Date: 1-29-04
Student: Mike Bonestroo
TA: Louis Brandy
Max Koessick
William Dubel
Instructor: A. A Arroyo

University of Florida
Department of Electrical and Computer Engineering
EEL 5666
Intelligent Machines Design Laboratory

Special Sensor Report

IR Package

The front of the robot is equipped with two IR Sharp GP2D12s. This sensor takes a continuous distance reading and reports the distance as an analog voltage with a distance range of 10cm (~4") to 80cm (~30"). The interface is 3-wire with power, ground and the output voltage and requires a JST 3-pin connector. They are used in the obstacle avoidance behavior and insure that the robot doesn’t run into a wall when it goes off the track. The IR input is read by the microprocessor and if the IR reads a number higher then the threshold that is declared in the code the robot takes action according to which IR says there is an object in front of it.

Bump Switch Package

The bump switches/sensors are located in the front and the rear of the robot. As of now I am only going to use one on each ends, but this may change as I learn more about the way the robot moves and gets stuck in corners. The way that these work is simple, the sensor send the microprocessor a signal that is given to the sensor with the use of resistors. Once the microprocessor receives a signal from the switch it knows that there is something in front of it or behind it and the robot takes action on moving away from the object. If there is no signal coming from the sensor then the robot carries on with its normal behaviors.

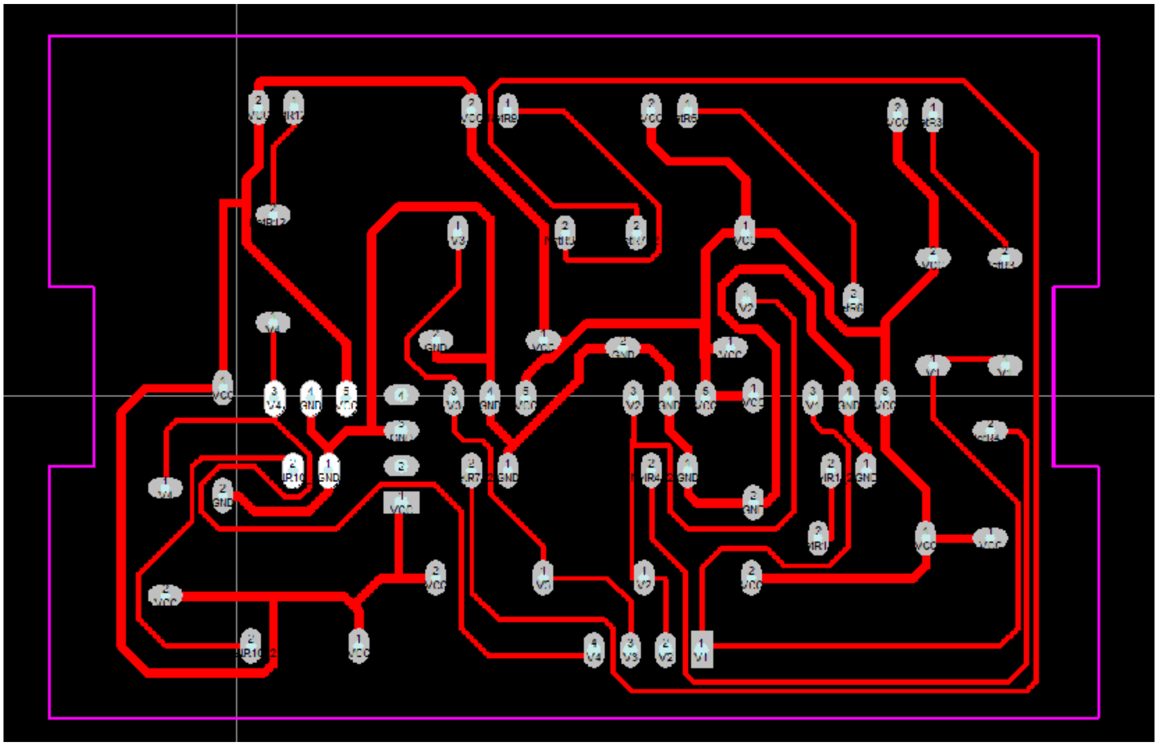
CDS cell Package

I am using these sensors to tell the robot to speed up or decrease speed. There is two CDS cells located on either side of the robot on the undercarriage. CDS work by having varied resistance according to the amount of light that they are subjected to. Under dim to low light, there is high resistance. Under high intense light, there is low resistance. With this varied resistor and a fix resistor you can use a calculation that will help you create a varied value input to your microprocessor using an A/D port. With the varying inputs you can declare a threshold to make a certain behavior run if broken. These sensors were going to be running under the car to change the speeds of the vehicle as it went around the track. Unfortunately the CDS cells are way to slow to react to a simple led on the track. So with the CDS not able to sense an LED to slow down the car I had to move to another choice. I would caution anyone that is choosing this sensor, if you are moving at any rate of speed faster then a servo would move, don't use a CDS cell.

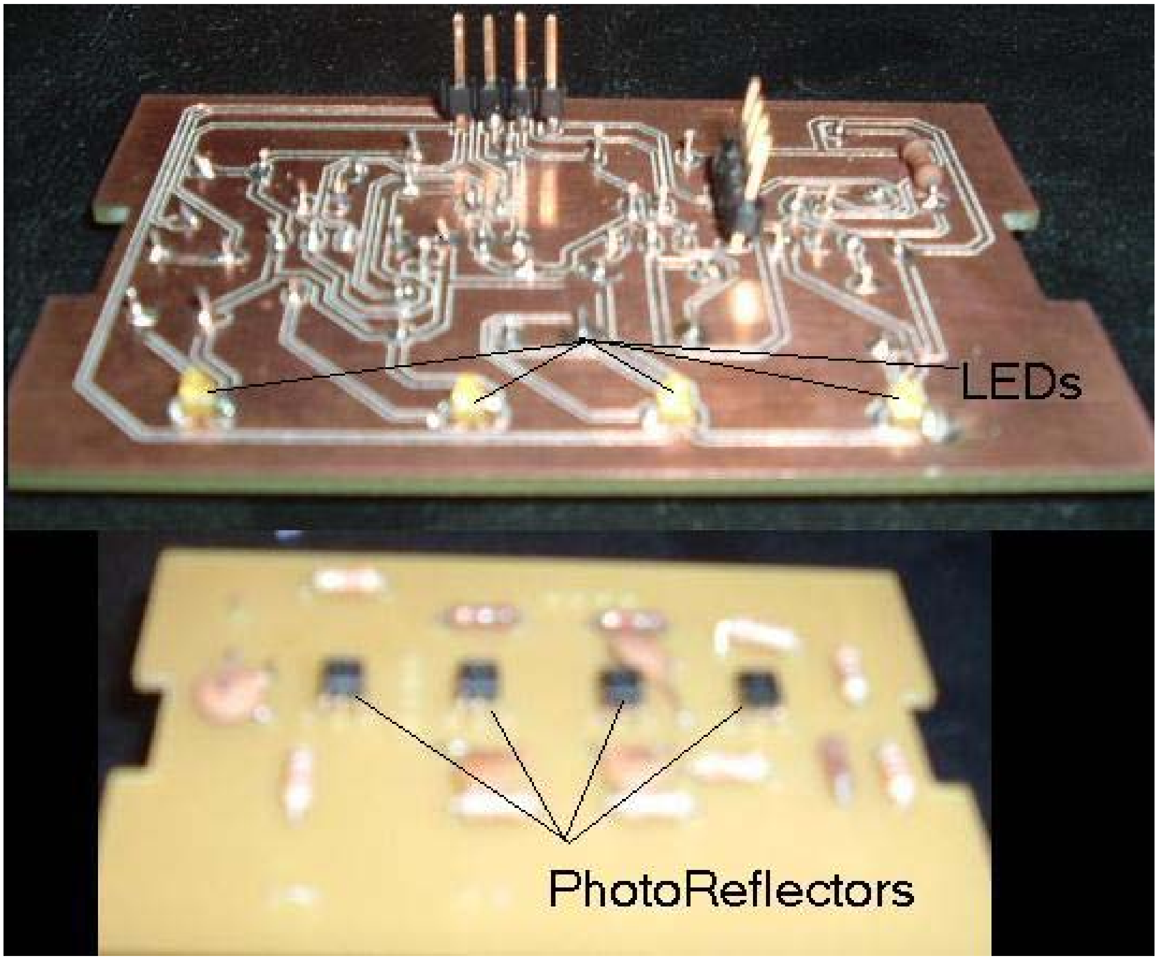
Line Following Steering Suite

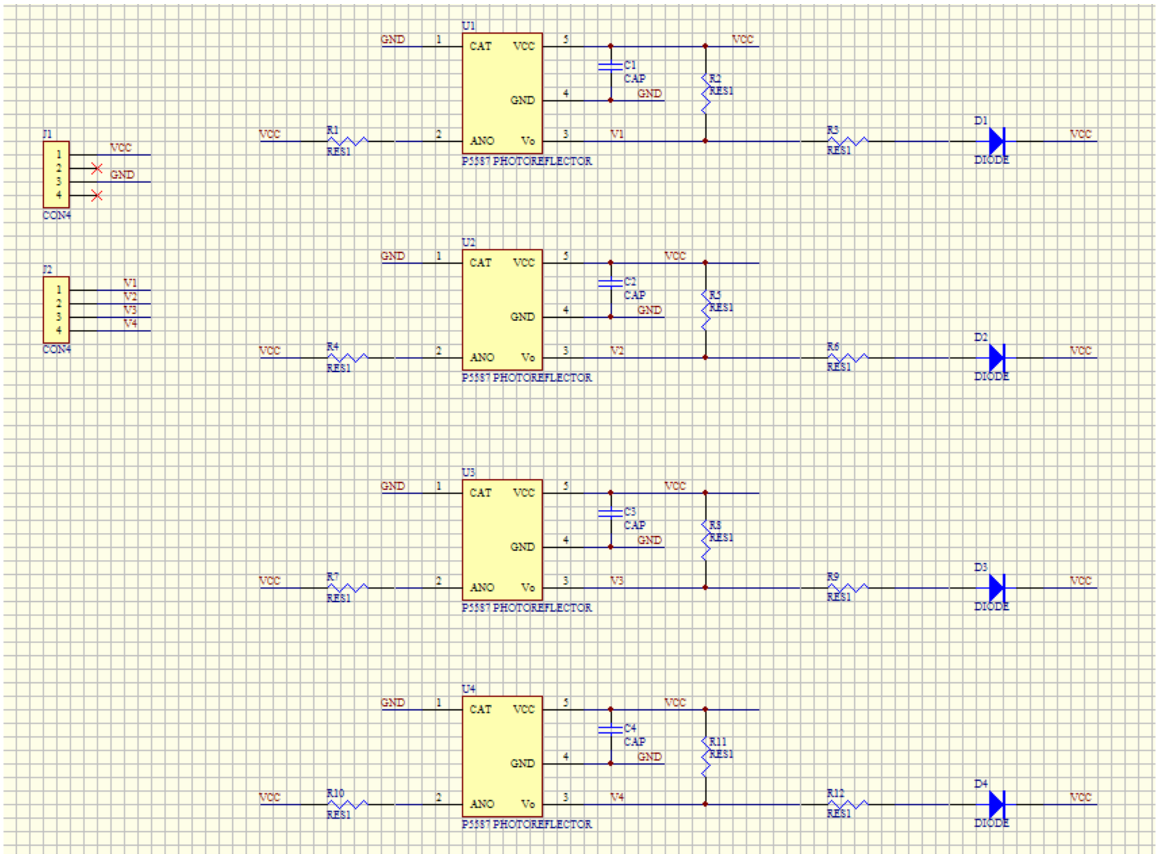
With not having much background in engineering, my special sensor is not much of a special sensor to most. In my opinion my whole project is a special sensor because it is a victory every time I get any results from my robot. But there is one unusual part to my car robot that interacts with a sensor that is not seen on most robots. So the combination of the two is going to be what my special sensor report is on.

My robot is a line following robot that is to be a second car for a young child to race when no other children are around to race him. To achieve the task of following a line I have used four Hamamatsu Photoreflectors. These four photoreflectors have been mounted onto a Protel board that was designed by Steven Pickles and Myself, which helps show if the photoreflector's see a line or not (below).



This board has four LED lights that are at the top of the board that signal it the line is being picked up by the Photoreflector. This is mainly a way for us to see if the sensor is working properly. Also on the board is the four sensors themselves, the five whole arrangements are there locations, and a series of resistors and capacitors. The schematics are as follows for this board.

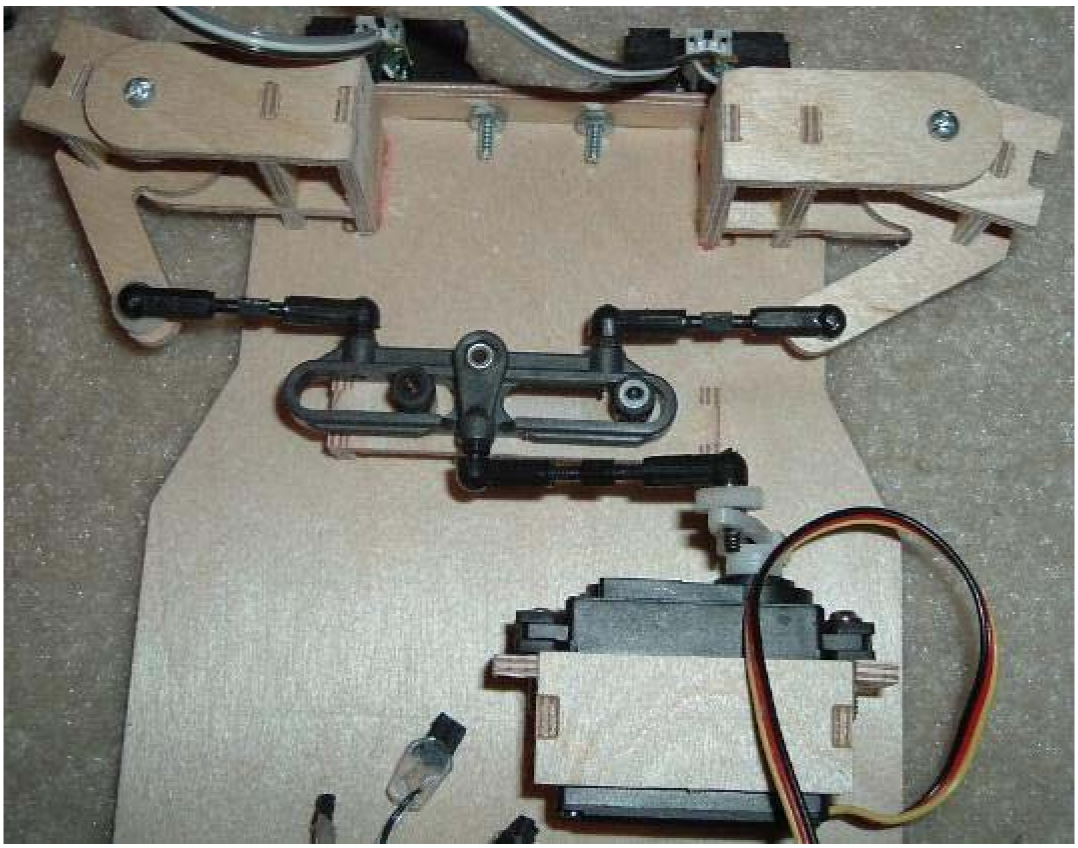

LEDs
PhotoReflectors



The arrangement of the photoreflectors is half of an inch between each one, so that I can get four different ranges of variables on how sharp to turn the robot. This configuration has proven to work extremely well with my robot. What I used as the line for the photoreflectors to pick up is three quarters inch electrical tape. Now you may be wonder what is different, this is all stuff that is the same as the past, well I moved in the outer two photoreflectors which is something that most robots haven't done, which has also turned into a great success. Another difference is explained in the next part of the report.

The steering in my robot is the part the sets it apart from the many robots done in this class. As it is going to steer like a true car that you drive. This type of steering has proved to be challenging, but I think that it has been done correctly and proves that front wheel steering is not as bad as one would assume.

The steering is off of a R/C car that I have had for some time, shown below, that I have converted to be used on my robot. It uses a series of ball and socket joints connected to dual eyelets to turn the front wheels. The servo is the actuator that creates the movement according to the input that it is given by the Microprocessor. There is seven different turning positions: straight, slight left, slight right, medium left, medium right, sharp left, and sharp right.



With the robot completely done, and steering tried and true, I would say that there are some bugs to be worked out. The steering system is limited by a few factors on my robot. In the picture above you can see that the steering is almost at it maximum right turn, but if you notice the servo is nowhere near it's maximum. The max right hand turn in this configuration is about a complete circle of ~4'8". To decrease the amount needed to make a complete turn all one would need to do is increase the "eye loops" of the centerpiece of the system. If you where to do that, make sure that you move the "eye loop pins" farther apart so that the distance of movement is increased. I am not sure of how tight of a turn that this setup would be able to make, but I know that you could greatly improve on the radius of the turn if you were to do this little modification. I would also advise that you be careful, as if the car has too sharp of a turning radius that it will start to push the front end straight and not turn. I would just play with different turning values and find this point by trail and error.

One other bug that I have run into is a problem getting centered onto a line off a curve. This problem I believe could be corrected in code, but I know it could be corrected in the mechanics too. As I mentioned earlier, there is 7 turning degrees. This I though at first was going to be great because I had 3 different rates to turn each way. But once I had the robot moving, I found three is not enough. When following a line on a curve the robot is going full right or left, so once a straight section come up the robot seems to have a hard to centering on it. The robot will go from the full right turn and drive right over the line causing it to go to the full left turn. To combat this I feel if you were to add two more photoreflectors to the line following board you can gain two more different rates of turning, and this would bring your different turning values closer to one another in the programming as too not "over correct" on a straight-away. As I said before there might be a way to code your way out of this, but since I am not proficient at all with coding, this mechanical fix is how I would modify the robot to fix the problem.