Final Report: Project # 2

Team 3:

Mark Branson
Amit Mathur
Matt Roman
Mike Taylor

March 31, 2003

Section 1

Hardware Description

1.1 Design

The overall design constraints or goals for our robot are listed below. These have been agreed upon by all of the group members before construction began.

- Small compact footprint
- Zero turning radius capability
- Rigid structure for sensor mounting
- Ground clearance for small obstacles
- Geared for speed

Each of these goals was met in our final construction of the robot.

The 6.5 x 6.5 in footprint was considerably smaller than other robots that we had the chance to observe. This allowed us the chance to maneuver around obstacles with greater ease because the robot could fit in tighter spaces. This also decreased the chance of running into an obstacle while turning.

The zero turning radius was a very important goal that needed to be achieved in order to simplify our code for easier positioning.

A rigid construction was necessary incase of running into any of the large rocks in the course. We wanted to minimize the chance that part of our robot could fall off in a collision. Fortunately the wheels we chose to use were over sized and would absorb most of the shock from a crash.

The over sized wheels also allowed us to increase the ground clearance in case we needed to drive over smaller objects. A large wire ran power to a single light in the middle of the course and could have caused problems for a robot with a very low ground clearance.

The robot was geared more for speed than torque because we wanted to avoid obstacles rather than move them. The higher speed would also allow us to visit as many lights as possible with in the 10 minute test time.

1.2 Motors and Gears

The robot only used two motors to drive and steer to reduce the amount of power needed to move. Four motors probably would have drained the batteries too fast and left our robot unreliable in the final minutes from false sensor readings due to low battery power. Because the motors by them selves cannot produce enough torque to move the robot they have been geared down 5:1 to the wheel using an 8-tooth pinion on the motor and a 40-tooth spur on the axel. At this ratio we can still move at a higher speed with out a large sacrifice in torque.

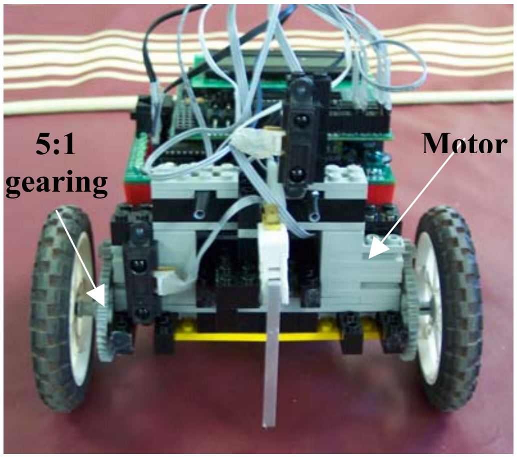


Figure 1 front

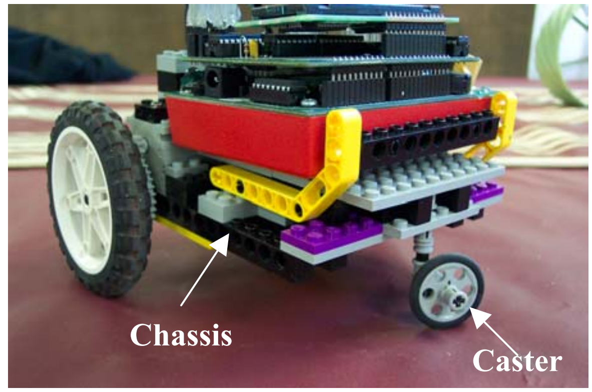


Figure 2 back

The figures above describe the layout of the robot, in figure two the free caster is visible it keeps the robot standing upright. The two large front wheels are the drive wheels. All the components have been mounted to a rigid rectangular chassis in which the handy board sits on.

1.3 Sensors

The sensors on our robot were decided on by the combination of the hardware and software teams. They decided to use 2 light sensors, 2 IR range sensors, and a single switch. The arrangement can be seen in figure 3 the two circles indicate the light sensors. The switch on the front of the robot was used as a bump sensor to tell the robot to escape if it collided with something. The two IR sensors were rearranged throughout our testing; one IR was moved higher to decrease the sensitivity to smaller objects.

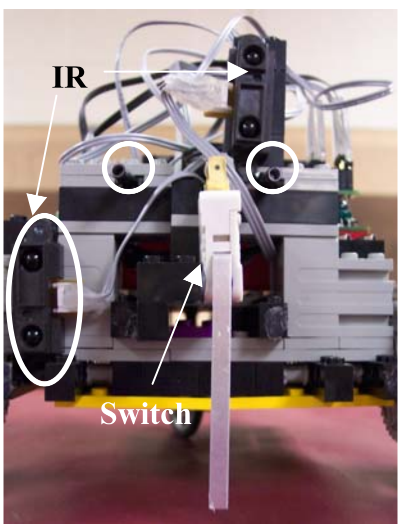


Figure 3 sensors

Section 2

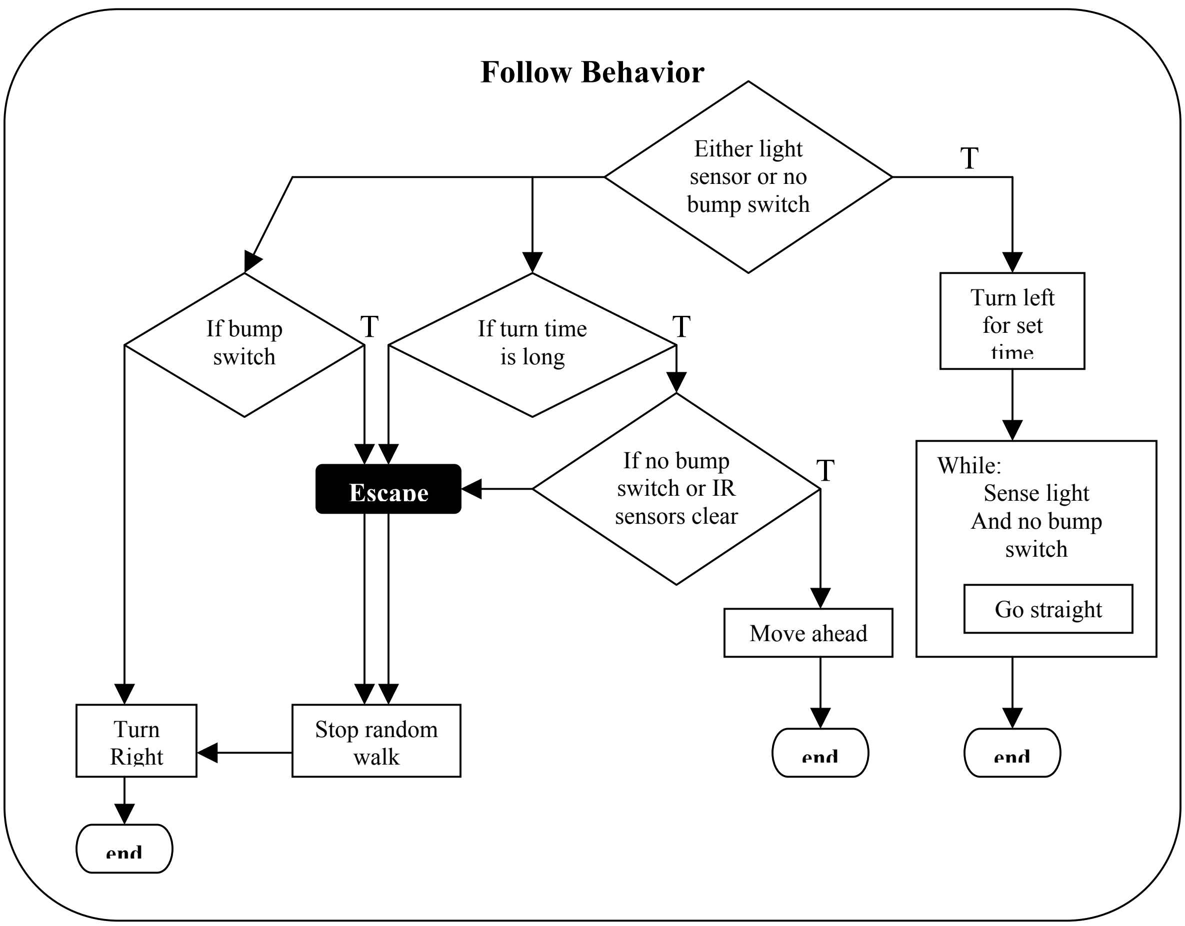
Software Design

2.1 Introduction

The software design was first based on the Lewis and Clark program from *Mobile Robots* (Jones et.al.). Each module of the code was checked and ran properly but when all were combined and run on the handy board it was not responsive enough. We believe the problem was in running multiple processes; this severely reduces the amount of computation time for each module.

2.2 The Code

The final code for our robot has been simplified from the multi-process example above. The main function calls out the behavior function follow() which consists of a light following routine and an escape function incase the bump switch is triggered.



The robot functions as described above in the Follow Behavior flow chart. This behavior is continually running in an infinite while loop, at each **end** the behavior begins again with new sensor values. The robot starts out making a counter clockwise spin until it detects a light then moves forward toward the light. If the robot has been spinning for a while without detecting a light it begins a random walk, it will move forward for a random amount of time while sensing for obstacles then stop and search for a light.

The escape function will only be entered if the bump switch is pressed or the IR sensors are viewing a tall obstacle. The function reverses the motors for a set amount of time to back away from the obstacle then turns until the IR sensors cannot see any hazards. The robot will move forward for half of the time it reversed and return to the behavior function.

2.3 Conclusion

Overall the code was very simple and used a random search if a light was not detected. More time was needed than anticipated to calibrate the light sensors and adjust the code to work properly in the test environment. A very important lesson was learned when the batteries get too low the firmware on the handy board will become corrupted and must be reloaded in order for the hard ware to work properly.