Summary

220514C, 220562U, 220071M, 220343B, 220004M - Pixie Bots

9/28/2024

Task 1: Counting and Line Navigation

Objective: Read a barcode as the robot follows a line by detecting patterns in the line (thick/thin lines) using IR sensors and encoders.

0.5em4plus2minus

- **Set up IR sensors:** Configure the IR sensors in the line-following robot to detect the presence or absence of the line. Ensure multiple sensors cover the width of the barcode to detect different line segments simultaneously.

- **Encode movement with the motors:** Use the motor encoders to track the distance traveled while following the line. This will help in synchronizing the robot's movement with the barcode reading.

- **Barcode encoding logic:** Barcodes consist of different width line segments (thick and thin). As the robot moves over the barcode, the IR sensors will detect the width of the line segments by measuring the time between transitions (black-to-white or white-to-black).

- **Sensor transition detection:** Use the IR sensors to record the transitions between line segments. A longer duration between transitions indicates a thicker line, while a shorter duration indicates a thinner line.

- **Encoder data for synchronization:** Use the encoder data to correlate the distance traveled with the barcode pattern. The robot will calculate the speed of movement over each line segment and compare it to known barcode patterns.

- **Decode barcode:** Based on the timing of transitions and encoder data, decode the barcode into the corresponding numbers or information.

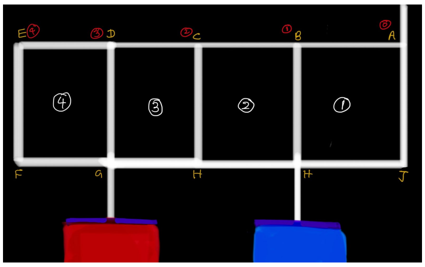
- **Process barcode data:** After the barcode is fully read, process the information, such as determining a task for the robot or storing the data for later use.

Task 2: Maze Navigation and Box Manipulation

Path Decoding and Calculation: After decoding and calculating mod 5 of the number from the code, the path is calculated according to the number and the wall placement. For efficiency, a hard-coded implementation is planned.

Virtual Boundary Navigation: 0.5em4plus2minus

- Checkpoints: There are 4 in small loops, but in some cases, the robot must complete 2 cycles combined. Therefore, the checkpoint count is either 4 or 6.



- Reverse or Straight: In some cases, the robot has to complete cycles in reverse. This can be derived from the arrangement of checkpoints but can also be input manually.

Sensor Integration: The robot will check the front black wall in front of the blue square using one-sided or front ultrasonic sensors. The readings from position H will help complete the path.

Junction Recognition: The robot, based on an IR array, will easily recognize encountering junctions and identify the location.

LED Signaling System: A function will control the LED to turn on and off.

Task 3: Color Line Following

The robot will follow a colored line (red or blue) using an 8-bit IR array designed to detect contrast between the line and its background. The process includes identifying the color of the line and then reading the IR array's analog output to track the line. A threshold will be set to identify the white line and color line.

PID Control System: A Proportional-Integral-Derivative (PID) controller will continuously adjust the robot's direction by comparing the line's position to the robot and adjusting the motor's speed to keep the robot centered on the line.

Task 4: Dashed Line Navigation

Objective: Follow dashed lines, adjusting movement when transitioning between dashes by leveraging previous movement direction.

0.5em4plus2minus

- **Line following logic setup:** Use IR sensors to follow a continuous line. When solid, the sensors detect constant transitions of black-to-white.

- **Detect dash gaps:** IR sensors will detect the absence of the line when encountering a gap (white space).

- **Store previous movement direction:** Record the last movement direction (left, right, or straight) before encountering the gap.

- **Continue moving forward during gap:** During the gap, the robot continues moving in the last recorded direction to prevent drifting off course.

- **Search for the next dash:** After moving a predetermined distance (based on motor encoders), expect to encounter the next dashed line segment.

- **Re-align on next dash:** Once detected, the robot resumes standard line-following behavior.

Task 5: Portal Navigation

At stage 5, the robot stops and waits 10 seconds, checking if the gate is opened. The ultrasonic sensor measures the distance to the gate, determining whether it is opened or closed. The robot moves forward when the gate opens.

Task 6: Box Arrangement

Objective: Arrange 3 boxes in ascending or descending order (blue line for ascending, red line for descending). The ultrasonic sensor measures the height of each box.

Process: The robot arranges the boxes in the empty square in front of their respective places before placing them in their correct positions.

Task 8: Coin Drop and Task Completion

Algorithm: Depth-First Search (DFS) for terrain navigation and X detection. To efficiently explore the area and locate the X mark, the robot uses a Depth-First Search (DFS) algorithm. The robot will rely on its sensors to detect obstacles (walls) and navigate over the uneven terrain.
Initialization: 0.5em4plus2minus

- Starting position: Robot starts near the area marked with three black stripes.

- Ultrasonic Sensor: Mounted on top of the robot, measuring distance to obstacles (walls).

- IR Sensor: Detects the X mark on the ground.

- Servo Motor: Holds the 2 Rupee coin, ready to be dropped.

Navigating the Terrain Using DFS: 0.5em4plus2minus

- The robot uses DFS to explore the area, changing direction upon detecting walls.

- It moves over uneven terrain with larger wheels.

Detecting the X Mark: Once the IR sensor detects the X, the robot aligns itself at the center.

Dropping the Coin: The servo motor releases the 2 Rupee coin onto the X, completing the task.