*Suggested Progression of Lesson*

|  |  |
| --- | --- |
| **Step** | **Content** |
| 1 | [Introduction](#_Introduction) |
| 2 | [Motors and Colors with MATLAB Functions](#_Motors_and_Colors) |
| 3 | [Study 3](#_Study_3) |
| 4 | [Stateflow Diagrams with Simulink](#_Stateflow_Diagrams_with) |
| 5 | [Proximity Lights with Motors (Obstacle Avoidance Robot)](#_Proximity_Lights_with) |
| 6 | [Study 4](#_Study_4) |
| 7 | [Simulink Dead Reckoning with Stateflow](#_Simulink_Dead_Reckoning) |
| 8 | [Taking It Further](#_Taking_It_Further) |

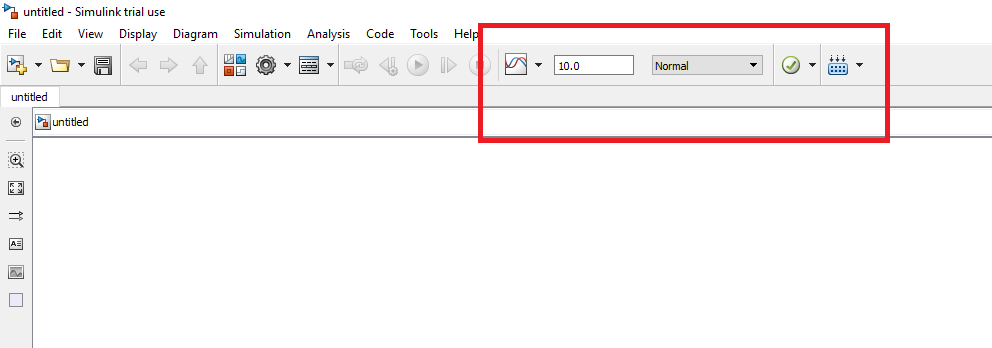
**Introduction**

*Setting Up External Mode:*

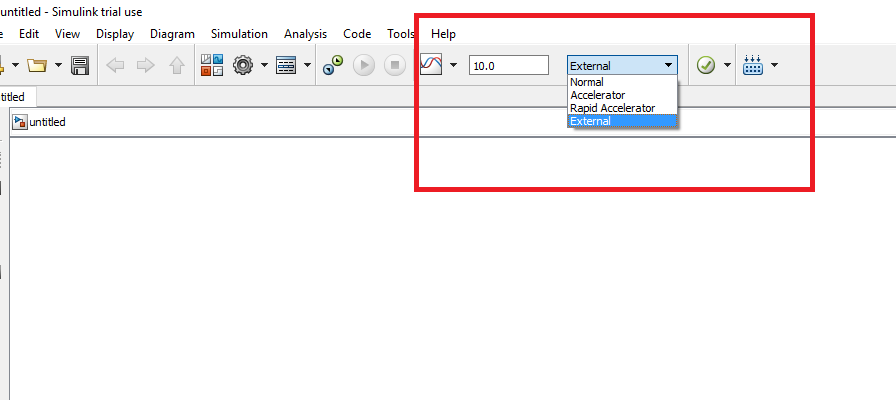
Video Reference: [Unit3a\_SettingUpExternalMode.mov](https://www.youtube.com/watch?v=CXeHnf8yrtU)

External mode allows us to run Simulink code continuously and view features of our model, such as “display blocks” change dynamically. This can be especially useful when testing the function of sensors on our miniQ robot.

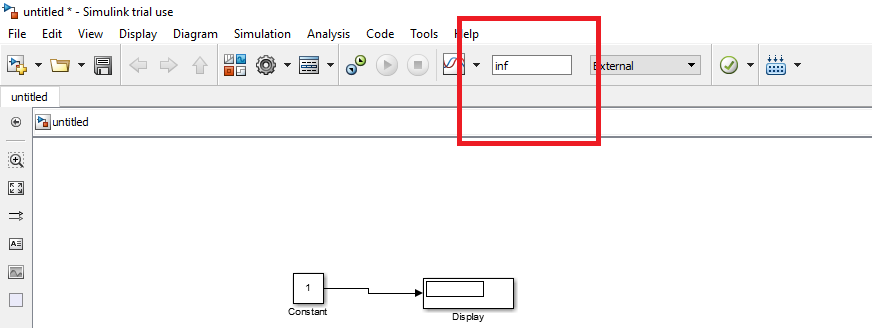
1. The red box highlights the region at the top of the Simulink interface where you can change the configuration parameters to external mode



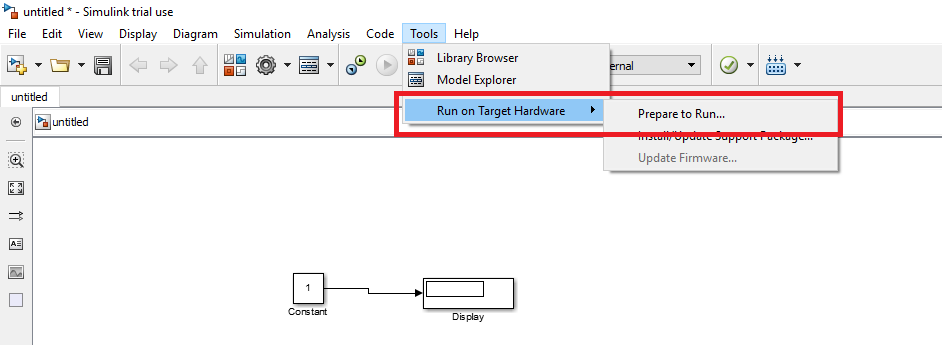
1. Select the drop down menu tab and select the option “External”



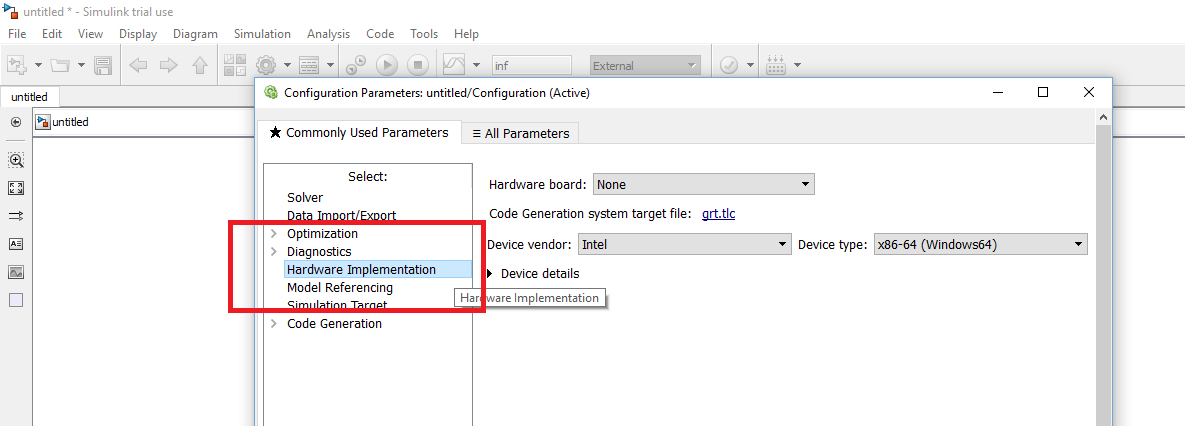
1. In the simulation time frame change the value “10.0” to “inf” as in the picture below. This will allow the simulation target time to run for an infinite period rather than just 10 seconds.



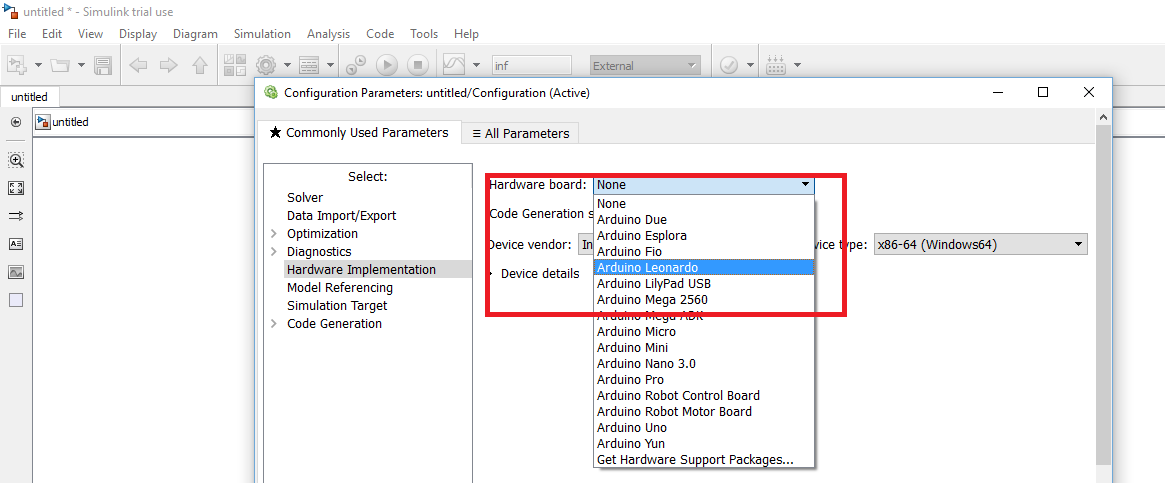
1. If you have not already done so, prepare you Simulink model to be deployed to the target hardware, the miniQ robot, by selecting the tools menu and then the “Run on Target Hardware” option. Click on “Prepare to Run”



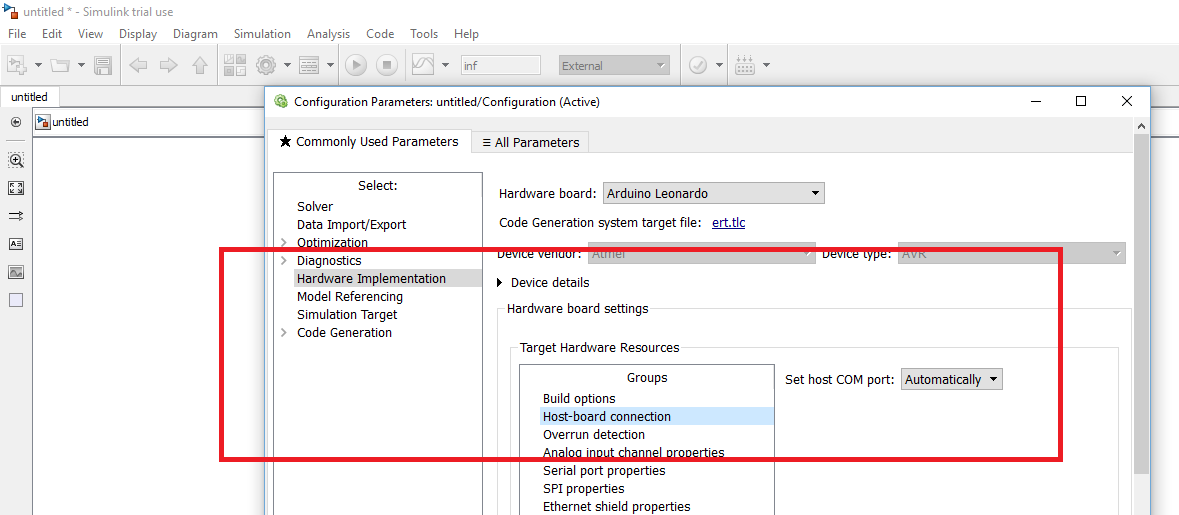
1. From the menu which pops up select “Hardware implementation”



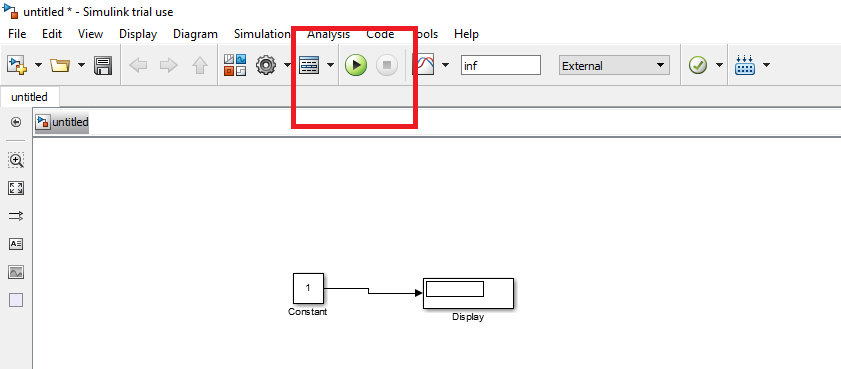
1. You will then need to select the hardware board being used. For our miniQ we are going to select the Arduino Leonardo option as seen in the picture below.



1. Make sure the right COM port is listed, you can have it set to “Automatically” but if you are having trouble deploying a model to hardware you will need to switch this tab to “Manual” and then write in the specific COM port value.



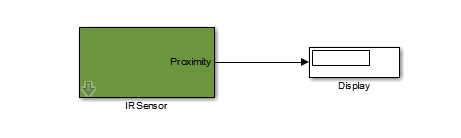
1. Once all the above steps are setup then green play button, highlighted in the red box below will, be shown (previously this button is a light gray). To run external mode click this green play button, the stop button next to it will be useful for stopping the simulation.



**Motors and Colors with MATLAB Functions**

*Run Initial Tests in External Mode:*

Students should consider running the following model in external mode to figure out the sensitivity of the IR sensor before moving into the “Color Machine” example:

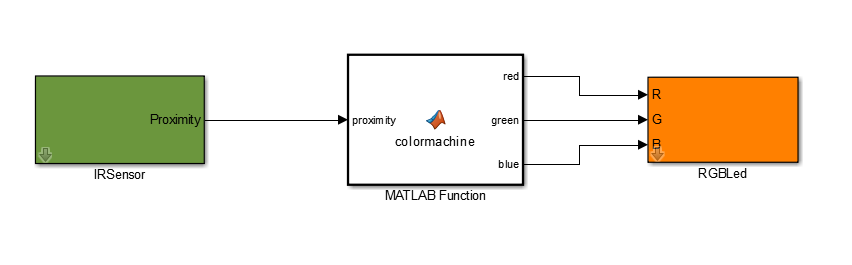


*“Color Machine”*

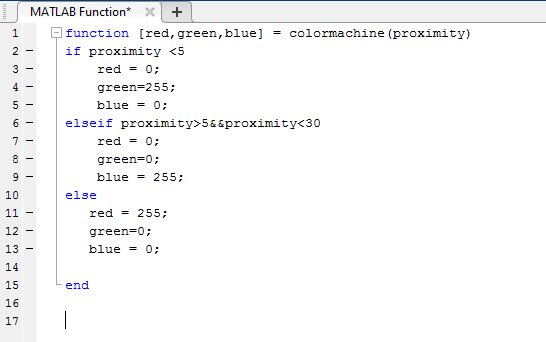
Example Model: ColorMachine.slx

Video Reference: [Unit3b\_SimulinkFunctionsWithMotors.mov](https://www.youtube.com/watch?v=bupeRE5uw04)

The Simulink model below takes in a proximity sensor reading and converts it into a specific combination of red, green and blue output values:



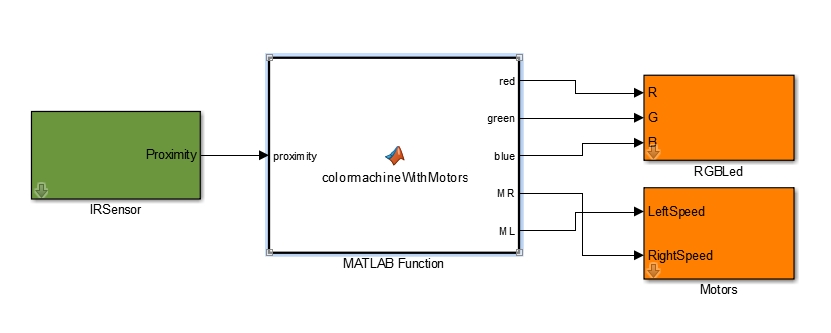
The function is structured as follows and notice the use of “if”, “elseif” and “else” control structures:



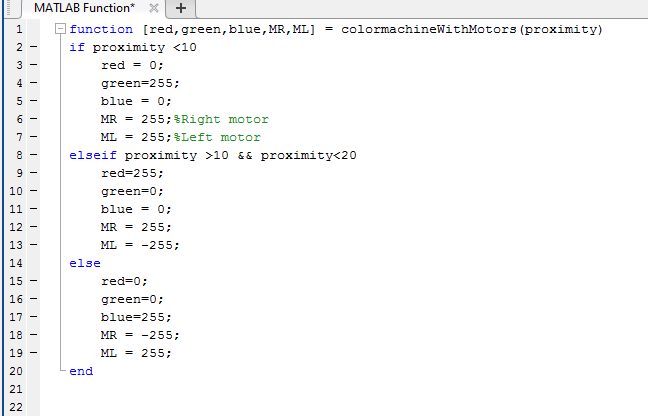
*“Color Machine With Motors (Obstacle Avoidance Robot)”*

Example Model: ColorMachineWithMotorsFunction.slx

The color machine can be expanded upon so that distance is also used to control motor direction. Here we have the begins of an obstacle avoiding robot which also outputs different colored lights:



The function for this model has been expanded so there are additional output values “MR” and “ML”:



*Activity:*

With StateFlowRGBBlendWithProximity.slxloaded onto your miniQ change some of the variables to improve the obstacle avoidance routine.

**Study 3**

*Note: Below is a copy of Study 3. For print-outs, the original document can be found in the Unit 3 folder*

Standards Covered: (a), (b), (c)**,**(d)**,**(e), (f)

Read the following material:

1. <http://discovermagazine.com/2013/may/14-bug-inspired-robots-designed-to-do-our-dirty-work>
2. <http://www1.appstate.edu/~kms/classes/psy5150/Documents/Braitenberg1984.pdf>

Discuss: Split into 4 groups, evaluate the article and discuss as a group. What do you find interesting *(20 min*)?

Watch: Engineering The Ultimate Robotic Fish <https://www.youtube.com/watch?v=9ISGXe0Vl5A>

Discussion *(20 min):*

1. How do you define artificial intelligence?
2. Could the robotic fish in the video be considered a form of artificial intelligence?
3. How about from the perspective of an actual fish- is the robot real?
4. When we talk about “artificial intelligence” what personality attributes are we giving to a machine?
5. When our miniQ robots are running their obstacle avoidance programs, StateFlowRGBBlendWithProximity.slx**,** how can the personality of the robot be described?
6. How about when the robots interact with each other in a closed space; do they have a personality or do they remind you of any animal in particular?

Activity *(20 min)*:

1. What can be changed in the robot’s programming to effectively give it an alternative personality/or represent a different animal?
2. Identify an animal or character trait you want your robot to represent when running ColorMachineWithMotorsFunction.slx.
3. Change the Simulink code to represent this animal and then share your work with a partner.

**Stateflow Diagrams with Simulink**

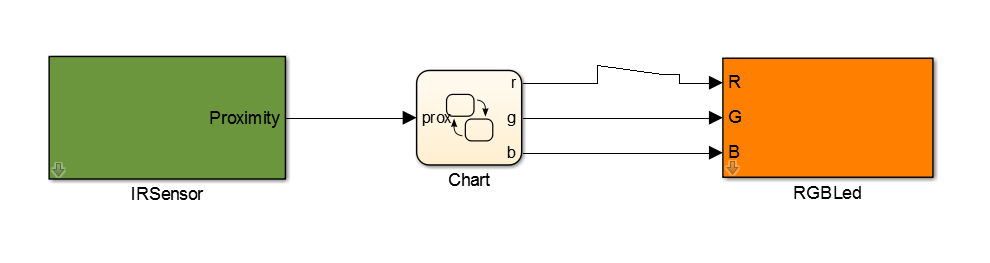
*For more information, reference the “Instructor Reference Lesson 1 Unit 3” document in the Unit 3 Instructor Reference folder*

Video Reference: [Unit3c\_SimulinkChartsWithMotors.mov](https://www.youtube.com/watch?v=bupeRE5uw04)

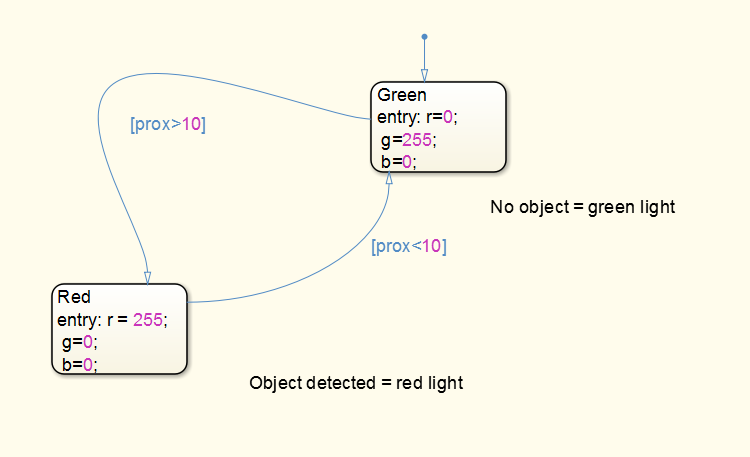
The example below expresses the “Color Machine” code from Unit 2 in an alternative MATLAB product, Stateflow, which represents code in the form of graphical “charts.” There will be times when internal processes or behavior for your robot are often better represented by the graphical, rounded, charts with connections flowing between them:

*“Proximity Simulink With Lights:”*

Example Model: StateFlowRGBBlendWithProximity.slx



1. Open the example model above
2. Double click into the “chart” block to view the following:



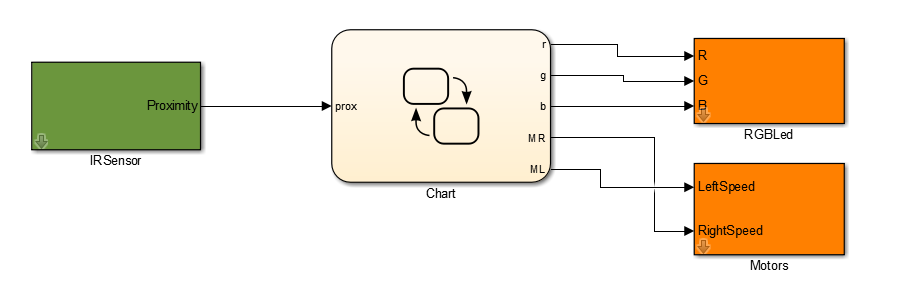
*Explanation*: The chart flows to different combinations of red, green and blue depending upon the proximity readings from the IR sensor block. The expression of code in a flow chart rather than a function helps aid the process of representing the robot’s behavior.

The code from unit 2, “Color Machine with Motors” can also be expressed in the form of a chart as seen below. Look to compare the formatting of a function with the example below.

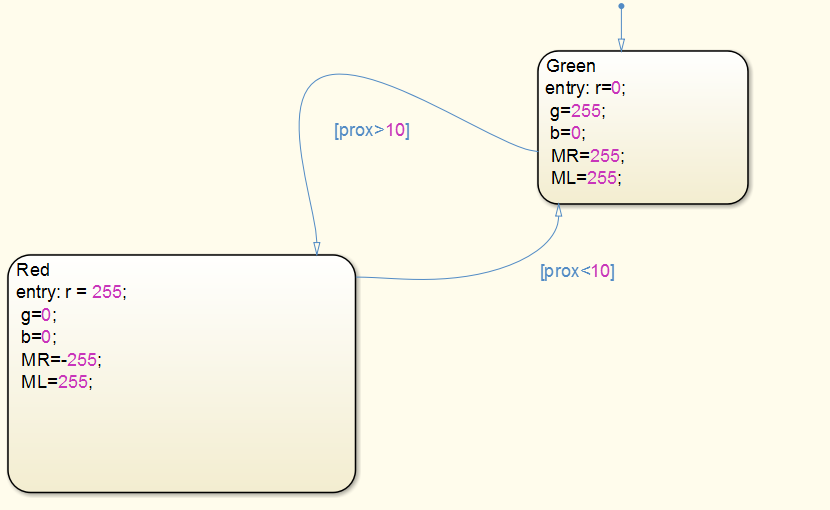
**Proximity Lights with Motors (Obstacle Avoidance Robot)**

*For more information, reference the “Instructor Reference Lesson 2 Unit 3” document in the Unit 3 Instructor Reference folder*

Model Index: StateFlowRGBledWithProximityMotors.slx



Create the above model in your Simulink platform. The chart should be expanded upon to create an obstacle avoiding robot which also has lights which signal specific types of movement:



Experiment with changing the values of prox and r,g,b values as well as the speeds for MR and ML in the charts.

**Study 4**

*Note: Below is a copy of Study 4. For print-outs, the original document can be found in the Unit 3 folder*

Standards Covered: (a), (b), (c)**,** (d)**,** (e), (f)

Sequence:

1. Discussion: What are some environments where we may see robots in the future? What kinds of tasks do you think they will have? *(10 min)*
2. In groups discuss the following article: *(25 min)* [https://timeandnavigation.si.edu/navigating-air/challenges/overcoming- challenges/dead-reckoning](https://timeandnavigation.si.edu/navigating-air/challenges/overcoming-challenges/dead-reckoning)
3. Elect a member of the group to share one interesting thing you read or a criticism/question you have about the article *(10 min):*
4. Rotate members of each group. Read the article and discuss *(25 min):*  <http://www.popsci.com/uber-parking-lot-patrolled-by-security-robot#page-2>
5. Elect a member of the group to share one interesting thing you read or a criticism/question you have about the article *(10 min)*:
6. Watch: Valet Parking Robot <https://www.youtube.com/watch?v=q3E7OG1JSKU>
7. Group Discussion *(20 min):*
8. Can you make a connection between how dead reckoning programs could be used for these robots?
9. What interesting ideas for robots can you think of from watching the video and from the articles you read?

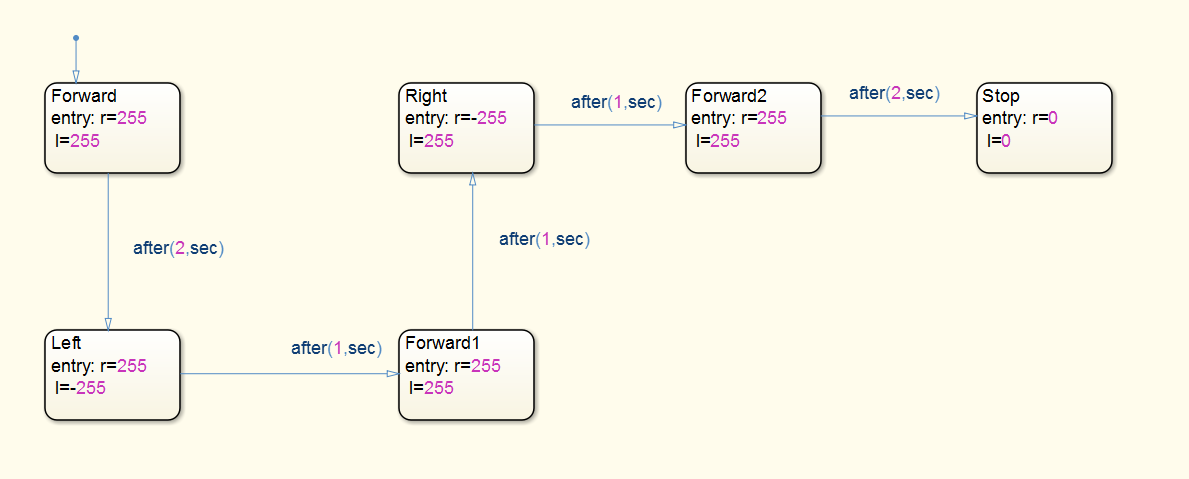
**Simulink Dead Reckoning with Stateflow**

*For more information, reference the “Instructor Reference Lesson 3 Unit 3” document in the Unit 3 Instructor Reference folder*

Example Model: SimulinkDeadReckoningWithCharts.slx

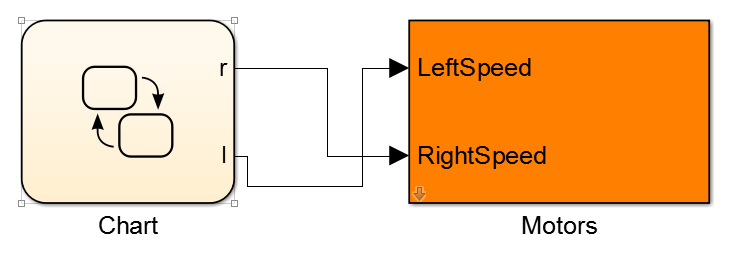
Video Reference: [Unit3d\_deadreckoning.mov](https://www.youtube.com/watch?v=B-bKLx2vvyU)

Charts allow us to represent the “flow” of our program differently than MATLAB functions and Simulink blocks alone. In many cases, programming a robot by chart flows makes things simpler. This is especially true when our robot is to follow out a set of complex movements which need to occur in a sequence.



Open the example model SimulinkDeadReckoningWithCharts.slx. Pressing ctrl + d will update the inputs and outputs on the chart so you can connect arrows to the miniQ motor block.

Arrange the blocks in the pattern you expect the robot to move in order to help you organize your thoughts for this “dead reckoning” robotics exercise:



Run the model above and observe the behavior of the robot. Press the reset button on the MiniQ to run the program deployed to the hardware again. Change some of the variables in the chart to suit the pattern of movement you desire.

*Intermediate Exercises:*

Use the dead reckoning chart to create movement for getting your robot to circle a table.

Then:

1. Create triangular movement
2. Create star movement
3. From a predefined location get the miniQ to move out into a distant location and go under a “bridge” or round a feature.

**Taking It Further**

*For more information, reference the “Instructor Reference Lesson 4 Unit 3” document in the Unit 3 Instructor Reference folder*

Video Reference: [Unit3e\_StateflowChartsAdvanced.mov](https://www.youtube.com/watch?v=wnUcv7pPSrI)

*Taking It Further Activity:* From a predefined location get the miniQ robot to reach a location and flash an “SOS” signal in Morse code using the RGB LED and Stateflow charts.

Creative Commons License

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/).