ECE544: Final Project Proposal

"Servo and Protecto"

Erik Rhodes
Caren Zgheib

Project Description

-Modified 5/26-

Our final project will involve programming a robot car to autonomously navigate in a given area. It will use an ultrasonic sensors to detect obstacles and edges and will adjust its course accordingly. While this is our primary goal, additional goals include using an object detection camera and controlling an apparatus when the object has been detected. In this report we include all possible configurations even though certain specifics have not been decided or are considered stretch goals.

Inputs

- Motor control
- Ultrasonic data
- Coordinates of object (from CMUCAM) via serial connection

Outputs

- PWM to motors to control bot movement
- PWM to servos for camera angle of cmucam
- Signals to sonar sensor for initialization and movement
- Signals to other externals once target is acquired

Components

- Robot kit (Arduino Robotics Kit from Oddwires.com)
    - http://www.oddwires.com/arduino-robotics-kit/
    - Arduino Uno (or Arduino Mega)
    - Ultrasonic module
    - Breadboard, circuit materials

- External batteries
- Servo x 2
- Possible extras:
    - CMUCam4
    - Servo robotic arm/claw

The Intel Galileo board was initially used, but after a day or two of experimenting with it and researching for components, we decided to not use it, primarily because there was:

- Less compatibility with libraries and shields
- Less online support
- High power dissipation (and inconvenient to require external power adapter)
- Slower I/O
- Sketches were not persistent

Design Approach

The robot car is controlled with PWM to each wheel, which we will control using the L293N motor drivers. This will be controlled with our arduino code, which will likely run in a main loop. It will periodically rotate the distance and edge detection sensor and take the appropriate action if an object has been detected. The timing behind this will be discovered with trial and error. While our turning algorithm hasn't been formulated yet, it will involve some type of logic to calculate which direction and angle to turn at. The CMUCAM will also use servos to pan back and forth. It sends the information serially to our arduino, which will then take the appropriate action once a specific object has been recognized. It can either stream the data as video or can be programmed to detect an object of a certain color and report those coordinates.

Measure of Success

In order to successfully function, the robot should be able to detect an obstacle in front of it and stop before it hits the object. At that point, it will turn the direction the object was less present in. The bot will turn a certain amount of degrees in that direction. It will then continue forward while still detecting obstacles.

We may need to set up obstacles around the room that have certain characteristics which make the sensor work fully. Being able to avoid the legs of the desks in the room, for example, will most likely prove too difficult. We will demonstrate the capabilities of our robot by letting it navigate around the room (or subsection of the room we have setup). It should be able to avoid all the obstacles we have setup.

If we complete the CMUCAM detection (stretch goal), we will demonstrate its functionality by using a ball or picture of a certain color and having it find, chase, or avoid the object.

If we are running out of time, we will consider:

- Not using the cmucam at all
- Using the cmucam just to stream the video output

If we have too much time, we will consider implementing:

- Object tracking to identify a target by color
- Stream video from car
- Have a physical mechanism that hits a target based on object tracking

Timeline

Week of 5/18

- Read through and acquire all resources for hardware components
- Create git repository for source control

Week of 5/25

- Assemble bot kit and sensor circuitry
- Run simple tests to get readings from each peripheral
- Get controls for car working
- Complete progress report
- Start integration of extra credit peripherals (CMUCAM)

Week of 6/1

- Finish logic for robot control
- Get working video feed from CMUCAM
- Check edge cases when testing functionality
- Enhance the physical appearance of the bot (including soldering)
- Complete extra credit