Team 5 - Delivery Robot Code

The following public github repository was used to store all code pertaining to the delivery robot. As of 5/5/2023, this is Version 1 (root branch) for the repository.

Github Repo:

https://github.com/tu-jkinter/delivery-robot

To continue this project, it's likely that each subsequent team will need to branch their own version of the repo from the previous team's branch.

There are two sections of code for this project; one is the C++ code associated with the onboard Arduino Mega, and the other is the C# code associated with the manual control for the robot.

Arduino Code (C++)

The Arduino code is the code that runs the control systems for the robot (motors, encoders, IMU, LiDAR, etc). The **Arduino IDE** was used to write this code (however, some other Ide or text editor should work, such as VS Code).

Arduino IDE Link:

https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE

Manual Control (C#)

The code to manually control the robot using a gamepad is programmed in C#. This code is encapsulated within the **RobotBluetoothControl** folder inside of the repository.

It's highly recommended that **Visual Studio 2022 Community Edition** (or later) is used as the Ide to support the C# code.

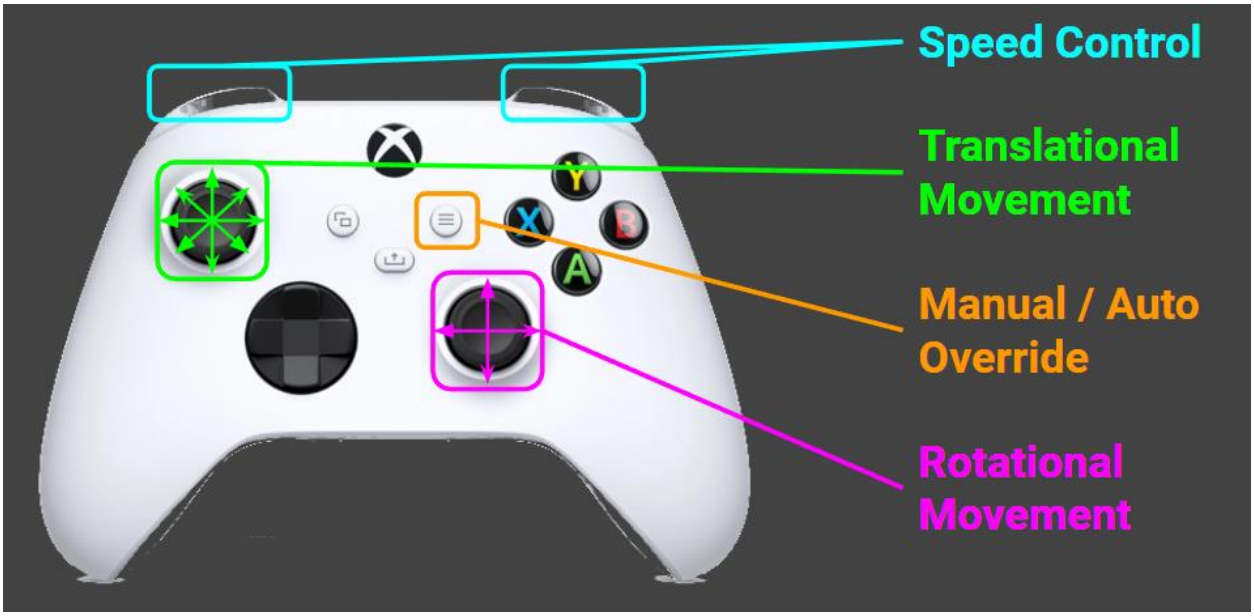
Visual Studio Link:

https://visualstudio.microsoft.com/downloads/

The code must be executed via Visual Studio. When executed, the code will open a serial communication with the HC-03 Bluetooth module on the robot (must connect the PC to Bluetooth first!). It then sends a single character, based on gamepad input, to the bluetooth module, where the Arduino code then handles execution (see Bluetooth.h). The serial communication port is currently initialized to COM13. However, this may need to be changed based upon the computer used. Additionally, the robot must be powered to receive serial communication, otherwise the program will fail to connect.

The C# program should identify any compatible gamepad, but an Xbox remote is recommended for ease and convenience. However, there may be issues with detection if multiple gamepads are connected to the PC running the software.

Controls Mapping:



The above image is from our final presentation slide that shows the available controls (as of the creation of this document).

Left Joystick. The left joystick controls the translational movement of the robot. By pointing the left joystick in any of the 8 shown directions, the robot will use its omni-directional wheels to move in the chosen direction.

Right Joystick. The right joystick controls rotational movement. By moving the stick left or right, the robot will rotate along its central axis counterclockwise or clockwise, respectively. By moving the joystick up or down, the robot will "arc" (rotate) along its back axis counterclockwise or clockwise, respectively.

Bumpers. The left and right bumpers are used to manipulate the speed; left decreases speed, while right increases speed. The minimum and maximum speeds, respective to the 0-255 range of the motors, are 50 and 150, respectively.

Note: When running at maximum speed, the motors consume more power. As more power is drawn, the Bluetooth module can occasionally fail. This is exacerbated by having lower battery power (less than 7V). When this happens, the manual control program will disconnect, and the robot will stop moving. If this occurs, reset the robot, and restart the program.

Start. The start button will transition between auto and manual movement. While in manual, this can be switched to auto at any time. While in auto, you must wait until the robot is moving forward (not running an obstacle avoidance algorithm).