Arduino Robotics Assignment CS12020

Jamaine Christian

Jac105@aber.ac.uk

16/12/2021

Overview

In this assignment, I was tasked with programming the robot to follow an obstacle course. The first task was to follow a black strip as its path. The second task was for the robot to avoid obstacles and then reconnect to the strip. The third task was barcode scanning, where the robot passed over horizontal strips with different amounts of strips and spacing between them, which corresponded to different actions taken by the robot. Finally, the robot had to then deviate its path depending on the barcode and continue to follow the strip with different strip colours and backgrounds or turn around and go back to the start.

Key functions

Driving the robot

To get the robot to drive I first included the servos.

Then, I tested the stop value of each servo by creating a loop utilising 'rightServo.write()', 'leftServo.write()' and 'Serial.println()' to change and display the servo speed at the time, I then wrote down the values they stopped at and put this into EEPROM.

After, I created a function called 'setSpeeds()', which adds or subtracts a float value to the stop values of each servo to get the robot to move straight. Then I created a function called 'distance()', this function uses the 'SetSpeeds()' function and delay function to get the robot to drive for a certain time at a specific speed, using the 'abs()' function, in either direction.

Finally, I created two function that turned the robot for a specific number of degrees, one for a left turn and the other a right turn. The function uses 'setSpeeds()' to have one of the servos to turn while the other is stationary and a delay before resetting the speeds to zero.

Following the line

For line following I setup the pins for each of the LDRs. I then tested the LDRs worked by using 'analogRead()' and 'Serial.println()' in the loop function to display the values on different surfaces.

After I created a calibration function which assigned the LDRs values at specific times to different variable to get a value of white paper and the black line. I then averaged this to produce the value of the grey.

Then I created a function called 'lightDark()' to display what colour the LDRs were detecting. I used a buffer variable which was equal to 100 and added or subtracted this to the previous LDR reading variables depending on if the current reading was a lower or greater value.

Afterwards, in the loop function I used if statements to compare these variables to what the LDRs are currently reading and then got the robot to react to each scenario of line following and either got the robot to drive straight or turn left and turn right to recentre the robot. I used the '&&' boolean operator to specify I wanted multiple scenarios to be true at the same time before it reacted. Here is an example of that code:

```
//DrivingStraightBlackStripGreyBackground

 else  if ((analogRead(leftLDR)>=(LEFT_LDR_MEAN - BUFFER)) && (analogRead(middleLDR) <=
(MIDDLE_LDR_BLACK + BUFFER))&&(analogRead(rightLDR) >= (RIGHT_LDR_MEAN - BUFFER))){

   setSpeeds(40, 40);

  }
```

Obstacle avoidance

For the obstacle avoidance, I setup the IR transmitter pin as an output and then test it can detect an object. To do this a created a boolean function called 'checkReflection()' which used 'tone()' to start the IR transmitter to send out impulses as a specific frequency and then 'digitalRead()' to read whether pin 2 (IR receiver) read either HIGH or LOW. If it read HIGH it would print "obstacle detected" and return true. Else it would print "No obstacle detected" and return false. I then called 'noTone()' to turn of the impulses to finish the 'checkReflection()' function.

To then get the robot to avoid the obstacle I created a new function called 'obstacleAvoidance()' which incorporated my turn functions , 'distance()' and 'checkReflection()'. This function would see if 'checkReflection()' was true, and if so, it would loop the first algorithm, which got the robot to turn right move forward a distance and then turn left, if the object was still present it would repeat. It would then break this loop and turn right one more time and move some extra distance (20cm) to reduce the likelihood of the robot still hitting the obstacle. It would then turn left move 20cm forward and then turn left. If the side of the obstacle is detected, it would loop the 2nd algorithm, which is the same as the first, until no object is detected. Finally, it would break this loop, turn right move an extra 20cm for safety turn left and drive the same distance it did for the front of the obstacle before turning right again to re-join the line.

Barcode scanning

For the barcode scanning I created an array called currentTime[6] and set all these values to zero. Each time all LDRs read "BLACK" it would set one of the currentTimes to the current time using 'millis()' and the same for leaving a bar (only middle LDR reading Black). Using these times, I worked out the bar widths and space between bars. I also used a fixed time variable to only count bars for a specific amount of time before deciding which barcode it was. Indicating this with LEDs that I setup with a function.

Finish

I used my ‘setSpeeds()’ and turn function to either get my robot to either curve left, right or turn around to finish the track.

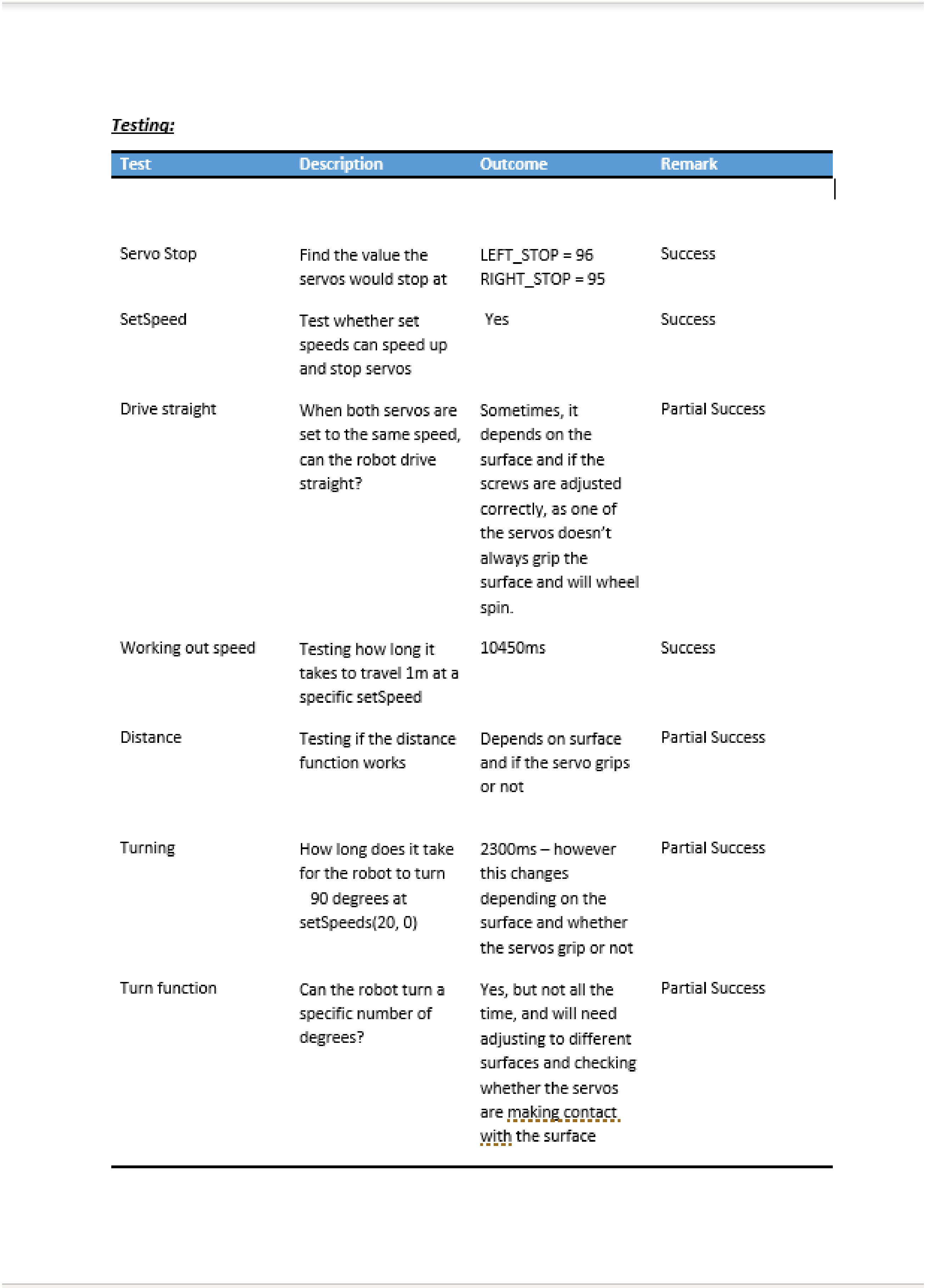
Results and Discussion

Results

1. **Follow Line –** This task was accomplished
2. **Obstacle Avoidance –** This task was partially accomplished
3. **Barcode Scanning –** This task was not accomplished
4. **Finish –** This was not accomplished but robot can follow lines with different shades

Discussion

I was faced with many challenges as my robot’s servo wouldn’t grip surfaces correctly and disrupted it from driving straight, this affected my obstacle avoidance. I used embedded if statements for the barcode scanning but I am unsure why it is not working, which didn’t allow me to finish the course.



Testing:

Test	Description	Outcome	Remark

Servo Stop	Find the value the servos would stop at	LEFT_STOP = 96
		RIGHT_STOP = 95
			Success

SetSpeed	Test whether set speeds can speed up and stop servos	Yes	Success
Drive straight	When both servos are set to the same speed, can the robot drive straight?	Sometimes, it depends on the surface and if the screws are adjusted correctly, as one of the servos doesn't always grip the surface and will wheel spin.	Partial Success
Working out speed	Testing how long it takes to travel 1m at a specific setSpeed	10450ms	Success
Distance	Testing if the distance function works	Depends on surface and if the servo grips or not	Partial Success
Turning	How long does it take for the robot to turn 90 degrees at setSpeeds(20, 0)	2300ms – however this changes depending on the surface and whether the servos grip or not	Partial Success
Turn function	Can the robot turn a specific number of degrees?	Yes, but not all the time, and will need adjusting to different surfaces and checking whether the servos are making contact with the surface	Partial Success

Pushbutton	Does the pushbuttons work?	Yes	Success
IR transmitter	Test if IR sensor can emit and receive signals	Yes – however the direction the sensor is pointing in can affect this outcome.	Success
Obstacle Avoidance	Can the IR transmitter detect and avoid the obstacle to rejoin the line following?	It can detect the obstacle and avoid it however the robot usually spins out, therefore cannot rejoin the line	Fail
LDRs	Do the LDRs work	Yes	Success
Distinguish colour	Can the LDRs distinguish different colours after the calibrateLDR function has been used?	Yes, but you have to keep the lighting the same otherwise it messes up	Success
Line following	Can the robot use the LDRs to follow the line	Yes – however one of the servos doesn't always grip to the surface and can wheel spin, which can hinder its progression across the line but will still follow it	Success
Barcode Scanning	Can the robot read the different barcodes and react to it?	No	Fail