Report

Objective:

- Implemented a line following car that avoids obstacles.
- The car follows a black line on the ground and changes its direction accordingly
- When the car faces and obstacle it avoids it and return back to the line after avoiding it
- When the car faces an obstacle the buzzer beeps

Equipment used:

- 4 motors
- Car body
- Motor driver L298
- 2 voltage regulators (5 volts)
- Logic convertor 3.3 v <-> 5v
- 4 battery cells 3.7 v
- Breadboard
- Jumpers
- FPGA
- Ultrasonic sensor
- 2 infra-red sensors (line-following sensors)
- Buzzer

Hardware:

Implementation:

- Constructed the car body by assembling the wheels, body and four motors
- Fixed the components of the car using nails and a screwdriver
- Fixed the breadboard on top of the car body using a double-face tape
- Fixed the ultrasonic sensor on top of the car body using holder and nails
- Fixed the 2 line-following sensors under the car body using nails
- Inserted the fpga on top of the car body
- 4 battery cells where fixed on top of the car body each 2 cells where fixed together using a holder let's give the first holder symbol A and the second holder symbol B
- Both battery holders were connected to the breadboard using jumpers
- A regulator was connected to the breadboard taking power from holder A thus giving output voltage of 5v The output of this regulator was connected to the fpga to supply power
- A second regulator was connected to the breadboard using jumpers
- This regulator took power from holder B thus giving output voltage of 5v let's give this regulator symbol R
- The 3 sensors were connected to the regulator R taking power of 5v
- Motor driver was connected to the regulator R taking power of 5v

- Motor driver was also connected to the breadboard (directly) taking voltage of 7.4v from battery holder B
- The motor driver was then connected to the fpga and connected to each motor using jumpers
- Jumpers connecting between each motor and the motor driver were soldered into the motor
- The buzzer was connected to the fpga using jumper
- 3 sensors were connected to the fpga however we used the logic converter to convert the output voltage from the sensors to 3.3v to be taken by the fpga as input

Software:

Summary of functionality of the code:

- The line-following sensors will give the fpga output 1 if it reads any shade white and 0 if it reads amy shade black accordingly the fpga will control the motors in order to follow the line
- If both sensors read white then both motors will move in the same direction forward
- If one sensor reads white and the other sensor reads black then two motors will move backwards and the other two motors will move forward causing the car to rotate towards the black line and continue on it.
- If both sensors read black the motors will stop
- The fpga causes the ultrasonic sensor to send a trigger.
- After the ultrasonic sends a trigger we enter waiting mode in which the fpga waits for the echo
- As long as we are in waiting mode we increment a counter
- when we receive the echo we calculate the distance between the sensor and the obstacle using a simple equation that relies on the speed of sound and count that we have reached in waiting mode then we divide the distance by two in order to get the actual distance between the sensor and the obstacle and the distance is displayed on the seven segments
- If at any point the distance between the obstacle and the sensor is less than 30 cm we stop acting on the input of the line-following sensors
- Then the following steps are executed systematically:
    1. The car stops and the buzzer beeps for 3 seconds
    2. Then the car rotates right for a specific interval of time. This interval of time was obtained using trial and error
    3. The car moves forward for another interval of time also obtained using trial and error
    4. The car then rotates left for the interval of time equivalent to the interval of time in step one
    5. We start acting according to the line-following sensors once again the moment the line following sensor detects

the black line again it will rotate right and follow the line once again

- This process is repeated every time the ultrasonic sensor detects an obstacle

Budget:

- Car body 275
- jumpers 100
- Ir sensor 55 * 2
- Battery charger 85
- Motor driver 50
- Ultrasonic sensor 45
- Breadboard 20
- Holder for ultrasonic 20
- Battery 50*4
- Voltage regulator 5 * 2
- Buzzer 5
- Logic converter 15*4

Issues and their solutions:

- Motors were supposed to take 5v from the regulator however since the regulator gives output of 5v and 1A the current was not enough to run the four motors so we solved the problem by connecting the motors directly to the 7v battery (holder B) giving the motors higher current and allowing them to run swiftly
- The output of the fpga was 3.3v and components required 5v input so we handled this problem by using logic convertor converting the output of the fpga to 5v and inserting into the motor driver
- 1 battery was not enough to supply power for all the components thus we used two batteries to supply sufficient power for the fpga and other components

Code analysis:

The code is composed of two entities a top level entity and the ultrasonic entity
The ultrasonic entity is responsible for operating the ultrasonic sensor and showing the distance that measure on the fpga's seven segments and takes as an input the clock and the sonar echo and the NotEnable and it outputs the sonar trigger and the 3 seven segments

In the ultrasonic process we send a trigger and wait for the echo and count the time then we convert this time to distance and display it on the seven segments and if the distance less

than 30 centimeters we make the stop equal to 1 which will be useful in the top level entity because the stop is related to the obstacle avoidance part

In the top level entity there is one process which operates when the clock is rising edge now if the stop is zero and the turn is zero we will do the line following part and if the stop is 1 and the turn is zero we will make the turn equal to 1 and last but not least if the turn is one and the stop is zero we will start a counter in the first three seconds the car will stop and the buzzer will turn on (beep) in the second part the car will turn left then it will move forward then it will turn right and if the counter is bigger than a certain value the count will reset to zero and the turn will be zero. This values of the time interval was chosen using trial and error.