Alan Deuchert

Spring 2020

Lab Bot User Manual

**MANUAL DRIVE (manual.html)**



This is the bot’s main UI for manual control.

1. Side Nav
	1. **PING Distance**: Displays the distance between the bot and an object in front of it in centimeters, updates every half second. If **manual.c** is not running on the bot while manual drive is, then **error** is displayed.
	2. **Amount of Left Ticks**: Displays the amount of left wheel encoder ticks has been made by the bot from moving. If **manual.c** is not running on the bot while manual drive is, then **undefined** is displayed.
	3. **Amount of Right Ticks**: Displays the amount of right wheel encoder ticks has been made by the bot from moving. If **manual.c** is not running on the bot while manual drive is, then **undefined** is displayed.
	4. **Current Bot Status**: Displays whether the bot can move or not. If the text is green, the bot can move, else the text is red, and the bot cannot. The text changes based on if it can move or not. If **manual.c** is not running on the bot while manual drive is, thennothing is displayed.
2. Controls
	1. **Forward**: Clicking and holding down this button causes the bot to move forward, accelerating to half of its max speed. Releasing this button decelerates the bot to zero speed. If the bot is 20 centimeters directly from an object, this button ceases functionality and the bot stops automatically.
	2. **Left**:Clicking and holding down this button causes only the right wheel to move forward, causing it to turn the bot to the left, pivoting on its left wheel. Releasing this button causes the bot to stop turning. If the bot is 20 centimeters directly from an object, this button ceases functionality and the bot stops automatically.
	3. **Right**:Clicking and holding down this button causes only the left wheel to move forward, causing it to turn the bot to the right, pivoting on its right wheel. Releasing this button causes the bot to stop turning. If the bot is 20 centimeters directly from an object, this button ceases functionality and the bot stops automatically.
	4. **Reverse**: Clicking and holding down this button causes the bot to move backwards, accelerating to a quarter of its max speed. Releasing this button decelerates the bot to zero speed. If the bot is 20 centimeters directly from an object, this button does not cease functionality and can be used to move the bot away from the 20-centimeter threshold and regain control of the other buttons.

**AUTOMATED DRIVE**



This is the bot’s main UI for automated control.

1. Side Nav
	1. **Start Location**: Displays the starting cell location **(row, column)** of the bot. This location cannot be located off the clickable grid on the right nor can it be on a blocked cell. During the bot’s movement, this is replaced with **Current Location**, telling the user where the bot is currently on the grid for each step.
	2. **Facing Location**: Displays what cell **(row, column)** the bot is facing. By implementation. this location cannot initially be located off the clickable grid on the right, nor can this location be diagonal from the start location. During the bot’s movement, it updates by telling the user what cell the bot is facing on (or off) the grid for each step.
	3. **End Location**: Displays the ending cell location **(row, column)** the bot is to go to. This location cannot be located off the clickable grid on the right nor can it be on a blocked cell.
	4. **Set Start**: Sets the starting location of the bot with the selected cell on the clickable grid on the right. This button only works for the same reason as described in **Start Location** and updates it with the new information. If overwriting a previous starting point, the facing location is reset to null.
	5. **Set Facing**: Sets what cell the bot is directly facing location of the bot with the selected cell on the clickable grid on the right. This button only works for the same reason as described in **Facing Location** and updates it with the new information.
	6. **Set End**: Sets the ending location of the bot with the selected cell on the clickable grid on the right. This button only works for the same reason as described in **End Location** and updates it with the new information.
	7. **Set Block**: Sets a pre-existing block on the grid. It is *highly* *recommended* that blocks be placed first given that the implementation does not allow for the removal of blocks. If a block is placed where one did not mean to, the webpage must be refreshed.
	8. **Go**: Calculates the optimal route from the start location and end location and makes the bot move to that location (it should be noted that the “optimal path” does not take into account the bot’s starting orientation, so extra turns may be necessary). Updates the start location to the current location of the bot at a given step, the cell that the bot is facing at a given step, and, once completed, nulls out the end location for future automated commands. This button requires that the start location, the facing location, and the end location be told.
2. Clickable Grid
	1. **Selected Cell**: When clicking on a cell, the selected cell will appear with a yellow background with red text instead of a blue background with black text. When selecting another cell, the previously selected cell will revert to its original formatting.
	To perform either **Set Start**, **Set Facing**, **Set End**, or **Set Block**, a cell on the clickable grid must be selected before clicking those buttons.