Robot Code Document

The goal of this project was to build a robot capable of moving around four squares placed on the classroom floor six feet apart. So, the goal of the program is to control the robot to accomplish the project goal.

The program was written based on the realization that the robot has to make a few basic movements for the robot to travel from one square to the other. It was found that the robot needs to take at least four different actions before completing travel between squares. Those are 'Departure', 'Cruise', 'Arrival', and 'Turn.' Later, it was not necessary to make 'Arrival' a separate action. So the jobs that have to be done was reduced to three, 'Departure', 'Cruise', and 'Turn.'

The main algorithm of the program is repeating those basic jobs to move from one square to the other until the robot makes a complete three circuits; these jobs needed to be repeated twelve time to make three circuit.

First, to make debugging and interpreting the program easy, several variables were defined at the beginning of the program and declare a few integer variables for encoder reading and counting purpose.

The program starts with calling 'ready' function. The purpose of the 'ready' function is to prevent for the robot start moving right after the power-on and to wait until start-button is pressed. After 'ready' function, one can see that there are only three function calls in main function corresponding the three basic movement defined above and they are repeated twelve times by the for-loop to make three complete circuits.

The first function to be called in the loop is the 'departure' function. Its goal is to align the robot to the black tape in front of the robot to direct the robot straight toward the next square. Since the robot detects the square using light sensors attached on both sides of the robot, the robot has to move straight toward the next square so that at least one light sensor catch the black line to stop the robot.

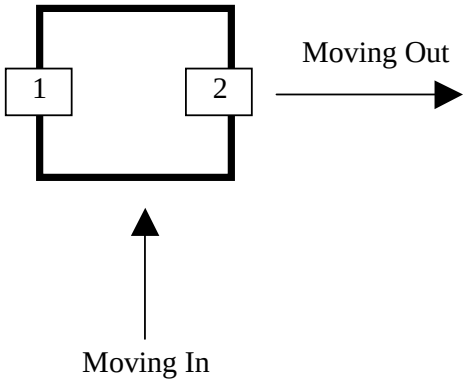
Aligning the robot to the black line is done by the 'adjust_to_line' function. This is how this function works. It starts with checking which light sensor is on the black tape. If both light sensors are on the tape, it does nothing and stops the robot. If only one of the light sensors is on the line, the function sets the motor power so that the side not on the black tape moves and the other side stays on the black tape. If neither of the light sensors are on the tape, the function makes the robot move slowly till either the sensors sees the black tape. This function is repeated several times to make sure the robot is align properly.

After 'departure' function, 'cruise' function starts. The job of 'cruise' function' is to make the robot run straight. If the robot is aligned proper against the black tape, running straight is enough to get the robot find the next square. The function achieves its goal by controlling the power on the each motor based on the encoder reading. At the beginning, readings from each encoder are compared. If there is difference, it reduces the power of the motor with higher reading and keeps it doing until the readings become the same. This function is executed until one of the light sensors

sees the black tape and right encoder reading is higher than 40 the reason for the limit on the encoder reading will be explained later.

When the robot arrives at the next square, it starts turning using the 'turn' function. The robot moves a little forward before turning to place itself at the middle of the square. Rotating each wheel in the opposite direction accomplishes the turn. To make a 90 degree turn, each motor stops when the encoder reading becomes six; it was found that the encoder reading twelve corresponds to one rotation of the wheel and a half turn of each wheel in the opposite direction is needed to make a 90 degree turn. Both encoders are reset to zero before the turn to ensure that the appropriate encoder reading of six is counted on both.

Now, it goes back to the 'departure' function to align itself to the black line. This time, there could be a problem after aligning. The robot could be aligned one of the two black line; either position 1 or 2 in figure below.



This is the reason that there is an encoder reading value limit to get out of the 'cruise' function. Without this limit, if the robot starts at position 2 there is no problem since both light sensors will not see the black tape until the robot arrives at the next square. However, there will be problem if the robot starts at position 1 since the robot reaches the next black tape of the same square and will think it has arrived at the next square. With this limit, the robot will not get out of the 'cruise' function before it gets the next square.

After looping twelve times, it finishes execution printing "I am done!!!!!!!."