Introduction

PURPOSE & GOALS
• To show the flexibility of the Montana State University ECE Robot platform and the versatility of the included parts.
• To solve a maze using any number of robots to find a suitable path to a defined goal.
• To share the exploration wirelessly from robot to robot using a peer to peer network structure.
• Compensate for cheap hardware using sound design principles and active control software running on the robot microcontroller.
• Achieve all tasks under a $500 parts budget for two separate working robots.

ECE 101 ROBOT PLATFORM
• To expedite the development of the robot, the ECE101 robot platform was used and modified using nearly all components included in the original kits given to the students of EE101. This also allowed the team to reuse code originally written for the robot kits.
• Parts that were added to the robot included 3 Infrared Distance Sensors to detect forward, left, and right walls., and 2 short range infrared sensors used to read the encoder on each wheel.

IEEE MICROMOUSE MAZE
• The maze that was chosen for the robot was the IEEE Micromouse maze. Although the official rules of the maze were not considered when the project was chosen, using the maze allowed for development of the robot and not the environment of the robot.
• The Micromouse maze was too small for the original ECE Robot, but modification of the body allowed it to run inside of the walls.

Size Constraints

IEEE MICROMOUSE MAZE
• The maze that was chosen for the robot was the IEEE Micromouse maze. Although the official rules of the maze were not considered when the project was chosen, using the maze allowed for development of the robot and not the environment of the robot.
• The Micromouse maze was too small for the original ECE Robot, but modification of the body allowed it to run inside of the walls.

Maze Solving

LEFT HAND WITH DEAD END REMOVAL
The algorithm chosen was one that would be easy to implement, take minimal memory, and allow for multiple robots to explore at a single time. To accomplish this task two methods were used. First, the robots would mark paths taken through the maze as an explored pathway. By having the robots always choose an unexplored path when possible, a spreading effect would happen with multiple robots in a single maze. If all paths were taken from a cell, a simple left hand wall follower was implemented.

The final step was to have the robot close off dead ends without trapping another robot. This was accomplished by creating a “One Way” wall in the maze. To create a one way wall, a virtual wall is added on the opposite side of a cell as the one that needs to be closed off. This way if you are in the closed off cell, there is an exit, but when looking back at the cell the robot will “see” a wall, like a one-way mirror.

STORING THE MAZE
• The maze is stored in a 17x17 matrix for both robots. After the robots traverse a single cell it will send the coordinates and value for the cell from which it came. And the surrounding cells.
• The other robot will receive the message and apply the logic for walls and exploration to the newly acquired cell.
• Below is one of the three byte data packets sent every cell.

Wireless Communication

FREESCALE ZIGBEE MODULE
• Thanks to donations from Freescale, the Zigbee module was able to be used while maintaining our budget constraints.
• By using the supplied modules existing C libraries of functions were incorporated into the codebase further expediting the development of the project.
• To receive data the Zigbee module is continuously polled for the detection of newly received data.
• To send data, a pointer is loaded and a flag is set. When main comes back to the function it will send the data without overwhelming the Microcontroller.

Members:
Josh Smith
Monther Abusultan
Clint Gramza
Colin Shirley
Advisor - Randy Larimer
Special Thanks To:
Freescale
IEEE
MSU ECE Department

Orientation and Position

MAINTAINING ORIENTATION
• One issue with running multiple robots that are turning is keeping orientation correct and aligned. This problem was solved by storing the initial forward as X (North) in the matrix and Y (East) in the matrix. Then when a robot moves through a cell it keeps track of the turns to maintain orientation.
• To make the robot start from any corner facing any direction inwards. The robot will flip the Y axis depending on the first non-wall it encounters.

DRIVING STRAIGHT
• To keep the robot driving in a straight path 2 methods were developed to compensate for the slow, inaccurate motors.
• First, encoders with 48 divisions/rev where added to each wheel and read with a photo-diode. This encoder increments a counter on each wheel that is used to compensate for different rotational speeds.
• Second, the wall sensors do double duty by measuring the distance to the walls and adjusting the robot so that they are equal.

Zigbot
Small Wireless Communicating Maze Solving Robots

Members:
Josh Smith
Monther Abusultan
Clint Gramza
Colin Shirley
Advisor - Randy Larimer
Special Thanks To:
Freescale
IEEE
MSU ECE Department