Date: 8/5/2003
Student Name: Frederick Reitberger
TA: Uriel Rodriguez
Louis Brandy
Vinh Trinh
Instructor: A. Arroyo

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

Written Report 3

Table of Contents

Abstract

The Treasure Detection Robot hunts for treasure. It uses a metal detector to find coins and other small metal objects, which it values as treasures. The thrill is in the hunt, though, so this robot does not attempt to collect the treasures. If it were to collect the treasures, then it would get bored and lonely once it has found all the treasures that it can find. Once it finds treasure, it performs a little happy dance in order to attract attention to itself. The Treasure Detection Robot wears an eyepatch and a pirate's earring as it searches for buried treasure.

Executive Summary

The Treasure Detection Robot is a robot that hunts for treasure. A metal detector is used to find metal objects and other treasures. Bump sensors and IR sensors are used to avoid collisions. Twin motors are used to drive around the room. A PIC microprocessor ties everything together. All the controlling software is written in Forth. When treasure is detected, the Treasure Detection Robot will alert those around it by performing a short dance and blinking its LED. After detecting treasure, the Treasure Detection Robot will wander elsewhere looking for more treasure.

Introduction

This robot is the Treasure Detection Robot, or TDR for short. TDR is a treasure hunting robot. Its job is to hunt around looking for treasures. Treasures are small metal objects such as coins. TDR will determine where treasures are hiding and inform the world that it has found treasure. The rest of this paper focuses on the major systems that are required to implement TDR.

Integrated System

The brain of this robot is a Microchip PIC 16F877A processor. This processor runs at 20 MHz. All of the code has been written in Forth. I chose Forth because the development tools are fairly good and Forth is a language that I have never used before. Forth is also well suited for microcontrollers. A JDM style serial programming cable was used to program the processor in-circuit. The driving motors are a twin DC motor gearbox. The main sensors that were used are several bump sensors and IR detectors for collision avoidance and a metal detector to find treasures. The main behaviors are collision avoidance, the searching pattern, treasure detection, and announcement when treasures are found.

Mobile Platform

The platform is a circular platform similar to the TJ robot. I purchased a circular robot platform to greatly simplify the design of TDR. This chassis is useful because it can be driven with two wheels easily and already has holes for the tires and mounting points for the ball-bearing caster and the twin DC motor gearbox. I used AutoCAD to design and build a cradle to attach to the underside of the plastic chassis so that the metal detector search coil is held a small distance away from the motors and other parts of the robot.

Actuation

The twin motor gearbox is a set of two independently controlled DC motors and gears connected to an axle. These motors are small DC motors that operate at 3V to 5V. The gearbox can select gear ratios of 58:1 or 203:1. The higher ratio delivers more power at a reduced speed, so that is the ratio that has been used. To drive these motors, I have a serial motor driver IC. This motor driver listens to an input line for a specific sequence of serial data, then will adjust its outputs according to the commands it received. One motor driver directly supports two motors and 254 forward and reverse speeds for each motor.

Sensors

Bump switches are situated a the edge of the robot chassis. These are simply pushbuttons that will be pressed when the robot runs into an obstacle. These will let the robot know that it cannot continue in the direction it was originally traveling and it must turn. These sensors are connected each to a different input port that has an internal pull-up resistor.

Two IR detectors are used to identify obstacles before the robot runs into them. Each consists of an IR emitter and IR receiver pair. Each IR detector has an internal oscillator that is used to generate a frequency that the emitter will emit at and the receiver will listen for. The receiver then outputs an analog voltage that is dependent on distance an object is from the emitter. These analog values are read through two analog to digital conversion ports to turn the analog values into digital signals that the robot can act on. The IR detectors are mounted on the sides of the robot and point slightly inward to behave like eyes.

The metal detector is the most unique sensor on this robot. A metal detector has been built from scratch that will detect coins about an inch under it. The metal detector works be amplifying an oscillator built around a LC circuit. The inductor in the oscillator is the search loop. As metal passes near the search loop, the inductance of the loop changes and the frequency of the oscillator changes. The oscillating output is then amplified and fed into a 74HC04 Digital Inverter. The inverter is set up so that it will be triggered by the metal detector output and produce a good digital output signal. This output signal is read into the microprocessor by a 16 bit asynchronous counter. When searching for treasure, the count will be cleared and the microprocessor will wait for a specific amount of time, then read how many counts happened in that time. This will give a specific value for a frequency. In order to deal with error and noise, the two most recent readings are averaged together to smooth out the results.

Behaviors

The first behavior that is required is collision avoidance. Bump sensors are used to determine if the robot has collided with anything, and if it has, then it will back up, turn away from the object, and then continue forward. IR detectors are used for this as well to find large objects before bumping into them. Again, when an obstacle is detected, the robot will turn away from the robot so that it will not hit it. Since the IR detectors work at a distance, they are more effective in keeping TDR from running into objects.

Another important behavior is the search pattern. TDR randomly moves for a short while before starting a spiral searching pattern. This will continue for a while before TDR gets bored, then randomly moves to another location to begin the spiraling search pattern again.

While traveling the search patterns, TDR will be listening to the metal detector. When it detects treasure, it will stop for a bit and announce that it has found treasure by performing a happy dance and blinking LEDs. Then it will continue on in the search for more treasure.

Experimental Layout and Results

Experiments were performed to determine how effective the collision avoidance algorithm is. The algorithm works by checking to see if anything is too close to TDR by looking at the IR sensors. If something is seen in only one of the IR sensors, then TDR will turn away from that side. If something is seen in both sensors, then TDR will back up until it does not see anything nearby, and then veer to the left or right randomly. Every few seconds TDR will decide on a new direction to travel. This is fairly effective in keeping the robot from running into anything, although it can still be trapped in a corner and not figure out how to get out. This situation has not been seen in testing though.

The metal detector search oscillator oscillates at about 85 kHz normally. When a quarter is placed on top of the search loop the frequency increases about 200 to 400 Hz. When the quarter is on the ground underneath the search loop, the frequency change is smaller – on the range of about 100 to 200 Hz. When using discs of carbon steel, the frequency changes are slightly higher, but on the same scale. Experimentation with the PIC microprocessor reading the frequencies was less successful. The readings varied widely for a while and once they settled down, the changes by bringing quarters or carbon steel were much more dramatic. An average reading when no metal was present would be about hex $2080, while bringing metal nearby would change the reading to hex $2A00 or higher. Averaging subsequent readings improved the accuracy along with requiring two positive readings in a row before declaring that treasure has been found.

Conclusion

Most of the goals set for the Treasure Detection Robot have been met. TDR can wander around a room looking for treasure, and if it is calibrated correctly, then it can reliably determine where treasure lies. Calibration is performed in software before starting the searching, but does not always properly work. The metal detector was the most challenging aspect of the robot. The variation in the metal detector makes it incredibly difficult to get accurate readings consistently. A more consistent or easier to use metal detector would simplify this design greatly.

Learning and using the Forth programming language has been both a challenge and a reward. Once I got used to the Forth syntax, I was able to write complex routines quickly. It does a good job of acting like a high level language yet leveraging the power of assembly code. However, there were a few cases where writings routines in assembly might have been easier. Specifically, any routines where I attempt to perform math functions on 16 bit values. My coding style also probably needs improvement, but it works and that is the important thing.

If I were to start this project over again, I would probably use a larger platform that has more space for the things I need. The platform I used was quite small which led to difficulties trying to place all of the parts for the robot, such as the IR sensors, battery pack, and any future additions. I also might use an LCD display for calibration and status feedback. To enhance this robot a sound chip could be used to provide audio feedback and an LCD display could also provide useful feedback. A redesigned platform could allow the metal detector to be close to the front of the robot so that metal can be detected sooner, rather than having to drive over the metal before it can be detected.

Documentation

http://www.elenco.com/user%20manuals/k-26.pdf – Simple Metal detector

http://www.olimex.com/dev/pic-p40b.html – Processor Board

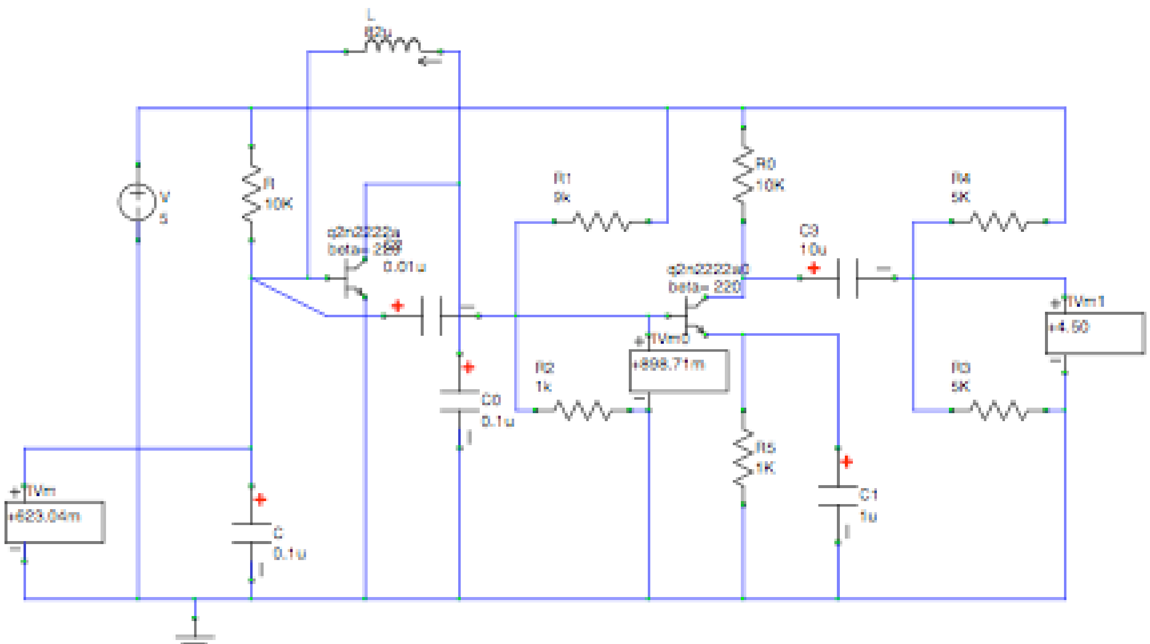
http://www.pololu.com/products/pololu/0250/index.html#combinations – Motor kit and chassis combination

http://home.clara.net/saxons/bfo.htm – Simple BFO metal detector

http://www.rfc1149.net/devel/picforth – PIC Forth Compiler

Appendices

Metal Detector Circuit:



Metal Detector Simulation: