

Beamos

Final Report

Research Experience for
Undergraduates

University of Florida
Machine Intelligence Lab

Prepared For:
Dr. A. Arroyo
Dr. Schwartz

By:
Clint Doriot
August 4, 2005

TABLE OF CONTENTS

Introduction	3
Integrated System	3
Mobile Platform & Actuation	5
Sensors	6
Behaviors	10
Conclusion	12
Source Code	13

TABLE OF FIGURES

1. Beamos Structure Chart	4
2. Platform	5
a. Mobile Base	
b. Sensor Mount Head	
3. Mavrik-IIB Microprocessor Board	6
4. OPB745 Sensor	7
a. OPB745 Sensor Package	
b. Required Circuitry	
5. Line Following Circuit	8
a. PCB Diagram	
b. Finished PCB	
6. Sonar Module	9
a. Circuit Board	
b. Operating Range	
7. AVRcam	10
8. Maze Environment	11

INTRODUCTION

Beamos is meant to simulate a predator in a predator-prey relationship by tracking and finding the prey within a maze. Beamos is meant to continually track the target destination as it moves throughout the maze. Therefore, the maze will contain no walls but will instead be mapped out using white lines on a black surface.

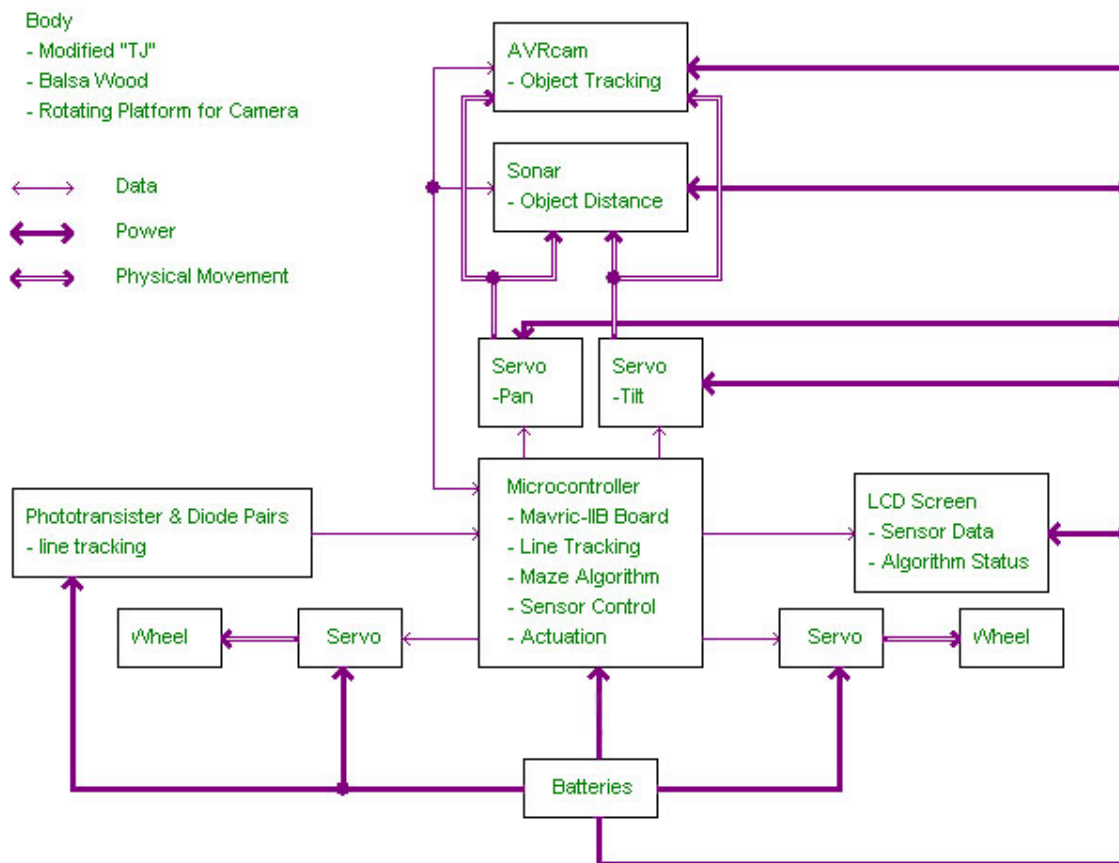
INTEGRATED SYSTEM

Figure 1 is a structure chart that shows how the component subsystems of Beamos will interact to provide the functionality he requires to accomplish his goals. These components are summarized below:

- A simple mechanical body provides a compact structure to house the various components.
- Two servos will provide the pan and tilt for the camera/sonar system.
- Two servo controlled wheels provide mobility and several casters create stability.
- The camera provides angular position data allowing Beamos to track the destination object, his “prey”.
- A sonar relays distance information so that Beamos knows how far he is from his target.
- Multiple emitter / phototransistor pairs communicate information about the environment, so that Beamos can move throughout the maze with out running off track.

- An LCD screen provides a textual interface for displaying information. This is especially important for troubleshooting and testing.
- Batteries provide the necessary power so that Beamos is not dependent on a stationary power source.
- Beamos' microcontroller ties these subsystems together and allows him to track his prey through a maze.

Figure 1 – Beamos Structure Chart

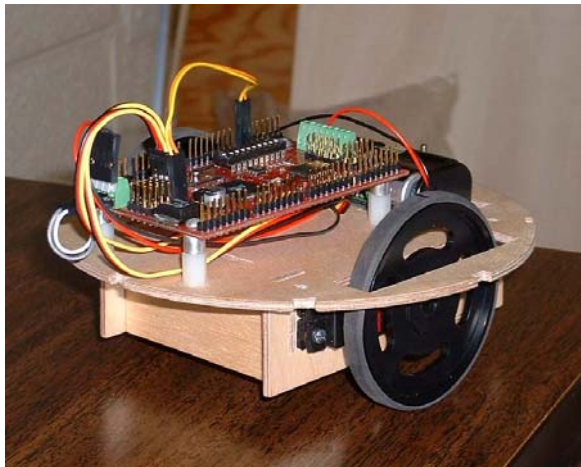


MOBILE PLATFORM & ACTUATION

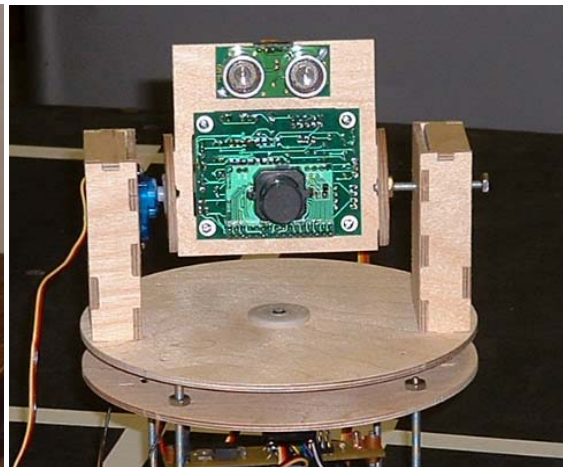
Mechanically, Beamos is a modified “TJ” style robot, made from interlocking balsa wood pieces. Mobility is achieved through two servo powered wheels, while three caster wheels are placed on the bottom to provide stability. Figure 2.a shows the completed base of the robot. The servos are hacked Hitec HS-422 standard servos

The main addition to the TJ structure is a rotating camera mount, suspended above the base using threaded rods. This “head” contains the sonar and camera sensors and is responsible for keeping track of the target object. Two un-hacked servos, a Hitec HS-422 standard servo and a Hitec HS-55 Feather servo, allow for 180° of pan and tilt respectively. This allows the sensors to pinpoint the destination within the maze. Figure 2.b shows the completed head unit.

Figure 2 – Platform



(a) Mobile Base



(b) Sensor Mount Head

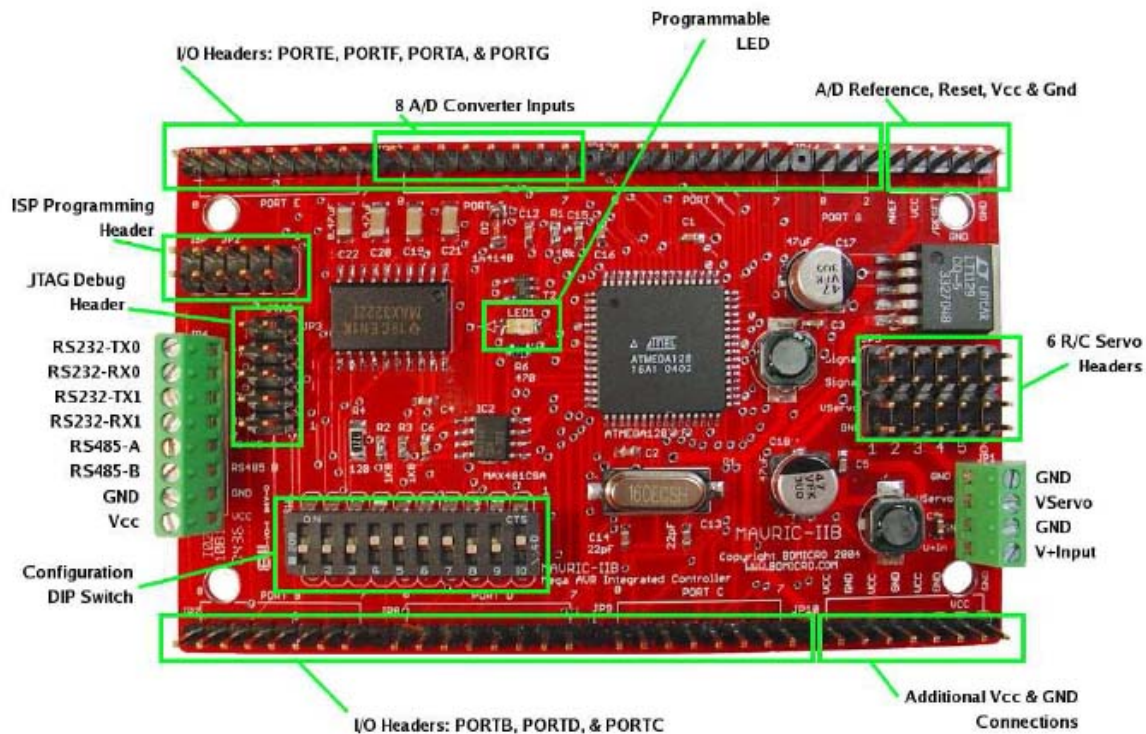
All actuation and sensor control is accomplished through the Mavrik-IIB microprocessor board, which is based off the ATmega128 microcontroller unit. This board allows for serial communication, pulse width modulation, analog to digital

conversion, internal timers, and interrupts, which all necessary components for this robot.

Figure 3 shows the Mavrik board, with essential components labeled (from

www.bdmicro.com).

Figure 3 – Mavrik-IIB Microprocessor Board



SENSORS

OPB745 – Diode Emitter & Phototransistor Pairs

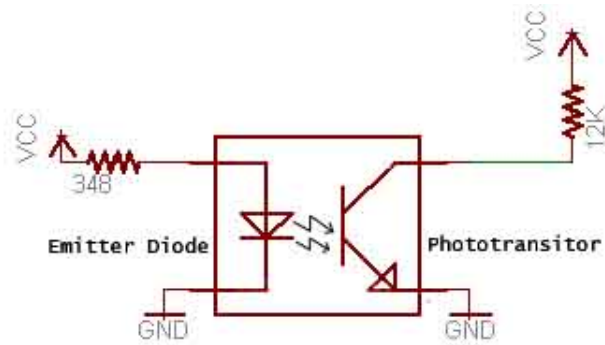
The OPB745 is a small diode emitter and phototransistor pair. When the diode is properly biased it emits light which reflects off a nearby surface and is then received by the phototransistor. The voltage across the transistor varies according to how much light is reflected back. By measuring that voltage with the microcontrollers Analog to Digital Converter, Beamos will be able to determine how much light is reflected back. Figure 4.a

shows the OPB745, while Figure 4.b shows the proper circuitry for operating the OPB745 (values measured in Ohms).

Figure 4 – OPB745 Sensor



(a) OPB745 Package



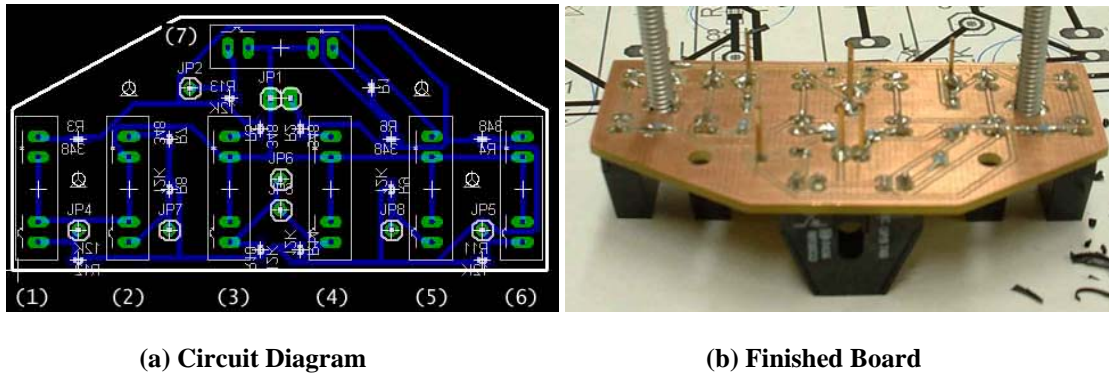
(b) OPB745 Circuitry

However, the amount of light that the sensor picks up depends on a variety of factors including lighting, distance to surface, surface color, surface texture, and surface luster. Through experimentation, the author found that surface luster affected the reflected light more than surface color or lighting conditions. Black and white glossy, smooth surfaces reflected nearly the same amount of light, independent of lighting conditions. Therefore it was necessary to construct the robot's environment (maze) using a dull black surface with a glossy white tape in order to produce the high contrast lines necessary for line following. Additionally, the sensors were mounted between 0.12 and 0.17 inches from the surface in order to properly reflect the emitter diode's light. Distances above or below that range could not reflect any light to the phototransistor.

Beamos required a series of seven emitter/detector pairs to properly traverse through the maze. Figure 5.a shows the circuit board diagram for the seven sensors and Figure 5.b shows the completed printed circuit board. Sensors (3) and (4) are used primarily for accurate line following. Sensors (1), (6) and (7) are used for recognizing the

multiple paths at intersections, while sensors (2) and (5) are used primarily to determine when the robot has entered into a new room.

Figure 5 – Line Following PCB



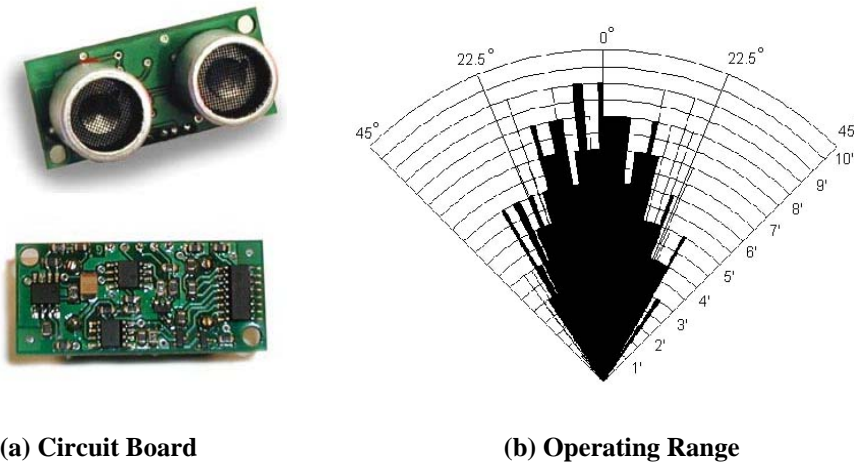
SRF04 – Devantech Sonar

The SRF04 is a sonar module used to determine the distance from the unit to an object. The sonar sends out a short sound pulse and waits for the sound to ping back. In addition to the power and ground lines, this requires a line for sending out the pulse and one to receive it. The microcontroller is set up to record the time between sending and receiving the signal, and the relative distance to the object can be computed from this time. Figure 6.a shows the SRF04 sonar module.

The sonar is capable of detecting very small and thin objects sufficiently within the five feet area of the maze. One possible scheme for using the sonar was to scan the area with the sonar and record the angle at which the shortest distance occurred. However, the transmitted pulse is sent radially away from the unit over an angular range. (See Figure 6.b). Therefore the sonar does not need to be pointed directly at the object in order to detect its presence. To account for this, it is possible to record the angle at which the object enters and leaves sonar's range, and average the values to determine the angle.

However, this reduces the range that the sonar can sweep across (the angle an object at the range's end cannot be determined precisely) and the overall angle calculations are not always accurate.

Figure 6 – Sonar Module



(a) Circuit Board

(b) Operating Range

While the sonar was fully interfaced during the course of this project, the author was unable to properly use its information to produce any beneficial behaviors.

AVRcam – Color Camera Tracking System

The AVRcam (Figure 7) is a compact camera tracking system based on the ATmega8 microcontroller unit. The camera uses serial communication to relay information, such as the current image or object tracking, to an external processor. This information could then be used to track the object as it moves within the camera's field of view, and adjust the position to keep the object centered within the camera's image. The AVRcam system was not fully implemented within the scope of this project, but has been left for future work.

Figure 7 – AVRcam System

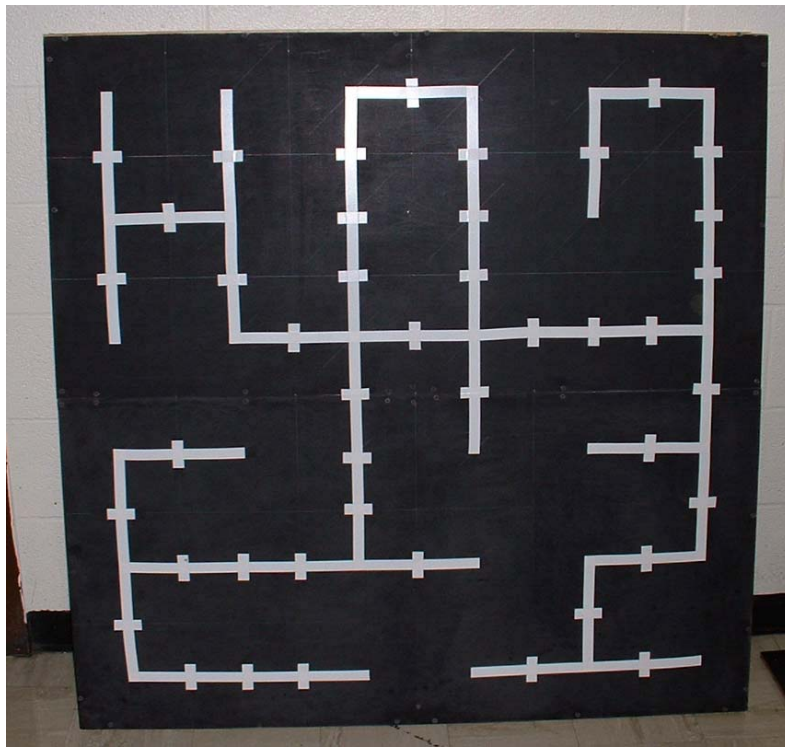


BEHAVIORS

Beamos' fundamental purpose is to traverse freely throughout a maze environment with a variable end point.

Beamos is able to traverse freely throughout the maze and maintain knowledge of his position and orientation using only the OPB745 sensors and internal memory. To do this, all seven sensors are read and recorded. They are then compared with the previous state that the robot was in, to determine its current state. These states determine how Beamos will adjust his motors to accomplish line following. Additionally, they keep track of what type of line Beamos is looking for, either an intersection or a doorway. When a doorway is encountered, Beamos records his new position within the maze and begins looking for an intersection. When an intersection is encountered Beamos determines the path to follow, makes the correct turn, and begins looking for a doorway. While these states sufficiently and reliably allow him to move around, they require an initial knowledge of what to look for first and the initial position and orientation. Figure 8 Shows the maze environment with intersections and doorways.

Figure 8 – Maze



To accomplish maze solving, Beamos uses an implementation of the A* Algorithm that takes into account changing information about the maze. Initially, Beamos assumes that all intersections connect with all other intersections, with the exception of border squares (that do not connect to squares beyond the maze) and the initial square (which is predetermined). When Beamos encounters the next new intersection, he records which paths can and cannot be taken to the adjacent squares. He will then recompute the path from the new current square to the finish square, and then follow the correct path towards that solution. If Beamos enters in a square that he has previously been to, then no new information has been learned about the maze. Therefore the solution to the target will not be recomputed.

While the A* Star algorithm allows Beamos to accurately find the optimal path to the target on the first try (assuming all environmental information is accurate), it requires knowledge of both the starting and ending squares within the maze. Further work must be done to allow Beamos to account for a variable destination using the camera and the sonar to determine the object's angle and distance respectively.

CONCLUSION

Beamos successfully combines several subsystems to traverse throughout a maze while learning and recording information about its environment. It is able to efficiently keep track of his position and orientation, but currently requires internal knowledge of its destination.