Robert Hill

University of Florida
Department of Electrical and Computer Engineering
EEL 5666 Summer 2002

Intelligent Machines Design Laboratory
Final Report

Table Of Contents

Abstract

For the love of two distinct things in my life, beer and engineering, I decided to build a robot that would combine both in a reasonably interesting way. It was this idea that spawned the Sloppy Gator Bot. I designed the robot to complete automate the task of finding a cup, checking if the cup is empty, and refilling the cup to the capacity if empty. Though the use of a micro-controller, various sensors and actuators, I was able to accomplish these behaviors.

Executive Summary

When it came time to investigate ideas for a robot's purpose, I knew that the best course was to involve something that appeals to myself. Thus the idea of the Sloppy Gator Bot was born. I hoped to use my engineering knowledge to create an actual beer-dispensing robot.

We were encouraged to investigate using other micro-controllers besides the Motorola MC68HC11 & MC68HC12. After researching alternative micro-controllers, I found that the MC68HC12 suited my needs best since I was completely familiar with the assembly language, had suitable amount of I/O and built-in subsystems, and also implemented fuzzy logic, which I was eager to use in my code.

For the robot's movement, two 'hacked' servos were used. These servos were driven by the 68HC12's PWM subsystem. This setup was selected since it was simple and provided flexible movement in any direction and speed.

Introduction

The purpose of this robot was to provide a practical application for robotics. My goal was to show that a robot application can have tangible goals and serve a common interest.

Once the robot is activated it begins searching for the coaster in which the cup is located. In its search it avoids obstacles and avoids the edge of the table. Once the cup is located it will check to see if the cup is on the coaster and also whether the cup is empty. If empty the robot will refill the cup to capacity without spilling.

Integrated System

I chose the Motorola MC68HC12 micro-controller as the brains of the robot. Some of the main advantages to this choice is ease in programming, fuzzy logic available, and a plethora of subsystems to interface with sensors and actuators.

The robot was programming completely using assembly language. Code was written using the MiniIDE enviornment. The programs were downloaded to the micro-controller serially though the onboard serial port.

A few problems with the MC68HC12 was there weren't enough A/D ports so I had to include an analog MUX. Also there weren't enough PWM ports so I had to use some the onboard Timer pins to create a PWM system.

Mobile Platform

The robot platform is fairly simple. It consists of three levels of round for holding the electronics, beer reservoir, wheels, etc.

The bottom level houses the two hacked servos attached to rubber wheels. It also houses the micro-controller board, the battery pack (8 AA) and various IR beacon sensors.

The middle level houses the IR sensors for collision avoidance, the prototype board with various interface circuits between the sensors and the micro-controller board. It also houses the cup holder and cup holding arms as well as the bump switches.

The top level houses the four beer can holders and associated plumbing.

Actuation

The actuation used in the robot is primarily used to move the robot around and for the purposes of filling the cup.

The servos are driven by the micro-controller though the onboard PWM and Timer systems.

Sensors

IR distance rangers are used to detect obstacles for collision avoidance as well as to check cup emptiness.

Bump sensors are used to detect object collision.

Edge sensors are used to detect the end of the table (edge of world).

An IR beacon is used in the coaster so the robot can locate the cup. A special IR sensor with filter and amplifier is used to detect the beacon.

Behaviors

The robot is normally in ‘go straight’ mode unless it detects the beacon or senses an oncoming obstacle.

Once the beacon is detected the robot tracks it until it is within a foot of the beacon. At this time the robot uses its IR rangers to position the coaster in the middle of the robot at which time the robot closes the arms around the cup and checks the emptiness of the cup. If empty, the robot will open the flow switch and monitor the level of the fluid in the cup. When the cup is full, the robot will turn off the flow switch and then open the arms and back up to a specified distance and ‘sleep’ for a few minutes, at which time it will begin again.

If the robot bumps something or detects it will back up and turn and continue its search.

Conclusion

Overall I am satisfied with the results of the Robot. For the most part the robot can complete its task. There are definite improvements that can be made in both the code and hardware, and I intend to address those issues in the future.

I am quite happy with the MC68HC12 micro-controller and would recommend it to anyone interested in building a robot. The Sharp GP2D12 IR Rangers are a must for collision avoidance, although they have excessive power usage. The Futaba servos were very reliable, and provided more than enough torque to carry the platform at a reasonable speed.

I have learned a few things from this experience. This was one of most time consuming projects I have ever encountered. After fourteen weeks, gallons of hot glue, ounces of solder, hours of debugging time, days of no sleep, and one lost job, I can say that I am one tired man.