

Remote Control Car with Collision Detection

Dominic Gaines, James Zhan, Joseph Telaak, Evan Trauger



Technical Approach to Reach Requirements

- Set up OS on Raspberry Pi and confirm LIDAR is connected and functions as intended
- This portion is written in Python 3 and should:
 - Be connected as long as the LIDAR has the correct IP and Subnet Mask
 - Sense any large objects within the front 270° of the car (135° Left and Right)
 - Poll an arbitrary number of pings from the sensor
 - Have a polar distance attached to each ping in mm (r and Θ)
 - Shut off any power to motors if an obstacle is detected within 6-12 inches (range will be adjusted for any latency issues)



Technical Approach to Reach Requirements

- Set up RC receiver to Arduino Mega via Serial. Arduino is written in C++ and should:
 - Read the radio input from the controller to receiver (at 9600 baud every 100 ms)
 - Run the serial input through an iBUS instance using the IBusBM library (this is the receivers protocol)
 - Split the input into 6 respective channels (we will be using 3 of these)
 - Confirm which Channels by using the remote controls
 - Ch1 will be Left/Right, Ch2 will be Forward/Reverse, Ch6 will be for lights
 - Send the decoded channel values to the Raspberry Pi



Technical Approach to Reach Requirements

- Return to Raspberry Pi; it should now be receiving values from the Arduino
 - Code a proportion to convert Ch1 and Ch2 values from $\{-100 < x < 100\}$ to $\{0 < x < 255\}$ (Negatives will denote which pin to output to)
 - Relay Ch1 signal to Power Steering Arduino to turn left or right accordingly
 - Relay Ch2 signal to Motor Controller to go forward or backwards accordingly
 - If Ch6 is 1, relay signal to Power Steering Arduino to turn car lights on
- Once Pi is coded and compiles correctly, make sure everything physically works and is wired correctly!



Measures of Performance

- The speed of the car could have been increased (around 6 mph)
 - Had to limit the speed so that the wheels could get traction
 - Creates more control
 - Original objective goal was for it to be moving
 - The threshold goal was 6 miles per hour
 - Reached max speed but gearbox stripped itself
- Could not reach any acceleration goal due to traction
 - Reached our objective goal
- Lidar works; however, we could not get a good measurement of performance
 - Too many libraries and functions needed to be implemented
 - Causes pi to be too slow



THANKS!

Resources for Reference:

Dominic Gaines

Final Group Project

dcgaines@email.sc.edu

ELCT201-003



**Engineering
and Computing**

UNIVERSITY OF SOUTH CAROLINA