Capital D-Bot

Jeremy Coffey
EEL 5666
University of Florida

Abstract

My robot was supposed to chase bugs and kill them. This meant that it must be able to see bugs. The chip I attempted to use for this was a Texas Instruments CCD chip. I never got it to work, and believe me, I tried. Well, if you can't kill them, join them. Or something like that. So my bot is not a bug. It wanders around, avoiding obstacles, and when it sees a person it turns and runs. It sees a person with a heat sensor.

Introduction

My main goals were to build an interesting robot and build it cheaply. My robot would have been very interesting had the CCD chip worked. It's not so interesting now that it is just a paranoid robot that runs from any person. I think I accomplished the second part, using things I had laying around to build my robot.

Platform

I used the T-Tech machine in lab to cut out a tear-drop shaped body and some large wood wheels. The wheels are 5.25" in diameter, with rubber bands glued around the wheel for traction. A free-spinning wheel is on the tail of the body.

Brains

The "brains" of the robot are in a Motorolla M68HCll with the ME11 expansion board from Novasoft.

Sensors

Infra-red Sensors. Infra-red emitters and sensors allow for collision avoidance. The emitters receive a 40 kHz square wave from the ME11 board which emits IR without burning the LEDs. The sensors were hacked to get an analog signal out, which is much more useful for my purposes. They have a nominal value of about 85 and go above 90 or even 100 when close to a wall. These values are measures through the analog-to-digital converters available on the M68HC11.

Bump Sensors. Four push-buttons on the front of my robot, connected to a thin piece of wood, cut to fit the shape of the front, form my bump sensor. These push-buttons are pulled high and connected to input pins. I added 16 input pins by adding two '573 IC's. The multiplexing needed to use these chips were already done in the ME11 board. Why I added 16 pins and not just 8 will be revealed later. This sensor works surprisingly well. Even more surprising is it's durability with my masking-tape design.

Vision. An incredible attempt to use a Texas Instruments TSL1301 CCD chip to "see" bugs. At first I tried just using the M68HC11 but I am using IC which does not give me full control over the timing and the timing for this chip was critical. Currently enrolled in EEL 4712, I know how to program a PLD using VHDL. I programmed a GAL to control the CCD chip which just took two inputs from the M68HC11 ('clock' and 'start'). The GAL then told the M68HC11 when to read from the A/D converter I had added to the board. This A/D converter had 8 converters built-in and you can select which signal you want to read through a three-bit multiplexed code. The A/D converter had to be faster than the ones already available, and besides, I had already used them all up on the M68HC11. It is the connections of this A/D converter that created the need for adding 16 additional input pins. This sensor never came to fruition. It only came to crap.

Low Power Light. Connecting power (positive voltage) to my A/D converter that I added to the board, I was able to monitor the voltage level to see when power was getting low. I used an LED as the indicator. Since I was reading from the +5V powering the chips and not directly from the battery pack it also served as a noise sensor. When the motors were working hard the LED would come on, then go off again when the motors weren't dropping the voltage as much. Since the A/D converter I had added to my board died the night before demo-day I no longer have this sensor. I did not blow this chip up. It just stopped working.

Heat Sensor. I hacked a home motion activated light. Ripping out the circuitry, it was easy find power and ground. I gave the board power and tested the output pins of the op-amp on the board with a multimeter. Eventually I found the pin that was outputting the correct readings. A rolled-up piece of paper fitted around the little heat sensor on the board acts as a barrel to only sense heat that is directly in front. The heat sensor's analog output is then read by the M68HC11's A/D converter. The nominal value for this sensor is around 128 and it goes above 140 or below 115 when heat is picked up. The circuitry already in the heat sensor calibrates itself automatically so it will work well despite the temperature of the room. This sensor works well.

Behaviors

My robot wanders around, avoiding obstacles. Some techniques were used to smooth most of the turning, stopping, and accelerating. When heat (a person) is sensed the robot no longer moves slowly and smoothly, but quickly whips around, doing a 180-degree turn and takes off full speed, for about five feet or until it encounters an obstacle. When a person is detected it also lights up an LED for a few seconds. This same LED is used during an initialization sequence just to show that it is working properly.

Conclusions

I was very disappointed that all the time and effort I put into that stupid CCD chip never paid off. My classmate Andor Almasi only got his CCD chip (the same kind I was using) to detect ambient light, but nothing close to recognizing objects. Overall I enjoyed working on the robot, especially when it actually started doing things. What little it does, it does well. With my add-on A/D converter breaking the night before demo-day I had to re-work some things on my robot. I had to remove an IR sensor to free up an A/D converter for the heat sensor.