Work in Progress - Undergraduate Interdisciplinary Research – IEEE Micromouse Competition

Robert D. Milos, Allan J. Hall, Neal Shine, Kyle D. See, Tyler R. Bednarz, Yonglian Wang*

Department of Electrical and Computer Engineering and Computer Science

Ohio Northern University

r-milos@onu.edu, a-hall.6@onu.edu, n-shine@onu.edu, k-see@onu.edu, t-bednarz@onu.edu, y-wang.2@onu.edu,

***Abstract* - This paper presents the research project conducted by a team of five undergraduate students for the IEEE Region 2 Student Activities Conference. A micromouse is being designed to find a predetermined destination when placed in an unknown maze in the shortest interval. The micromouse is required to meet the specifications provided in the 2010 Micromouse Competition Rules. This project applies the knowledge from different fields, such as computer science, computer engineering, electrical engineering, and mechanical engineering. It is a great opportunity to utilize interdisciplinary theoretical fundamentals to create a physical project. This research is carried out in five phases: algorithm development with high-level programming simulation, choosing parts based on specifications, part testing, construction of the micromouse, and integration of components. The micromouse employed the flood algorithm to find the shortest path to the destination. A Java code simulation of the micromouse's behavior is created to verify the algorithm and then is translated to C code to be implemented on the microcontroller. Testing is conducted on the components individually and implemented on the constructed chassis.**

Index Terms – Flood-Fill Algorithm, IR Sensors, Micromouse, Microprocessor, Stepper Motors

1. INTRODUCTION

A micromouse is a completely autonomous robot designed to find a predetermined destination when placed in an unknown maze in a shortest interval. The maze is made up of a 16x16 grid of cells. Each cell is 18cm square with walls 5cm high. The official 2010 Micromouse Competition Rules can be found in [1]. Additional information regarding the construction of a micromouse can be found in [2]. To accomplish the task, the micromouse needs to map the maze through intelligent exploration, and then track the optimal (shortest) path which will allow the mouse to run in the shortest possible time.

This research, launched by the Honors Program of Ohio Northern University (ONU) and sponsored by College of Engineering and IEEE ONU student branch, aims at maintaining the research interest and curiosity of outstanding undergraduate students through experiential learning. More importantly, this research provides the team a chance to apply their interdisciplinary knowledge to a tangible use, which is one of our educational goals. Among the five undergraduate students, Allan (Electrical Engineering), Robert (Electrical and Computer Engineering), and Neal (Electrical and Computer Engineering) are junior students; Kyle (Computer Engineering) and Tyler (Electrical Engineering) are freshmen students. The junior students focus on high-level programming, debugging, simulation, and guiding the freshmen engineers as project leaders while the freshmen students focus more on micromouse chassis design, choosing parts, and parts testing (e.g. sensors). This research is being carried out in five phases: algorithm development and high-level programming simulation, choosing parts based on specifications, part testing, construction, and integration of components. The rest of this paper describes how to implement the design from scratch and how part of the finished design should look like. It is organized into the following: part 2 Algorithm Development and High Level Programming Simulation; part 3 Parts Selection based on Specifications; part 4 Part Testing; part 5 Construction of the Micromouse; Part 6 Conclusion.

2. Algorithm Development and High Level Program Simulation

The algorithm employed to locate in finding the shortest path on the microprocessor is called the Flood-Fill algorithm [3] which models the map being filled with water from a start square. When the maze is slowly filled with water, the shortest path will be taken by the first drop of water to get to the center. The algorithm involves three stages. The first stage assigns each cell a distance from the current position of the micromouse until the center cell has been given a distance. Prior to entering the maze, the micromouse has no knowledge of the maze, so the most direct path would be straight up and straight right. The second stage is similar to the first, but backwards. It starts at the center finding adjacent squares with a distance one less than the current square, recursively. In this manner, the micromouse finds the shortest known path to the center. The third stage involves moving the micromouse a unit cell along the path and using the sensors to detect new walls. The micromouse then starts at stage one with the new information.

The algorithm was first implemented using a high level programming code (Java), which allowed us to test the algorithm without a physical micromouse and verify its functionality. The Java simulation was created to simulate the movement of the micromouse. The simulation was graphical in nature, and used the same principles of the Flood-Fill algorithm. Also a maze editor was developed that would allow testing the micromouse in many different situations. The simulator that was developed is shown in Figure 1 with numbers added to indicate the first stage of Flood-Fill algorithm at the initial setting.

3. Parts Selection Based on Specifications

The step following the implementation of the flood-fill algorithm in high level language is to select the appropriate components to create a physical realization of the simulation. The basic integration of the micromouse components are shown in Figure 2.

The most vital device in the micromouse is the microcontroller. The controller and its corresponding development board are the control unit for motion, sensor processing, maze mapping, and low level algorithm storing. The criteria used to select the microcontroller include size constraints, system frequency, available ports, analog-to-digital converter (ADC) capabilities, and memory. Based on the previous specifications, the 8-bit Atmel ATmega64 microcontroller is chosen. The Atmega64 has 64KB of programmable flash and 2KB of EEPROM which makes it sufficient to store the algorithm as well as contain the array needed to store the map of the maze. In addition, the Atmega64 contains a 10-bit ADC capable of processing the analog inputs from sensors. Finally the Cerebot development board from Digilent© at 8 cm^2 fits into the desired size constraints and include JTAG interfacing and C-code development software. Those features make it easier to convert the high level language and to implement it on the microcontroller.

Next, appropriate motors and sensors to interface with the selected microcontroller were chosen. The high level algorithm simulation did not incorporate the inherent errors present in the physical implementation of the micromouse design. Thus, the major factor governing the selection of both the sensors and motors is their capability to provide precise input and output values. Furthermore, the ease of integration with the microcontroller should be assessed to simplify design and testing.

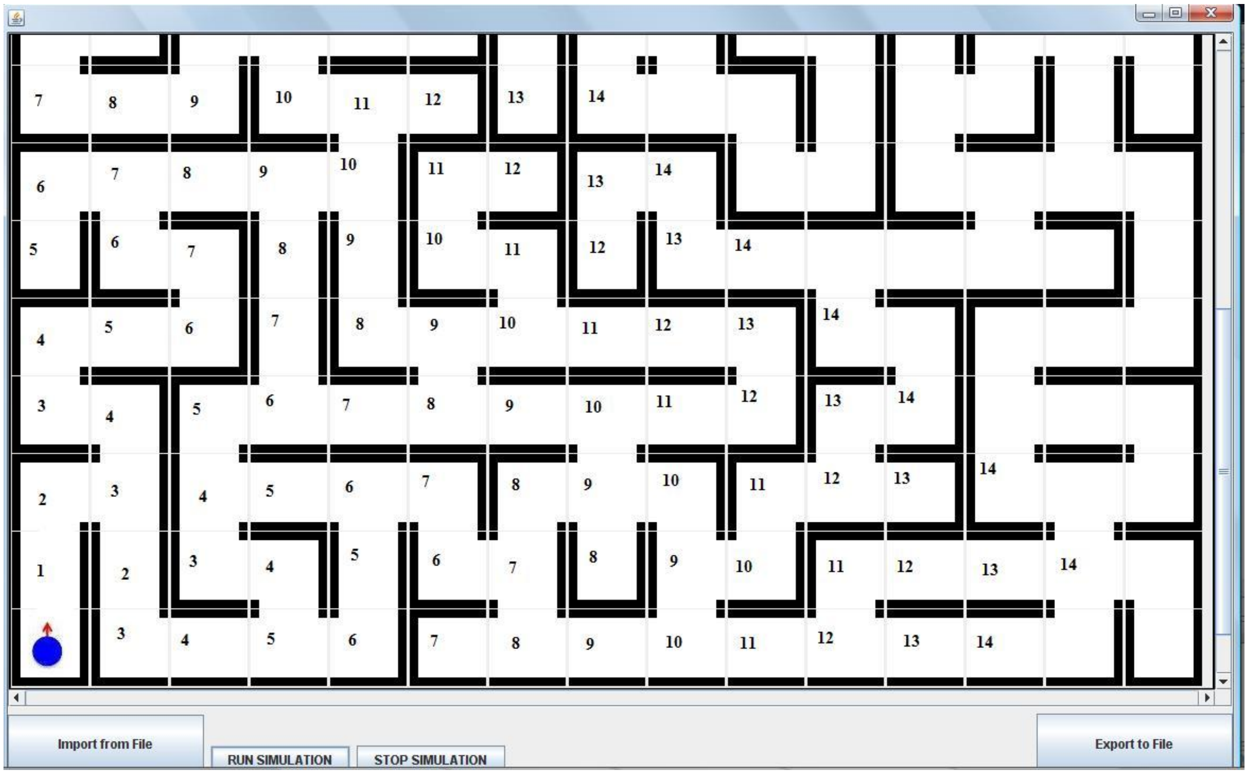

Figure 1. Java simulation with implemented Flood-Fill algorithm

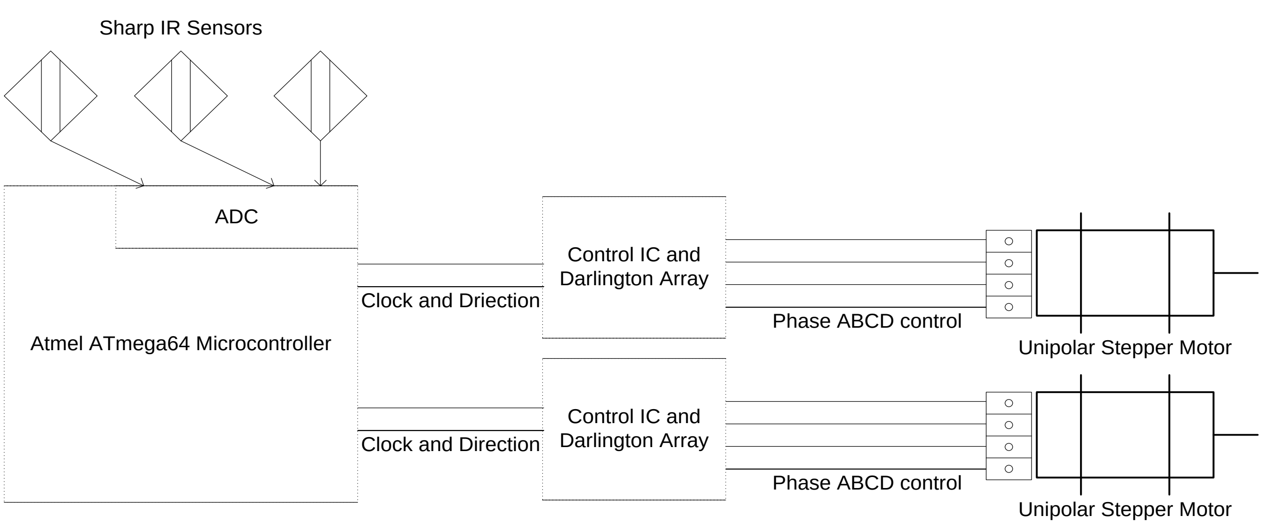

Figure 2. Micromouse Elementary Diagram

Two Nema 17 frame unipolar stepper motors are chosen to meet constraints based on system clock of the microcontroller. Thus, the exact amount of distance is deterministic so that the micromouse is ensured to move to the center of the next maze unit cell. The selection of stepper motors illustrates the design's preference of accuracy over speed which could be achieved with servo motors. In addition to being accurate, the Nema 17 frame motors' small size, weight, and torque capabilities make it feasible to incorporate them into the design.

Three Sharp GP2D12 Infrared (IR) sensors are chosen to provide reliable analog voltage output based on small distances from the sensor to the reflective surface. The IR sensor consists of a transmitter (a single LED and lens) and a receiver (a lens and a position sensing device). These sensors allow sensing on all grades of reflective surface and have a strong correlation to their specified output voltage to distance characteristics. Due to their accuracy, these sensors will not only detect the presence of a wall, but also allow position correction that is based on the detected wall distance.

4. PART TESTING

Part testing follows the component selection. Tests are performed on the IR sensors to analyze the trend of voltage output versus distance from the sensor. The sensor is set at measured distances from a wall modeling the maze. Voltage outputs are then recorded and plotted against the distance between the sensor and the wall. Figure 3 exhibits some testing results.

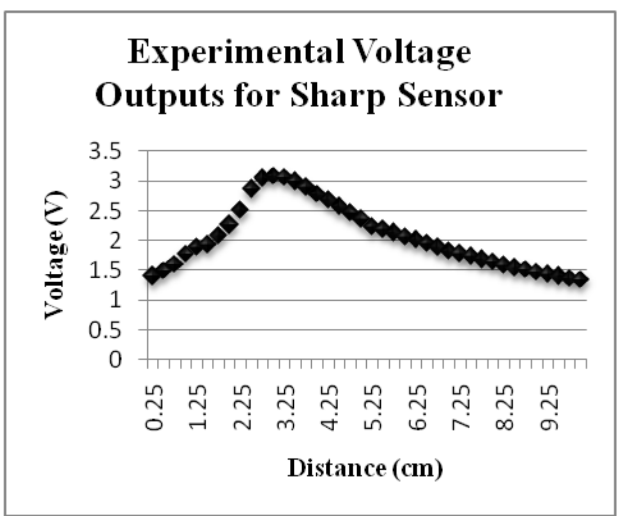

Figure 3. Laboratory test results of distance versus voltage for Sharp distance sensor

Based on the shown experimental results from testing the sharp sensor, the decision is made that the sensor will always be 3 cm or further from the wall. This design decision keeps all voltages to the right of the voltage peak, satisfying the assumption that the sensor will never be closer than 3 cm to the wall.

The testing results demonstrate that multiple sensors of the same type have a variance between them. However, the variance is a fixed offset value allowing for a basic calibration change for our design. Tests are also performed to determine variance that can be caused by different walls, light or dark, finished or shiny. In those experiments no substantial variance is found.

With the assumption that the sensor will always be 3 cm or further from the wall, the allowable tolerance is only taken into consideration in implementing an analog to digital converter for the sensors. In this case whether a wall does or does not exist become very important. During the testing, some techniques are utilized such as constructing an integrated circuit to handle the actual motor control and using transistors as a current amplifier to increase torque of the motors.

5. CONSTRUCTION OF THE MICROMOUSE

After testing is completed for the individual components, construction of the chassis begins. The design of the Micromouse is entirely based on the physical restraints of the maze. Not only does the robot have to be supported on a very small footprint, but the height of the wall is also very limited (5 cm), which greatly limited the position of the distance sensors in relation to the wheels. As the wheels have a diameter of over 5 cm, the sensors can't be placed over the wheel either. As a solution to this problem, the sensors are mounted in a vertical orientation in front of the wheels. A forward facing sensor is also designed to be mounted on the forward support beam that connects the top of the chassis section with the bottom chassis section.

Parts selection for the chassis is determined by the availability of materials in the campus machine shop. Per the recommendation of the shop supervisor, Plexiglas is chosen as the building material for the chassis. The chassis design is a top plate and a bottom place connected by two large vertical supports. The motors are mounted using custom fabricated brackets between the two vertical supports. These brackets are made of metal and drilled to ensure alignment of the two motors. The Plexiglas is cut, drilled, and assembled. In joints that connect metal with Plexiglas, nuts and bolts are used for added strength and stability. Connections between

Plexiglas sheets are facilitated by high strength, fast setting glue.

Due to the physical size of the microprocessor, it is mounted on top of the top mounting plate. This makes it the highest item on the mouse, standing well above the walls. In consideration, given the size of the plates, the top plate is made larger than the bottom to allow for mounting room for the microprocessor. The bottom plate is built to be large enough to support the motors, but as small as possible to decrease the footprint of the robot in the maze.

6. CONCLUSION

A research project of micromouse design for the IEEE Region 2 Student Micromouse Maze competition has been carried out by five undergraduate students. So far, 80% of this undergraduate interdisciplinary research project has been completed though chip verification and integration of components are still in process. Each phase of the project has been an opportunity for students to apply their engineering fundamentals and problem solving skills. Furthermore, the junior students have been able to gain leadership experience through mentoring the freshmen students and encouraging them to become more interested in their engineering study. The project provides great experience for every team member in terms of team cooperation, knowledge transformation, and commitment. The successful completion of the project will greatly improve the team members' interests in studying and working in the engineering field.

REFERENCES

[1] "Micromouse 2010 Competition Rules." *IEEE Region 2 Student Activities Conference 2010 Web Page*. Web. 21 Nov. 2009. http://www.temple.edu/students/ieee/SAC/competitions.html.
[2] *Micromouse Book | Micromouse Online*. Web. 08 Jan. 2010.
[3] "Micromouse - Maze Solver - Theory | Member Robot Tutorials." *How to Build a Robot Tutorial - Society of Robots*. Web. 17 Dec. 2009. http://www.societyofrobots.com/member_tutorials/node/94.
[4] "Mobots : Micromouse 2006." *Welcome to Mobots*. Web. 17 Dec. 2010. http://mobots.solarbotics.net/micromouse06.html.

AUTHORS

Robert Milos is a junior electrical engineering and computer engineering student at Ohio Northern University. He is the treasurer of the IEEE student branch.

Allan Hall is a junior electrical engineering student at Ohio Northern University. He is the secretary of the IEEE student branch.

Neal Shine is a junior electrical engineering student and computer engineering student at Ohio Northern University.

Kyle See is a freshman computer engineering student at Ohio Northern University.

Tyler Bednarz is a freshman computer engineering student at Ohio Northern University.

Yonglian Wang is a Visiting Assistant Professor at Ohio Northern University. She teaches Microprocessors, Embedded Computing Systems, Advanced Electronics, Analog circuits, and Control Systems. Her research interests include speech signal processing and recognition, FPGA design, and Embedded Systems.