ENGINEERING JOURNAL

COMP20170
Introduction To Robotics

RONIT – 18204009 TARAN – 18203372
KRISHNA – 18203023 ENAMUEL – 18729365

GROUP 2

24th January 2020

- **Our goals for today were:**

  - To construct the driving base for our robot.
  - To add all the necessary sensors and wiring.

- **Our accomplishments today were:**

  - Completion of the robot. We constructed the driving base and added 4 sensors (Gyro, Ultrasonic, Touch and Colour) to it.

- **Some difficulties we encountered included:**

  - Struggling to find certain pieces required to build the robots base.
  - Waiting for people to finish looking for their pieces.
  -Forgetting to add a piece and having to disassemble and reassemble the robot again.
  - Staying focused on the task and looking for the necessary pieces at the right times.

- **Personal roles or contribution(s) to the group effort was/were:**

  -Everyone helped each other in finding the robot parts and making the robot.

31st January 2020

- **Our goals for today were:**

  - To start writing the robot’s code.
  - To test our programs on the robot.

- **Our accomplishments today were:**

  -Overcoming the obstacle of a faulty motor and making our robot work again.
  - We downloaded programs on the robot and tested them.
  - We programmed the robot to move and turn.
  - We programmed the robot to use sonar and touch sensors effectively.

- **Some difficulties we encountered included:**

  - Dealing with a faulty motor. The left motor was faulty and hence it didn’t let the robot move forward straight.
  - Re-assembling the robot to replace that motor.

- **Our personal roles or contribution(s) to the group effort was/were:**

  Ronit- Writing codes to test the touch the sonar and touch sensor.
  Emanuel and Taran- Writing code to test the movement of the robot. Identifying that the left motor is faulty. Re-constructing the robot.
  Krishna- Couldn’t attend the meeting but helped Ronit with some code that he had prepared.

6th February 2020

- **Our goal for today was:**

  -To work towards the completion of challenge 1.

- **Our accomplishments today were:**

  - We made the robot go forward and hit the box with its touch sensor.
  - The robot then reversed, turned left and found its way to the red line.
  - It then followed the red line to the calibration area.
  - It reversed into the calibration area then turned right and left.

- **Some difficulties we encountered were:**

  - The motors malfunctioning and not being able to move the robot forward.
  - Trouble with coding the sonar and light sensors to perform reliably.

- **My personal role or contribution(s) to the group effort was/were:**

  Ronit and Krishna - Coding and making sure the robot reaches the wall and identifies the red line and moves efficiently and smoothly.

  Taran and Emanuel - Overcoming the problem of the motors malfunctioning.

7th February 2020

- **Our goal for today was:**

  -To complete challenge 1 and film the robot completing the challenge.

- **Our accomplishments today were:**

  -We made the robot successfully trace the red line back from the calibration area to the finishing zone.
  -Filmed the robot completing the task.

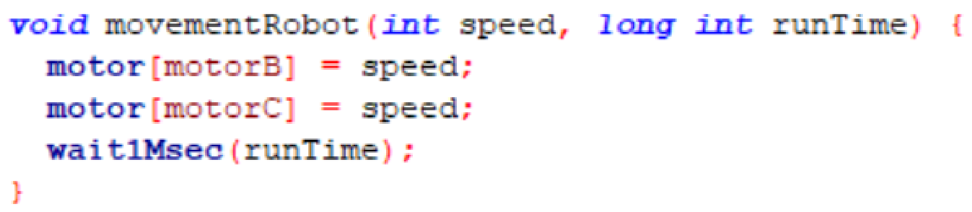
- **Some difficulties we encountered were:**

  -the robot not reversing correctly into the calibration zone.
  -the robot confusing the red lines in the calibration zone to the original red line it was meant to follow.

- **My personal role or contribution(s) to the group effort was/were:**

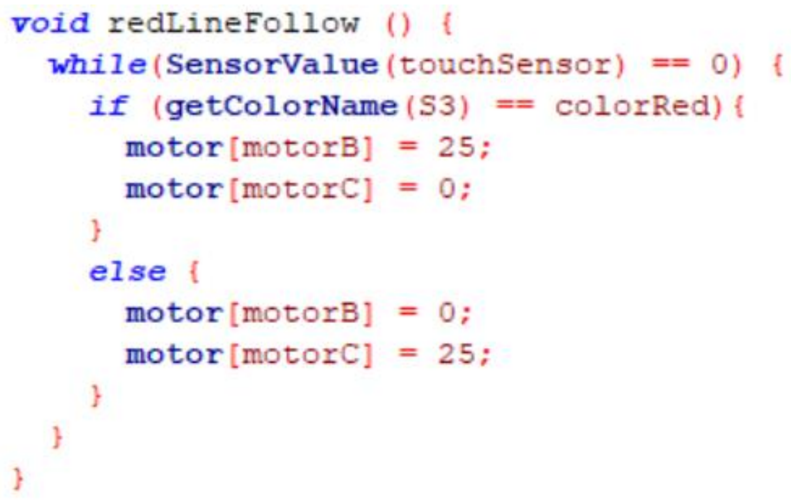
  Ronit and Krishna- Making sure everything in the task was running smoothly and following the red line back to the box. Video Edited & Effect Added.
  Taran and Emanuel- Modularising the code and making it more efficient. Overcoming the problem of the robot identifying the wrong line. Filming the video.

The following are the functions used in the body of main:



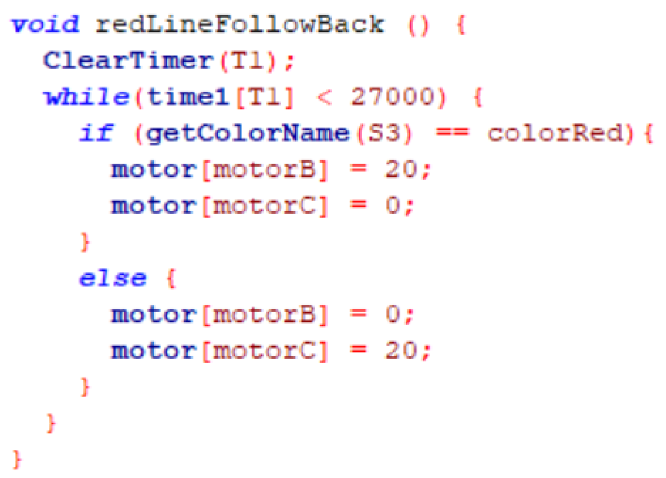
```
void movementRobot(int speed, long int runTime) {
  motor[motorB] = speed;
  motor[motorC] = speed;
  wait1Msec(runTime);
}
```

This code replaces all the instances the robot moves forward or backward. It takes an integer parameter as input and its value becomes the power value assigned to both motors. Therefore, the robot moves in a straight line. Another integer parameter is taken is input whose value determines the amount of time the robot moves in a straight line.



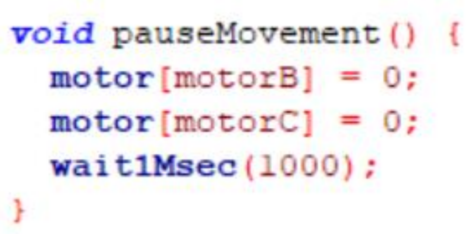
```
void redLineFollow () {
  while(SensorValue(touchSensor) == 0) {
    if (getColorName(S3) == colorRed){
      motor[motorB] = 25;
      motor[motorC] = 0;
    }
    else {
      motor[motorB] = 0;
      motor[motorC] = 25;
    }
  }
}
```

When the robot approaches the red line which it must follow to the calibration area, this code is executed. While the touch sensor isn't pressed, if the robot detects a red line, it turns right and if it doesn't sense a red line it turns left. Therefore, it always stays on the red line until the touch sensor detects a wall.



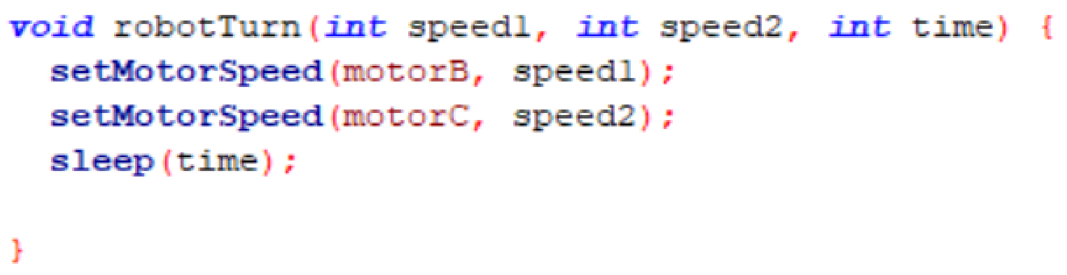
```
void redLineFollowBack () {
  ClearTimer(T1);
  while(time1[T1] < 27000) {
    if (getColorName(S3) == colorRed){
      motor[motorB] = 20;
      motor[motorC] = 0;
    }
    else {
      motor[motorB] = 0;
      motor[motorC] = 20;
    }
  }
}
```

This function is practically the same as the redLineFollow() function in the while loops body. But there are a few changes. A timer has been declared and in a time of 27 seconds, the robot follows the red line back at a slower speed until it gets close to the goal area.



```
void pauseMovement() {
  motor[motorB] = 0;
  motor[motorC] = 0;
  wait1Msec(1000);
}
```

This function is called after every other function and the robot stops moving completely for 1 second. This is done to improve the functionality and efficient movement of the robot. This also helps to check functionality of each function.



```
void robotTurn(int speed1, int speed2, int time) {
  setMotorSpeed(motorB, speed1);
  setMotorSpeed(motorC, speed2);
  sleep(time);

}
```

Whenever a right or left turn is made, this function is called. It takes three integer parameters; 2 different values assigned to the speeds of the left and right motor and another value which determines the amount of time the robot will remain still. If speed1 has a value greater than that of speed2, the robot turns right; left if speed2 is greater than speed1.