “8”?
- How about driving along rectangle path?

Question?

Course website

http://robot.caece.net