Multi-Node Mobile Robotics

New Mexico Institute of Technology
EE382: Intro to Design
Group 2

Prepared For:
Dr. Kevin Wedeward
Dr. Aly El Osery

Written By:
Lee C. Baker
Patrick C. Bastien
Jamison J. London
Phillip P. Lopez
Eric J. Makarewicz
Michael T. Pace
Andrew D. Targhetta
Antonio I. Velasco S.

May 8th, 2006

Abstract

This paper passes on the knowledge that we acquired during our journey in EE 382. The class goal was to create a set of mobile nodes capable of finding a black box. We discuss our methods of localization, searching, design, and implementation. Each topic is covered thoroughly, from the choice of infrared sensors, to the interface for all the node components with the MicaZ. The reader can use this document as a basis for reproduction or modification of the final design. This paper serves as both a conduit for the information, tips, and design tricks that we picked up as well as an owner's manual for the mobile nodes which we created.

Table of Contents

Hardware

Software

Appendix A

Appendix B

List of Figures

Introduction

Our Junior Design group at New Mexico Institute of Mining and Technology has been designing, constructing, and testing a network of mobile robotic “nodes” over the last three months. The main goal of our project is to create a group (two minimum) of mobile nodes which can locate a black box, which measures 30cm x 30cm x30cm, within a playing field bound by 50cm tall white walls which measures 3m x 3m square. The terrain of the playing field is flat with no obstructions except for the black box.

The nodes must begin the operation of finding the black box by first localizing themselves. For our project, localization is defined as: the nodes knowing their position within the playing field relative to the objects around them. The process of localization must be independent upon the number of nodes. By making this requirement, the localization scheme can be very robust and versatile; we will be able to add and remove nodes as needed.

Once the nodes have been localized, the nodes are to begin searching the playing field for the black box. The nodes need to be able to avoid walls and each other while they search. Once a node locates the box, the location is to be broadcast and the remaining node(s) are to converge upon the box. In doing this, we are able to verify the black box’s location and existence.

Localization

Localization is the part of the project that maps where all the robots are in reference to each other and the arena. The localization scheme as a whole includes the hardware on the base station, the hardware on each individual node, the software on the base station computer, and the software on each individual node. The hardware concerning our project's localization scheme will be the main topic in this section.

Each team was allowed to decide on their localization method. Our group chose to use a hybrid type of localization, which includes the use of lasers and the existing RF communication capabilities built into the MicaZs. More specifically, we decided to use an aerial style laser beacon for scanning both the x and y coordinates of our mobile nodes. To scan either of the coordinates, we needed a laser module that can project its beam downward in a linear fashion from one end of the 3 meter arena to the other, parallel with the opposite axis. We also needed to find a method, which we could use to sweep this line laser beam from one end of the arena to the next, moving along the primary axis. We realized that if we had an instrument that measured the angle at which our beam is directed downward in reference to the z axis (the axis normal to our arena) and a sensor specifically made to detect our beam mounted on each mobile node, we would be able to determine the robots location along the x axis. This method of course assumes that there is a reliable and nearly instantaneous communication link between mobile nodes and the aerial beacon. The concepts concerning the x axis scanning just described above could then be applied to the y axis scanning, giving us a fairly reliable 2 dimensional localization scheme. In addition to the 2 dimensional scanning, we realized that if we added a second detector on each of the mobile nodes, we could actually calculate the direction each mobile node is facing. After testing this theory, we realized that this information could be an extremely useful aid for the act of searching for the black box. Below are some figures that demonstrate the basic concepts behind the localization scheme our group has engineered. The last picture shows the relationship between the base station MicaZ, the aerial beacon, and the base station desktop computer. The desktop computer was where we determined all of the localization calculations would take place. In addition, a GUI would run on the desktop mapping out the location of each node. The text that follows these figures will explain how we attempted to achieve these basic concepts.

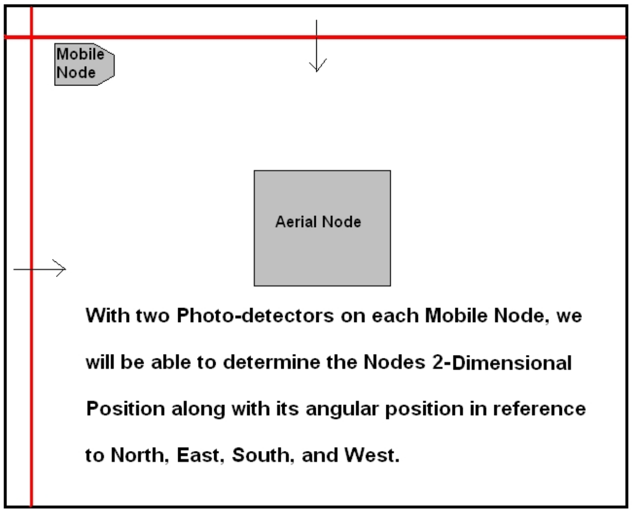


Figure 1: Here is the view above aerial beacon showing the beacon, a mobile node and both the x and y laser beams in the arena at the same time (for demonstration purposes only).

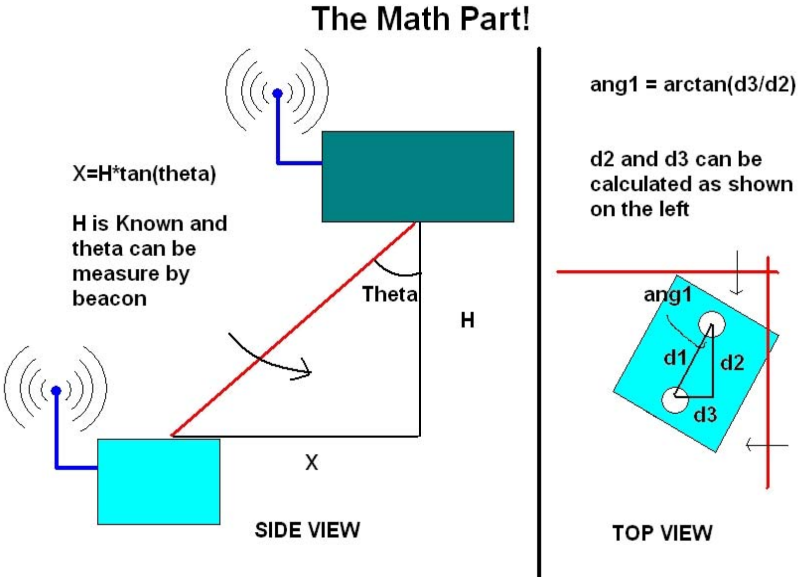


Figure 2: Here is the mathematics behind our localization scheme, which proves that this scheme is achievable.

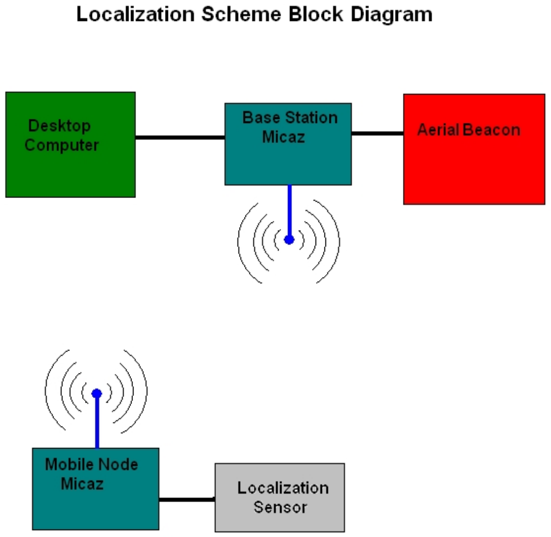


Figure 3: Shown above is a simplified block diagram showing how the major components in our proposed localization scheme work together.

The first hurdle we had to overcome for our localization method to work was to find a laser module setup that would project a linear shaped beam on the arena. If we could not find a method to do this, our scheme would prove to be impossible to implement. Our localization team searched the internet and found some line laser modules that had a 90 degree fan angle and a 3mW power rating. We felt the power would be adequate and were quite happy about the fan angle. With a 90 degree fan angle, out laser modules would have to be 1.5 meters above the arena in order to cover the entire 3 meter area. The next problems we had to overcome involved how to sweep the laser beam and how to detect it.

To solve the first of the two problems, we searched for a gearbox that would have two output shafts perpendicular to each other. We wanted the output shafts to be geared at the same ratio relative to the motor, and we wanted this ratio to be high (in other words we wanted the output shafts to be low geared). This gearbox also had to have an encoder so we could calculate the angle at which the output shafts are positioned. Once we found the gear box we needed, we could mount mirrors to the shafts and direct the laser modules towards the mirrors. As the mirrors rotate, our laser beam would sweep the arena. See figure 4 below.

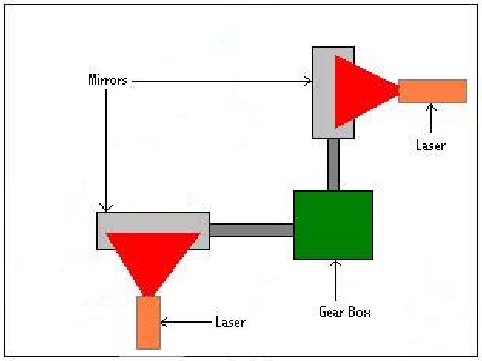


Figure 4: Aerial beacon representation.

Unfortunately, we could not find an off the shelf gearbox that would perform the way we wanted it to so we looked to fabricating it ourselves. The final solution was to modify one of the existing gearboxes given to us at the beginning of the project. With the aid of angle aluminum, Plexiglas, a few erectors set parts, and many man hours, we were able to develop a gear box which met the specifications we stated previously. The gear box had an encoder shaft, which allowed us to mount a Vex optical encoder we bought from Radio Shack. The encoder to output shaft ratio was 12.25:1, while the motor to output shaft ratio was on the order of 300:1. This meant we could rotate our output shaft slowly, while rotating our encoder at a fairly high speed in order to maintain accuracy. The optical encoder was rated at 90 pulses per rotation and could handle up to 18 rotations per second. However, testing in lab verified that we could safely exceed this rating with little waveform degradation. Assuming an ideal communication scheme between the base station and mobile nodes, we calculated an overall system accuracy of 3cm with this particular encoder. The group agreed this was up to par. In the first model, we designed the system with an output angular velocity of 1 rotation per second in mind. This would allow us to safely stay under our encoder manufacturer's maximum rating, and yet give us a system refresh rate (assuming double sided mirrors) of 2 times a second, which was four times the requirement. In finishing our gearbox, we spent a lot of time cutting and gluing the mirrors, which would mount on the shafts of our gearbox. We had to cut out 0.75x1.5 inch mirror rectangles and glue them to machined plastic pieces that would fit on the hexagonal output shafts of our gearbox. Our team felt that a double sided mirror configuration would work best and realized the importance of a straight gluing job! If glued crooked, the mirrors would cause the line laser beam to wobble as it is swept across the playing field. Shown below is a picture of our finished gearbox. Note the laser modules, photo-transistors, and mounting brackets. The photo-transistors allow the MicaZ to have an angular position it can reference to.

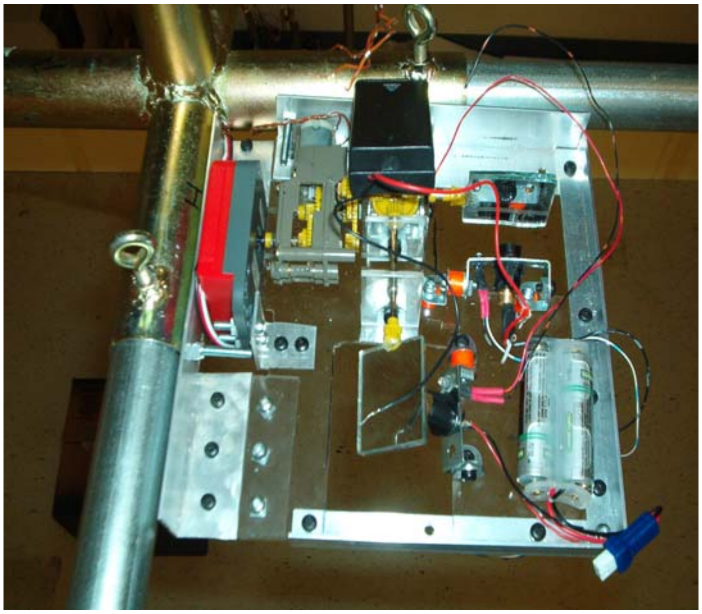


Figure 5: Aerial beacon final hardware.

The next component to be built for our localization is the board that controls the aerial beacon. The block diagram, shown below, displays how this control board fits into the overall scheme.

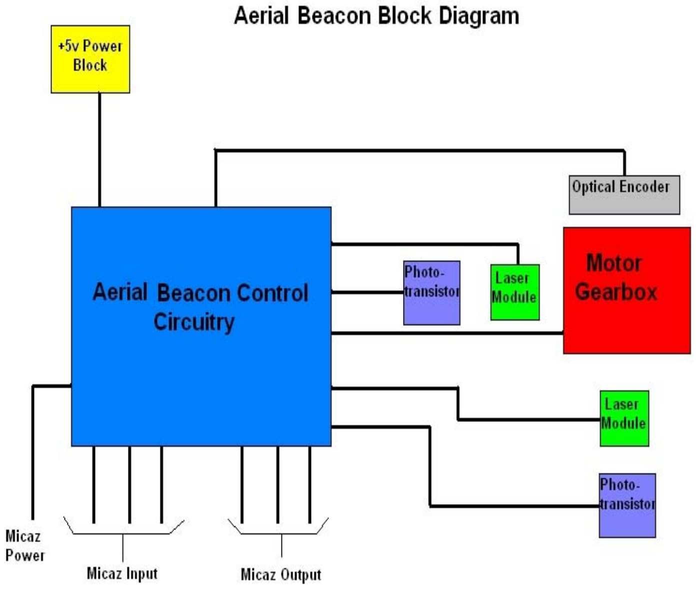


Figure 6: Aerial beacon block diagram.

The aerial beacon control circuit provides power to the gearbox motor, encoder, and the laser modules. It allows these devices to be turned on or off with the MicaZ's 3v digital outputs. Also it allows the triggering of the MicaZ interrupts when the angular reference photo-transistors detect the laser beam. Adjustment from the two potentiometers allow for reliable operation that can be tuned to the lighting environment of the gearbox. Shown below is a picture of this circuit. The board was designed in the freeware board design software, EAGLE, and its schematic and PCB drawing can be found in Appendix B.

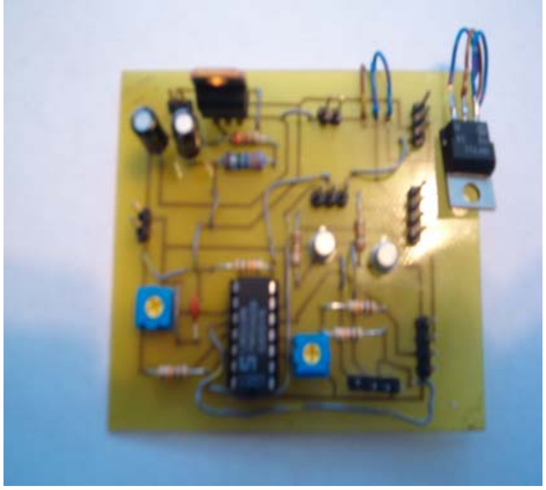


Figure 7: Aerial beacon control board.

The next item we had to address was the sensor that would be mounted on the mobile nodes and be used for the detection of our laser beam. The requirements for this sensor were that it would be able to pick up on a weak laser beam at a wide angle in an environment in which the ambient light could fluctuate. To do this, we designed a circuit with two photo-transistors and a comparator. The voltage reference being fed into the comparator was both electronically and manually adjusted by a combination of a manual and a digital potentiometer. Output and input pins along with power and ground allow the MicaZ to adjust the reference voltage according to the ambient light and receive triggers according to which photo-transistor detects the laser beam. Shown below is the block diagram of our localization sensor. Schematics and PCB figures are shown in Appendix B.

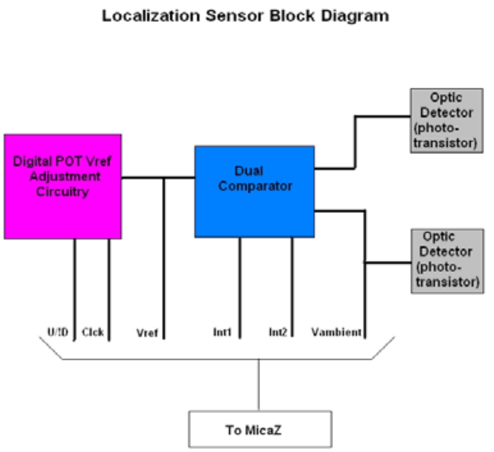


Figure 8: Localization sensor block diagram.

Shown below is one of our finished boards. Two more were made in addition to this one.

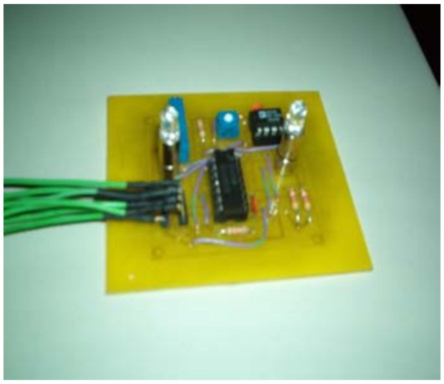


Figure 9: Localization board.

The final component in the hardware part of our localization is the design and fabrication of the part that will hold our localization over the arena. We needed something to elevate our aerial beacon in the air and not block the view of the arena below. We considered hanging the beacon from the ceiling but we were concerned with the beacon oscillating or swaying in the air from the motors and the rotating mirrors. So we needed to come up with a platform extending from the ground for stability. That's when we decided to use parts from an old tent to build such a platform, which included four elbow brackets, eight one inch conduit poles, one T-connector, and four tent feet. The aerial beacon must be mounted a little off center from the middle of the arena in order for both lasers to be located at the center of their respective coordinates. In order to due so we had to cut two of the four poles that extend over the arena shorter then the other two.

Shown is the arrangement of the platform over the arena.

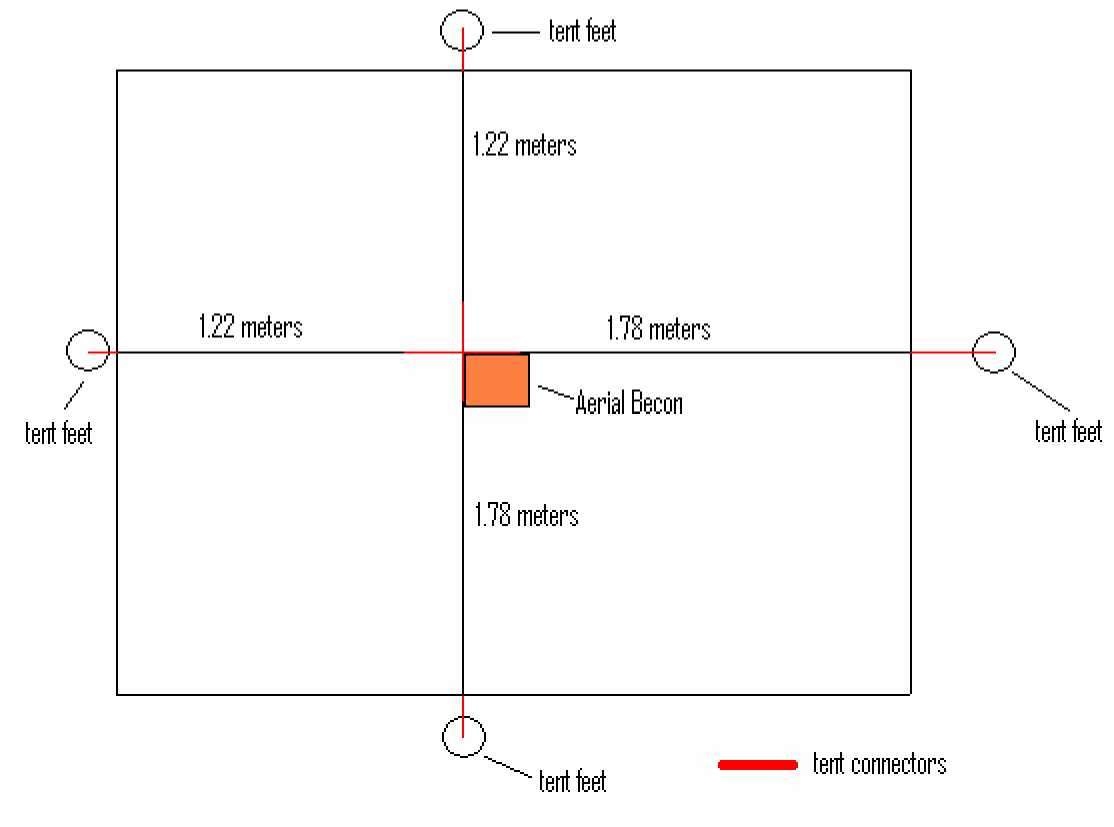


Figure 10: Top view of platform.

Before we cut the legs for the platform, because the poles were 3.05 meters tall, we hand held the aerial beacon over the lab room floor to make sure the laser swept a three meter area. This is when we found out that the beacon had to be raised higher then a meter and a half. We found that the shortest height that the aerial beacon had to be was 1.78 meters, which we cut the length of the poles at.

The localization hardware proved to work fairly reliably in lab. The control circuitry was successful in controlling the components on the aerial beacon. The photo-transistors on the aerial beacon were proved to trigger the MicaZ's interrupts successfully. The aerial beacon was shown to actually sweep both the x and y coordinate laser beams without wobble. The localization sensor was proved to initialize first with inputs from the function generator in lab, then with the MicaZ control. The localization sensors also appeared to be triggered by the laser modules regardless of angle and distance. However, the laser was proved to be incorrect once an attempt to integrate all the systems together was made. After we hung our aerial beacon up in an attempt to test our sensor boards in the field, using the actual beacon, we found that our sensors lost sight of the beam when placed more than 1.5 meters outside of the center of the arena. We determined this was primarily due to the angle at which the laser beam was coming into the photo-transistors. Even with our auto-adjustment circuitry in full operation, we lost sight of the beam. In order to overcome this problem we considered three options. The first option we considered would involve raising the aerial beacon higher so the lasers would be insight of our photo-transistor.

Shown below is how a raised beacon would let the lasers be insight of our sensor.

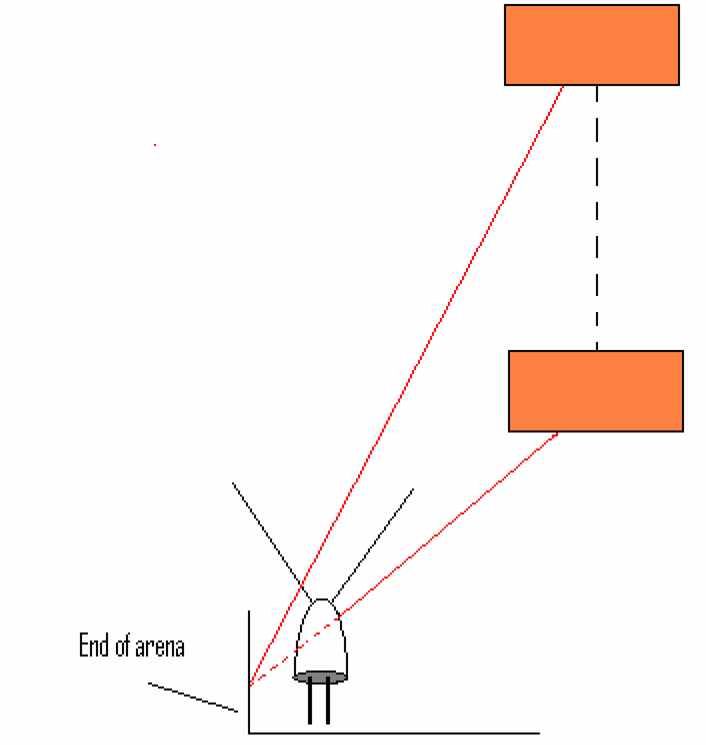


Figure 11: A quick fix to sensor problem

With this option our accuracy of our aerial beacon would decrease because we would decrease the angle which the beacon sweeps the arena with. So we would have to split the three meter area with in a smaller angle, thus a bigger accuracy error. The second option we considered involved adding multiple photo-transistors arranged in a flower bouquet pattern around our original photo-transistor. The outer photo-transistors will be angled slightly outward so the photo-transistors viewing angle overlap and that the lasers are always insight of at least one of the photo-transistors. The problem with this option is that it would involve buying more photo-transistors when we were already in a tight budget. Plus this is for only one node, we would have to add more photo-transistors to each node we plan to use.

Shown is how the extra sensors would be arranged so they overlap each other.

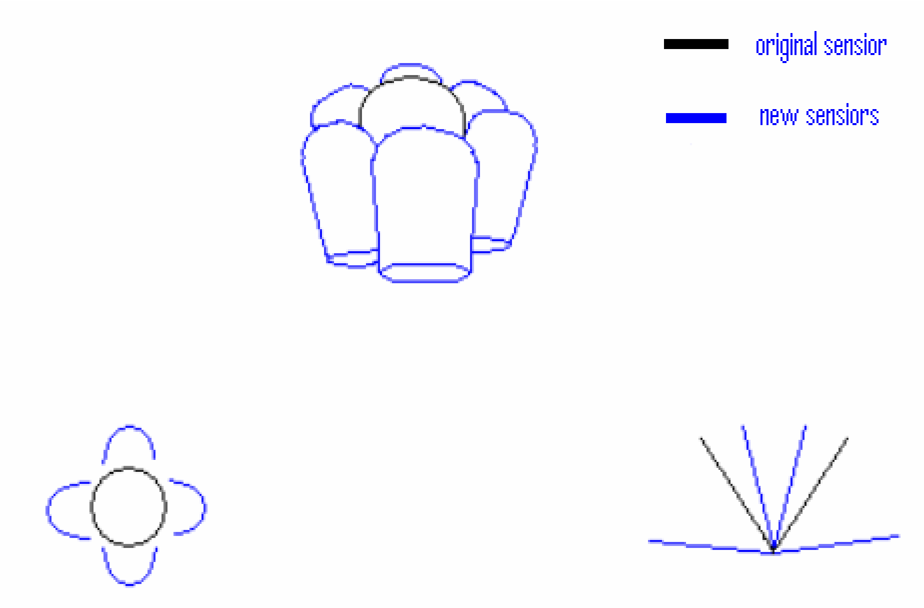


Figure 12: Second possible fix to sensor problem.

The third option involved a new location and design of our original photo-transistor. We would have to build a metallic dish so the laser would reflect off the dish and into our photo-transistor, kind of like a satellite dish.

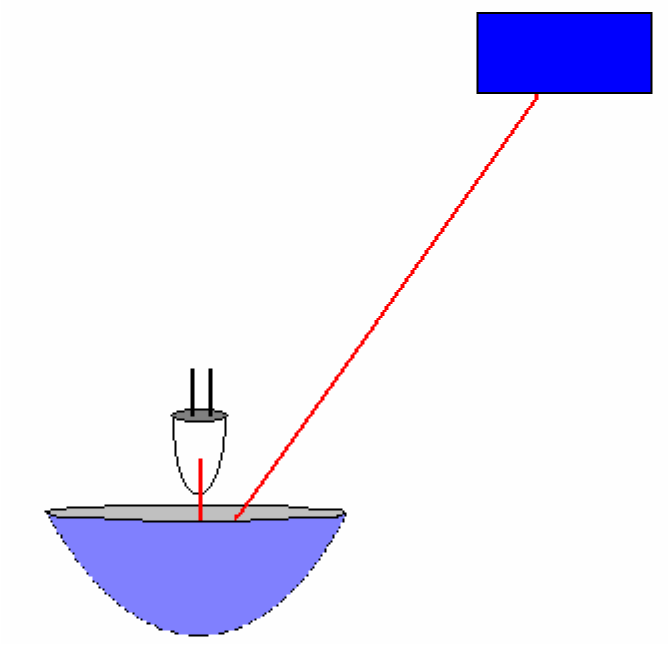


Figure 13: Third possible fix to sensor problem.

This third option would involve building a dish for each node that we would use. Out of the three options the first option seems to be the most reasonable, because it would only involve longer platform legs. The other two would require a lot of research in optics and testing in the lab to align correctly.

As we stated earlier, our localization hardware worked only within 1.5 meters of the center on the arena. However, this was not the only short comings we had with our localization scheme. First, we had a laser module fail, and we are not quite sure what caused the malfunction. We were able to replace it with a module from a laser level purchased from the local hardware store. This module had a 90 degree fan angle and a 5mW laser. Regardless, the optical part of the module that fans out the laser beam proved to be of lower quality than our original laser module. The center of the line was far more intense than the outer parts and voids in the line could be found towards the ends. The second problem we had with our scheme had to do with the communication side of the project. More on this will be discussed later on.

Sensor selection

Sensors, which are the means by which a robot interacts with its environment and surroundings, give nodes the majority of their functionality and versatility. When we initially began planning the sensor platform on the mobile nodes, we had several objectives which we hoped to accomplish. First, we wanted to be able to receive three distance measurements continuously in real time. Second, we wanted to be able to detect the black box once a mobile node came upon it. Finally, we also wanted to remain within the budget of the project. With a playing field of 3 meters x 3 meters, we knew that our sensors should have a range of magnitude of at least 75 cm. This would give our readings a greater deal of resolution for us to work with.

With our parameters defined for us, we chose the Sharp GPD12 Infrared Sensors. These sensors were the perfect balance of cost, range, and versatility. The GPD 12 Infrared Sensors output an analog voltage which is proportional to the distance between the sensor and the closest object directly ahead of the sensor (See Appendix C). These sensors are capable of output distance measurements of 8 - 80 cm. We decided that these sensors should be directed at 90º angles from each other on the front of the mobile node's chassis, having one facing straight ahead, one facing left, and one facing right (See Figure 14). By choosing this configuration, the mobile nodes were able to perform wall following in either direction while still scanning the area ahead for the box.

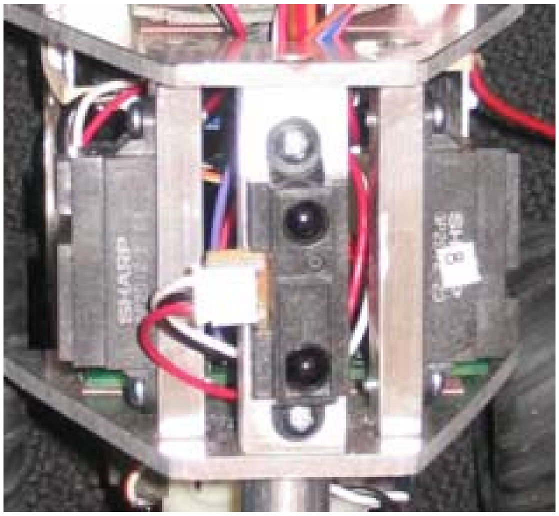


Figure 14: The Sharp GPD12 IR Sensors mounted on the chassis.

With accurate distance measurements continuously being read by the MicaZ, we needed an effective way of detecting the black box. We accomplished this by adding a simple circuit, consisting of two photo-transistors, to the front of the chassis (See Figures 15 & 16). This simple circuit enabled us to determine if an object, which was in front of a mobile node, was black. The transistors allow current to flow under normal ambient lighting, but when faced with a black object the transistors limit the amount of current flowing; thus, causing the output voltage to drop below 1.2 volts. If this drop in voltage is detected by the MicaZ, the MicaZ will in turn assume that the node has discovered the black box. An obvious downfall to this design is in the fact that if the lighting on the playing field is not uniform or fluctuates in time, then false triggers can occur. This could easy be solved by adding infrared emitters to the circuit and filtering the transistors to only be affected by this wavelength of light.

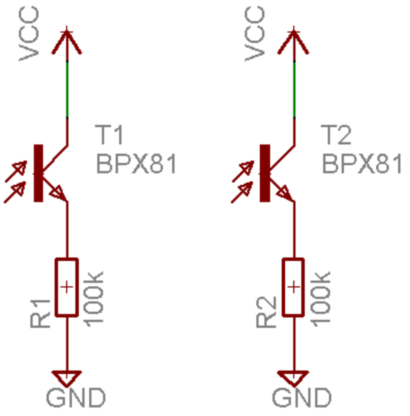


Figure 15: The "Black Box" Detector Schematic

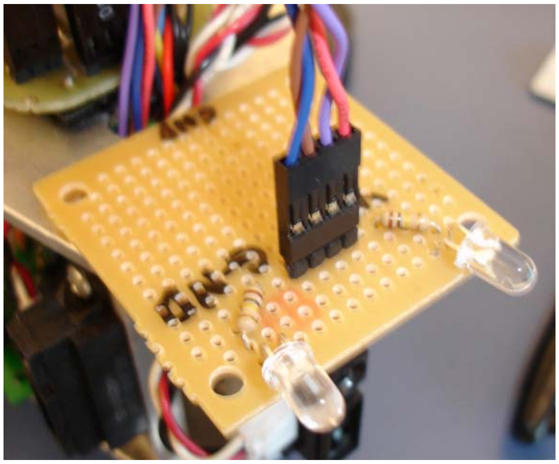


Figure 16: The "Black Box" Detector

We were able to obtain very good results from all of our selected sensors. Wall following and obstacle avoidance was achieved by adjusting wheel movement in order to maintain a minimum output voltage from our GPD12 sensors. Though we hardly had a collision with a foreign objects (walls, other nodes, etc.), we found that the robots had a few blind spots where they could not see a narrow object in front of them. This problem could be solved easily by adding two more sensors to our initial design.

Chassis Design and Construction

From the beginning of the project, we wanted to have a very solid and robust chassis to support our robots. We decided to spend a good portion of our time planning the design of the robot so that manufacturing of this design would be simpler later on in the project. We began this process by laying the desired chassis out in a design suite (See Figure 17).

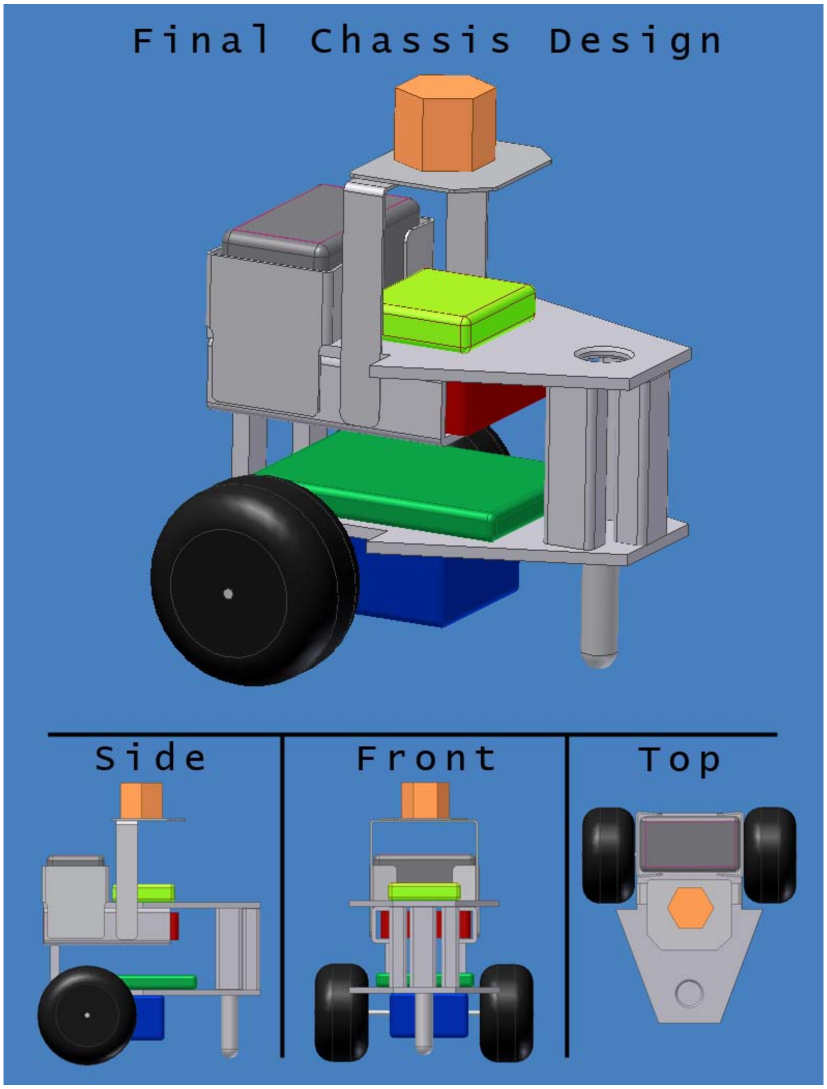


Figure 17: The final chassis design from multiple angles.

Once we completed our design phase, we began to produce a prototype chassis. We choose to use 50-50 aluminum alloy throughout the chassis due to its light weight and ease of machining. We also wanted to eliminate the use of nuts and washers in our design so, we decided on drilling and tapping all of our pieces so that everything fits together in a neat and orderly fashion.

By having a three layer design, we were able to fit all of the electronics needed on our chassis. Our "decks" are laid out in a logical fashion, which helps to organize the whole robot. The overall deck layout can be seen below in Figure 18.

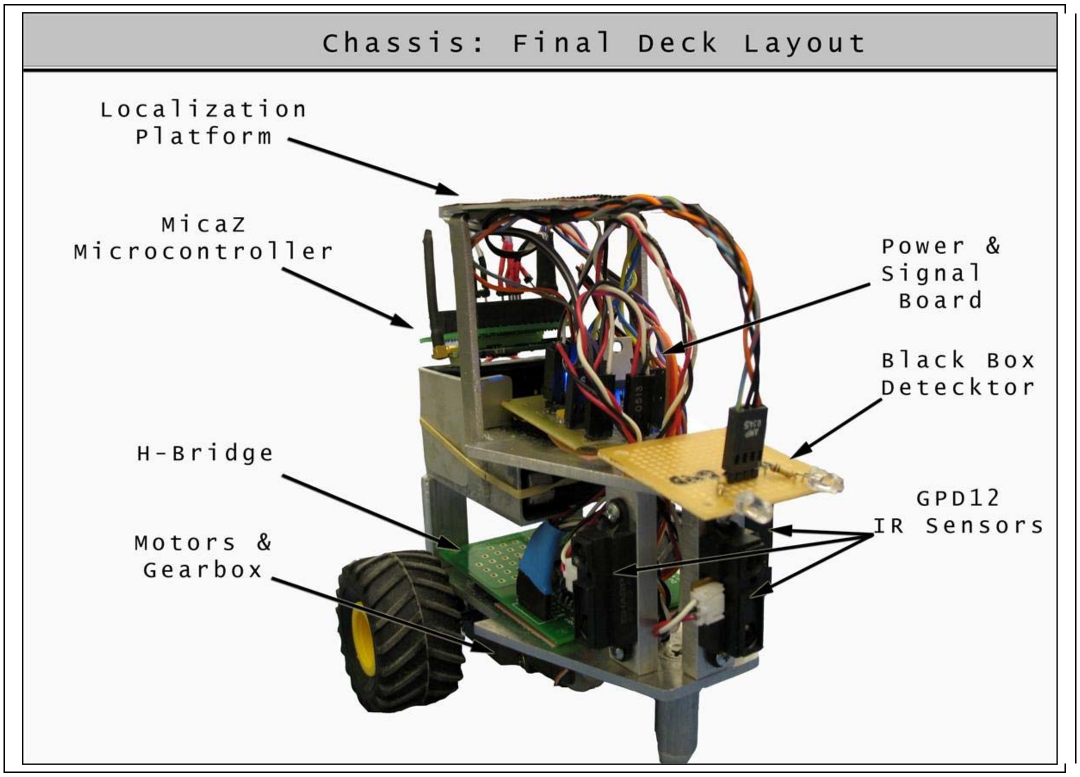


Figure 18: The Final Chassis Deck Layout.

The Upper Deck is used primarily for the localization hardware. On the Middle Deck of the chassis we have the battery compartment, the MicaZ microcontroller, the power and signal distribution board, and the black box detection circuit. The supports between the Middle and Lower Decks mount our three GPD12 sensors and allow for angle adjustment on these sensors. The Lower Deck of our chassis is the home of our H-Bridge (motor control) and the gearbox, motors, and wheels.

Once we had our prototype up and running we decided that our chassis design was suitable enough to go into full production and we made two more identical chassis'. All of our nodes can be seen below in Figure 19.

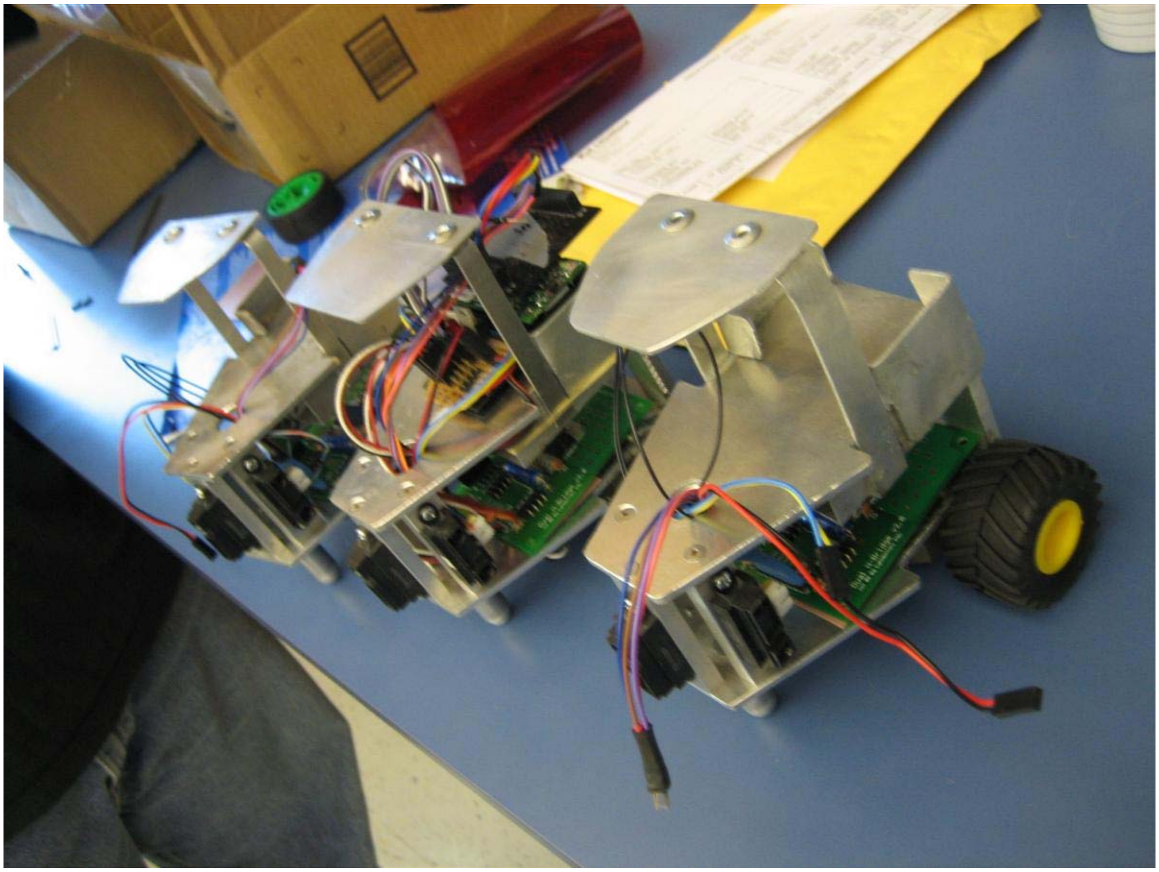


Figure 19: All of the completed chassis.

We had considerable success in our chassis design and production. We had very little problems with these chassis' and found them to be very serviceable. In the end, we were able to have a fast and maneuverable chassis while maintaining a sturdy platform for all of our mobile nodes.

Power Systems

Early in the design process we decided that we would need a separate power supply for our motors and sensors. The MicaZ is able to provide around 3 volts from two standard alkaline batteries; however, we did not want to draw too much current from this source and risk causing the MicaZ to fail or malfunction. For the motors, there is no doubt we needed another power source because their average current draw is on the order of 650mA. We also needed to provide a 5 volt rail for our onboard sensors, the Sharp GPD12s.

Due to all of these concerns, we decided to use 2500 mAh NiMH AA 5-cell battery packs. We decided to run the motors off the unregulated 6 volts to maximize power and to keep our regulator dedicated to our electronics. This would help keep motor noise off of our sensor line. We also needed to interface the H-bridge with the 3 volt logic levels from the MicaZ (3 Volts). With all of our electronics coming together on our chassis, we needed a solid method of dealing with our signal and power buses.

To handle our concerns, we designed a voltage regulator and signal distribution board using EAGLE. Our board was used as an interconnection point between our 3 volt logic levels and our 5 volt regulated power lines and our 6 volt motor power lines (See Figure 20).

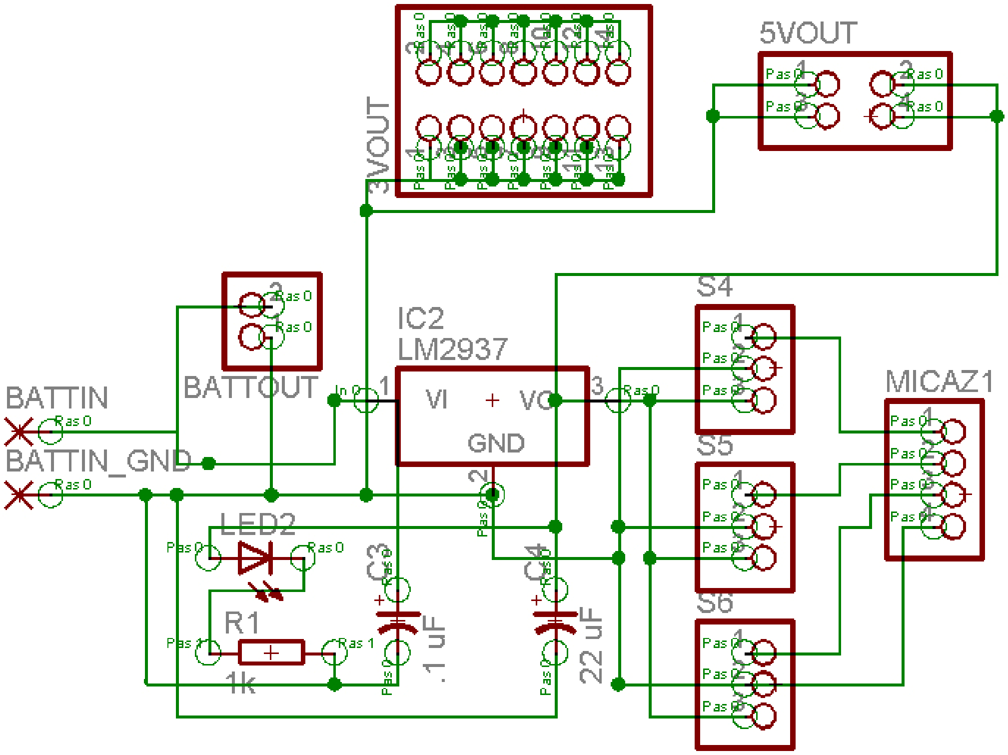


Figure 20: The power board schematic.

Once we had laid out the board and double checked it, we went ahead and etched it in the EE Tech Lab. The basic board layout can be seen below in Figure 21.

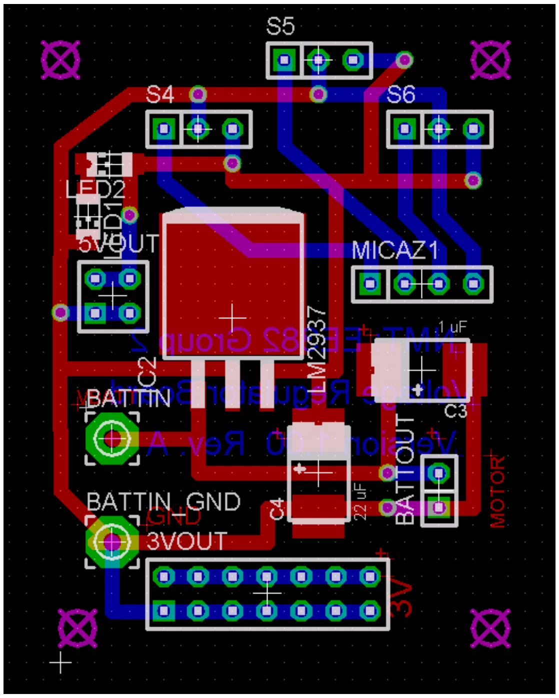


Figure 21: The final power board layout.

Once we had etched our dual-layer boards, we drilled them, populated them, and made sure that they functioned appropriately. The final board can be seen in a working robot below in Figure 22.

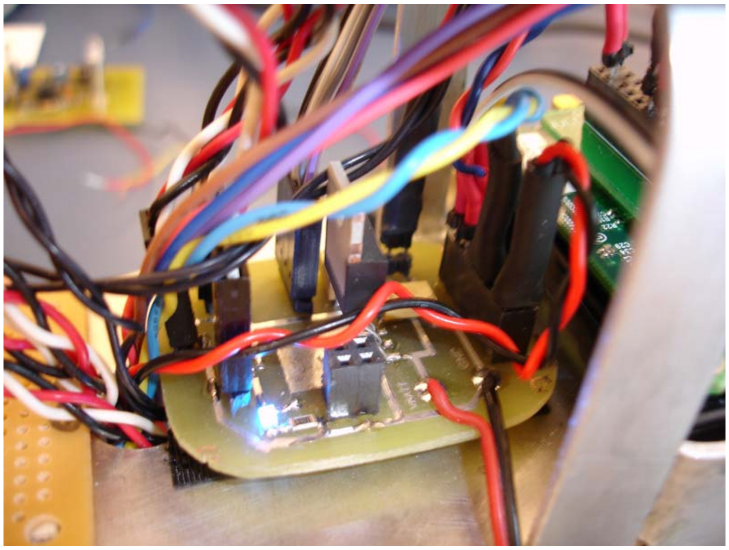


Figure 22: The final power board installed in one of our nodes.

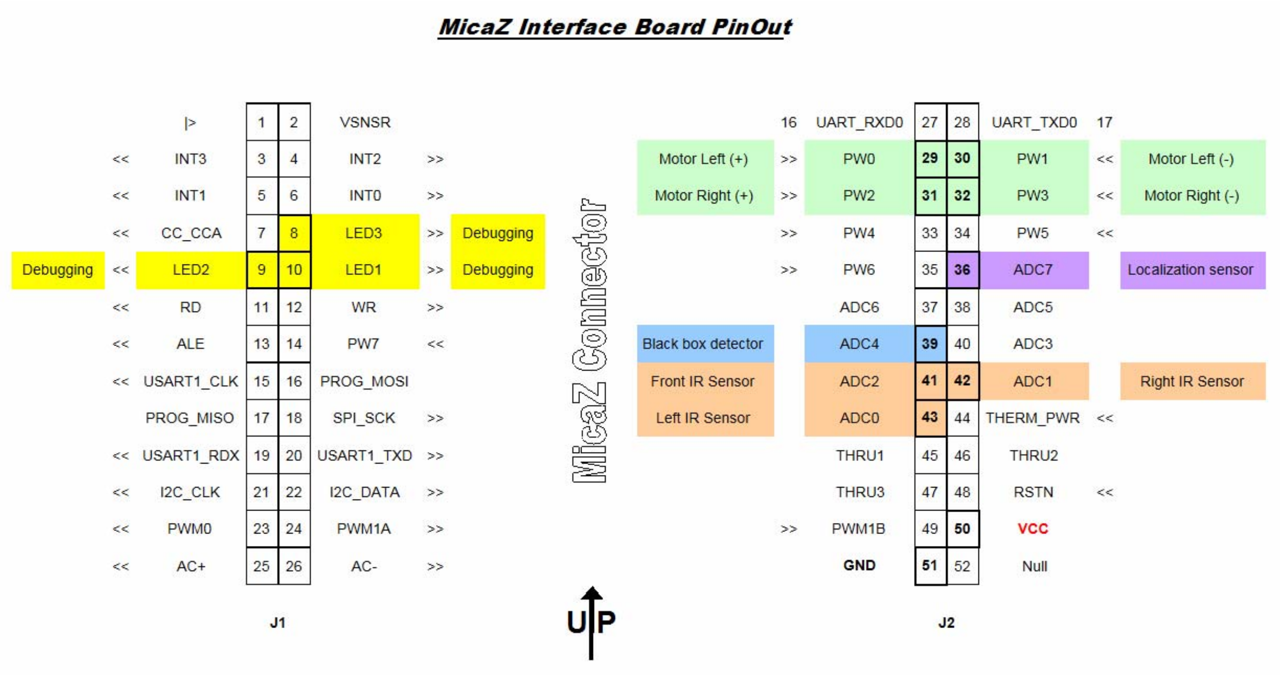
With our power boards built and installed we were able to achieve very impressive operation times. Our overall power budget can be seen in Appendix A; however, it is safe to say that we were able to operate the robots for up to 90 minutes at a time without seeing a significant effect on the charge of the NiMH battery packs.

By integrating all of our signals onto one compact board, we were able to simplify our design and create a more aesthetic look for our robots. We were also able to distribute power through out the robot in an orderly fashion, which helped to establish reliable operations.

Software

MicaZ Pin-Out

Before going into the programming, we should define what we used in order to get everything attached to the MicaZ and under control. Also, we needed a useful schematic which will help us to have a very good idea of what we should plug-in in each port. So, the final schematic can be seen below in Figure 23.



MicaZ Interface Board PinOut

		|>	1	2	VSNSR		
	<<	INT3	3	4	INT2	>>	
	<<	INT1	5	6	INT0	>>	
	<<	CC_CCA	7	8	LED3	>>	Debugging
Debugging	<<	LED2	9	10	LED1	>>	Debugging
	<<	RD	11	12	WR	>>	
	<<	ALE	13	14	PW7	<<	
	<<	USART1_CLK	15	16	PROG_MOSI		
		PROG_MISO	17	18	SPI_SCK	>>	
	<<	USART1_RDX	19	20	USART1_TXD	>>	
	<<	I2C_CLK	21	22	I2C_DATA	>>	
	<<	PWM0	23	24	PWM1A	>>	
	<<	AC+	25	26	AC-	>>	

J1

MicaZ Connector

UP

	16	UART_RXD0	27	28	UART_TXD0	17	
Motor Left (+)	>>	PW0	29	30	PW1	<<	Motor Left (-)
Motor Right (+)	>>	PW2	31	32	PW3	<<	Motor Right (-)
	>>	PW4	33	34	PW5	<<	
	>>	PW6	35	36	ADC7		Localization sensor
		ADC6	37	38	ADC5		
Black box detector		ADC4	39	40	ADC3		
Front IR Sensor		ADC2	41	42	ADC1		Right IR Sensor
Left IR Sensor		ADC0	43	44	THERM_PWR	<<	
		THRU1	45	46	THRU2		
		THRU3	47	48	RSTN	<<	
	>>	PWM1B	49	50	VCC		
		GND	51	52	Null		

J2

Figure 23: The final schematic for our MicaZ interface board.

Then, we needed to determine which port was going to be used for the motor control, for the obstacle detection, and for the searching algorithm (the infrared sensors), for the localization sensor and finally for the box detection (color detection). Therefore we used, as can be seen in Figure 23, these pins:

- PW0 to PW3 for controlling the rotation in the wheels and drive the node.
- ADC7 for get the signal of the localization board that was picking up our lasers in the aerial beacon.
- ADC4 for the two photo transistors that were set to detect the black box in the white arena.
- ADC0 to ADC2 for the infrared sensors that were used for our search algorithm and wall detection.
- And, we used the LED1 to LED3 for debugging our code and hardware to make sure that everything was working properly.

Code Implementation

After making the pin assignment, we started to program our code and this is how we implemented the search algorithm and wall detection:

Our selection of search method leaves much room for improvement, but suffices with the lack of integration. As it stands the search method is a random, non-coordinated search, where the nodes attempt to right wall follow at a distance of 6 inches.

The right wall following is accomplished by commanding the robot to drive forward until the front IR sensor detects an object at 6 inches distant. The robot then pivots to the left and continues forward. The wall is supposed to be kept within 6 inches of the right sensor by having the robot turn to compensate as it continues forward.

Upon reaching a corner, where both the front and the right sensors are registering 6 inches, the robot is to make a 90 degree turn and continue straight. These steps will continue until all 3 sensors register an object, meaning that something is close to the robot that is not a wall. It would then turn towards the disturbance in the force and verify that the object is the box by running a lap around it.

In reality the robot acts very little like it is programmed, as there are no encoders on the wheels, causing all turns to vary in angle from 90 to 180 degrees. This means that the robot rarely achieves wall following, instead ranging around the arena at random. The box locating subsection can not operate as well with these conditions, and in order to make up for this a separate sensor was designed to trigger when a black object is found. This circuit is discussed in more detail in the next section.

TinyOS

We were initially provided with several TinyOS 1.1 development installations. After several weeks of development using TinyOS 1.1, we became extremely frustrated with the lack of documentation of the API, as well as the inconsistent interfaces. A search for alternative solutions revealed that TinyOS 2.0 beta had just been released, and had fairly thorough documentation already written. Example code using the 2.0 API was rather sparse, but we felt confident enough to try to use the new API for a three to four week trial period. During this trial period, we accomplished many things with the new OS that we had previously accomplished with TOS 1.1. However, TOS 2 had several new and different quirks that slowed development, and eventually demonstrated the immaturity and bugginess of this early version. The first unfunctional module that we discovered was the example basestation module, in which communications was nonfunctional in one direction (from the computer to the basestation via the serial link). This was eventually solved by rewriting the serial communication protocol in both directions on both the TinyOS side as well as the computer interface application side. The rewritten communication protocol had many limitations not present in the (theoretically working) original version, such as a limited packet size of 4 bytes, and no error detection or correction, as well as no addressing. However, this presented no limitations for us, as our design did not require these features.

The second TinyOS 2 system to present us with problems was the analog-to-digital converter subsystem. This system always returned a reading of 0 volts, regardless of configuration. This was also rewritten, using the HAL 1 (Hardware Abstraction Layer 1 – the basic hardware interface layer, interfacing directly with registers) TOS 2 interface. This system presented no problems for the rest of the project, and eventually became the only reliable system on the robot nodes.

During system integration, communication became less and less reliable until it finally ceased to operate. The cause appeared to be a conflict between the communication system and another system integrated in the node module, though we could not figure out which one. All debugging attempts during the limited time we had failed. As time for the project waned, we decided to go with a simpler solution that did not rely on communications as much. We felt that this was definitely feasible in a short amount of time, as the code interfacing with the various IO pins had not failed since the rewrite of the ADC modules.

The final program used to drive the robot (“node2”) is a modified version of a previous program of the same name used to test the ADC sensors once the ADC code was rewritten. Upon receipt of a “go” message from the computer/basestation, each node begins searching for the target object. It proceeds to move randomly about the box until encountering the target object. This is accomplished by simply sampling ADC values for the three distance sensors, as well as the values of the two black/white sensors in order to both avoid objects and to detect the target object. The code compares the voltage read from the distance sensors to preprogrammed values in order to determine the proximity to a wall or other object, and then sets the motor control to respond accordingly. Another set of two ADCs measures the voltage read from the white/black infrared sensors in order to determine if the target object is present in front of the sensors. If the object is present, the motor control is shut off, and an LED is turned on to signal positive detection of the target.

The final code consists of three different modules. The final basestation module, (“finalbase”), consists of code that relays messages from the computer to the node, as well as monitoring the encoder on the localization board in order to keep track of the angle at which the laser is currently aimed. This provides the computer with angles necessary to calculate coordinates for the localization system.

The second module is the final node module, which was not used in the final hardware demonstration due to communication issues mentioned above. This module monitors the radio for instructions directing motor control, as well as monitoring the localization sensors for signals instructing it to communicate with the basestation. This module is also responsible for adjusting the digital potentiometer on the localization board in order to detect the passing of the laser.

The third module ("node2"), actually used in the hardware demonstration, contains code as described above. This is some of the simplest but most robust code in the project.

Conclusion

Throughout the semester we worked on solving the problem of finding a box with mobile nodes in a given area and finally we succeed in it. Everything we set out to do with this project was successful to at least some degree. And, if it was not totally accomplished, we came up innovative ideas and concepts of how to accomplish our task. The main difficulty we had was the total system integration, but eventually we made almost everything work: the search algorithm, the obstacle avoidance, the chassis, the sensors, the aerial beacon and computer communications. The only problem that we could not solve in the allotted time was the two-way communication between the nodes and the base station and the localization board. We had setbacks with the phototransistors because of their angle dependence. They lost sight the laser beam if the robots got further than 1.5 m diameter circle from the center of the arena; however, even without localization we still managed to find the box at the end with our two nodes. So in the end, our goal was close to being achieved successfully.

Appendix A: Budgets

Fiscal Budget

		Item	Cost	Quantity	Total
Andy's Shop Purchases					
	3	Perf board	$1.00	1	$1.00
	1	Pin header	$0.50	1	$0.50
Various Connectors(1)	1	Crimp pin	$0.15	59	$8.85
Small Component Parts(2)	1	Connector housing 3-pin	$0.75	8	$6.00
Circuit Boards and Supplies(3)	1	Connector housing 2-pin	$1.00	4	$4.00
	1	Connector housing 4-pin	$0.75	4	$3.00
	1	Connector housing 6-pin	$0.75	1	$0.75
	2	10k precision pot	$1.00	1	$1.00
	2	Switch	$0.50	1	$0.50
	3	Blue etch paper	$1.50	1	$1.50
	2	240ohm Precision resistor	$0.25	1	$0.25
	2	150mH Inductor	$0.50	1	$0.50
	2	220mH Inductor	$0.50	1	$0.50
	2	Comparator	$1.00	4	$4.00
	2	10k single turn pot	$0.75	3	$2.25
	2	5v regulator	$0.50	1	$0.50
	2	Heat sink	$0.50	1	$0.50
	2	100k precision pot	$1.00	1	$1.00
	3	Perf board	$1.50	1	$1.50
	3	1 sided PCB per square inch	$0.10	40	$4.00

Fiscal Budget (continued)

		Item	Cost	Quantity	Total
Machine Shop Purchases					
February's Charges		Solderwick	$3.25	1	$3.25
	1	Nuts/Bolts	$0.28	1	$0.28
March's Charges	1	1 ft Velcro	$2.50	1	$2.50
		8-32 Tap	$4.00	1	$4.00
		Saw Blade	$15.00	1	$15.00
	1	Nuts/Bolts	$0.62	1	$0.62
Misc. Stuff		2 IR Sensors paid to M.T. Pace	$27.75	1	$27.75
		5 IR Sensors paid to M.T. Pace	$51.90	1	$51.90
		Voltage regulators	$15.14	1	$15.14
		PCB	$12.30	1	$12.30
		Line lasers paid to A. Targhetta	$54.95	1	$54.95
		TOTAL SPENT			**$229.79**
		TOTAL REMAINING			**$20.21**

Power Budget

Main Battery Pack:

	Quanity	Current Draw (mA)	Total Draw (mA)
IR Sensors	3	50	150
Motors	2	650	1300
NiMH Rating (mAh)		NiMH Total	1450
2500			
		Total Operating Time:	1.72 Hours

Appendix B: Localization Hardware Layout

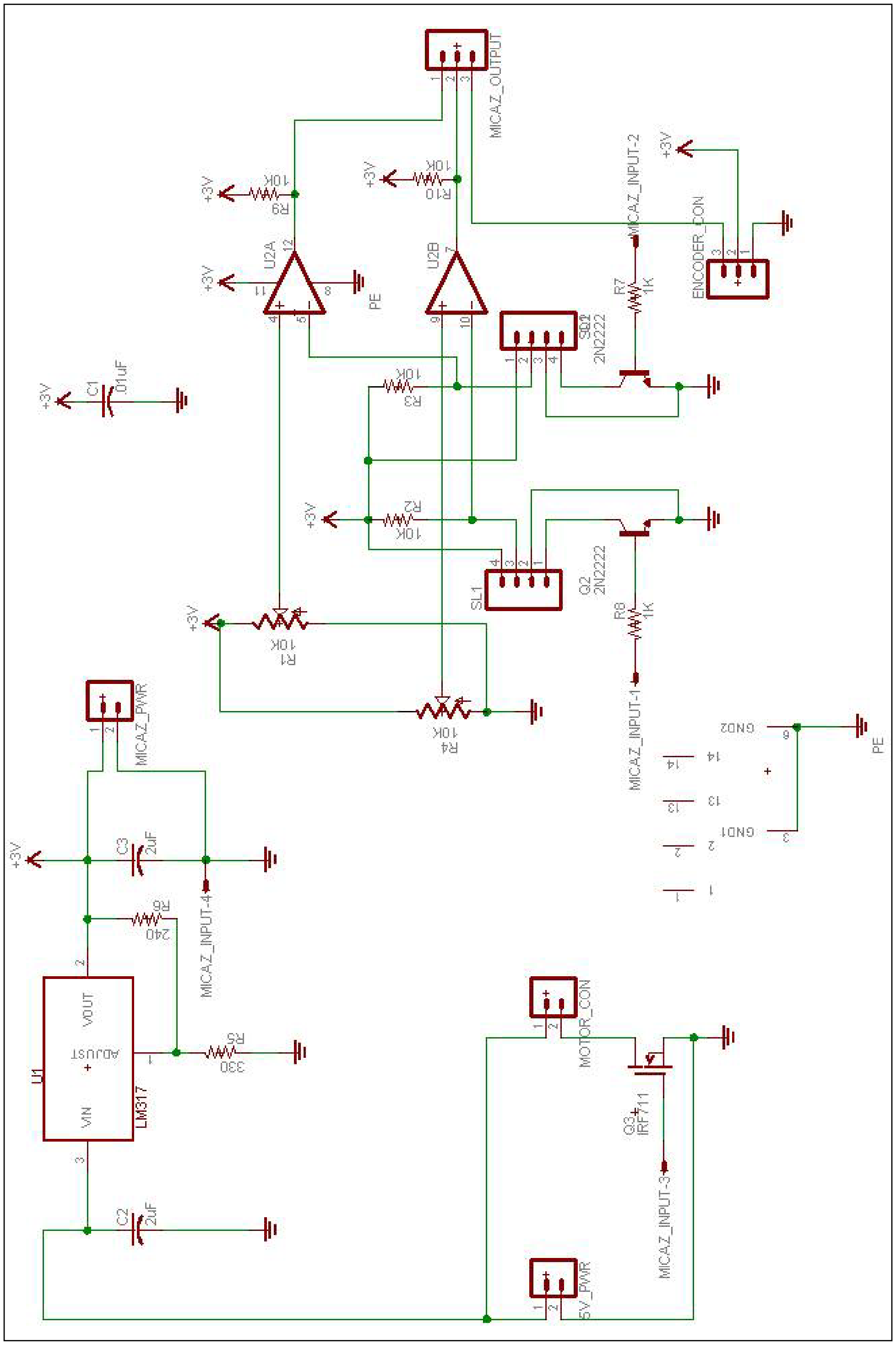


Figure B.1: The Localization base station schematic.

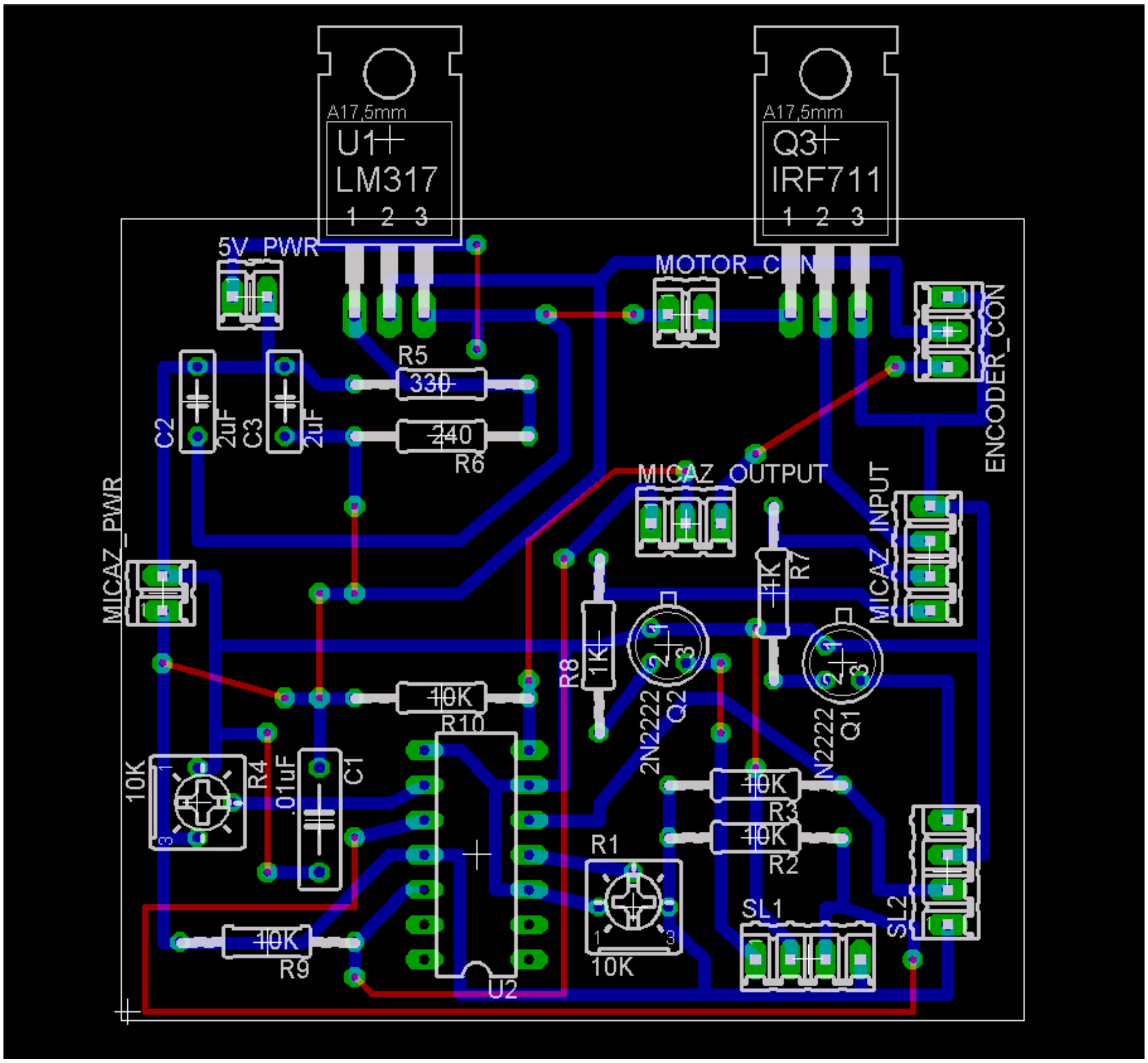


Figure B.2: The Localization base station board layout

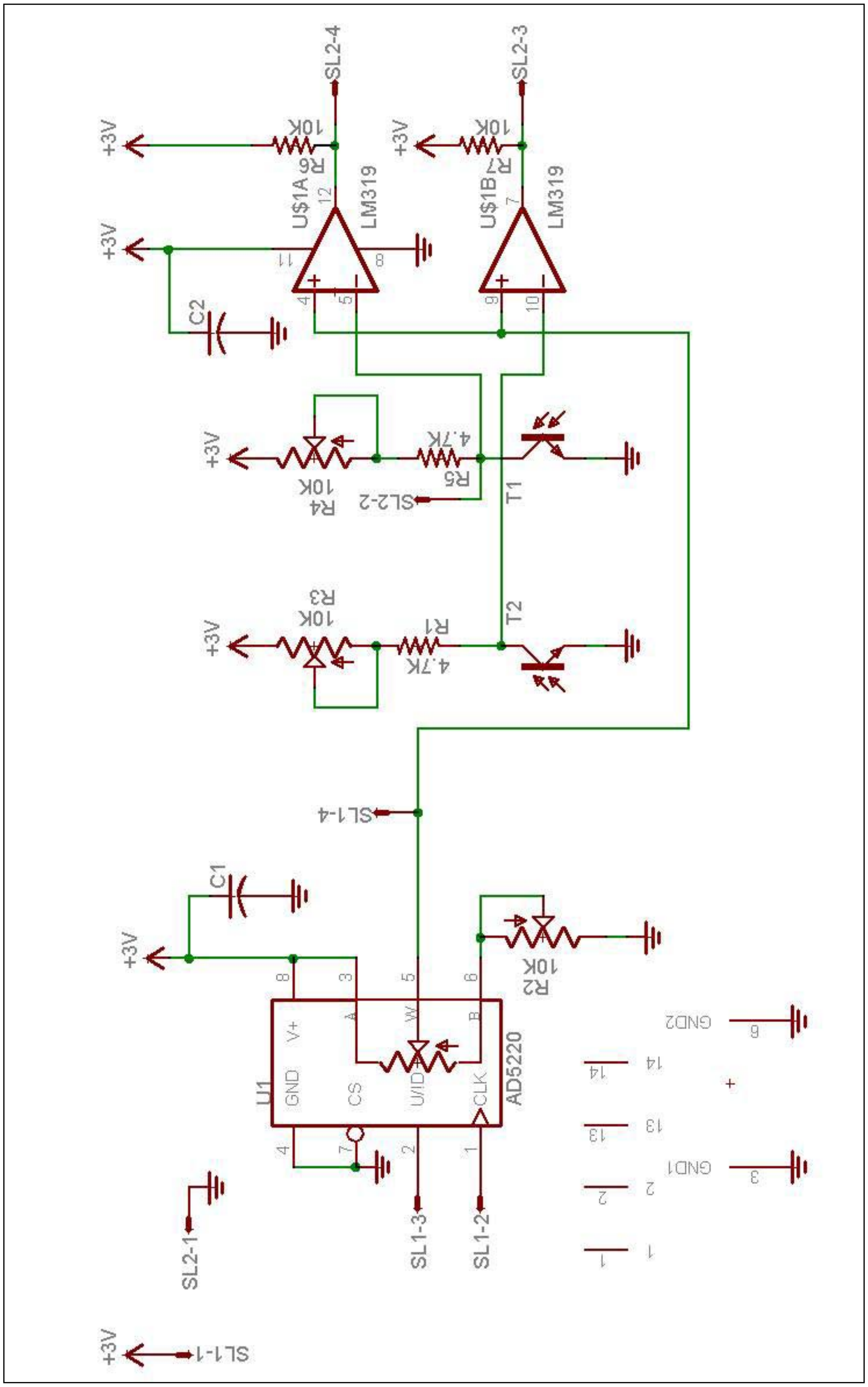


Figure B.3: The Localization sensor Schematic

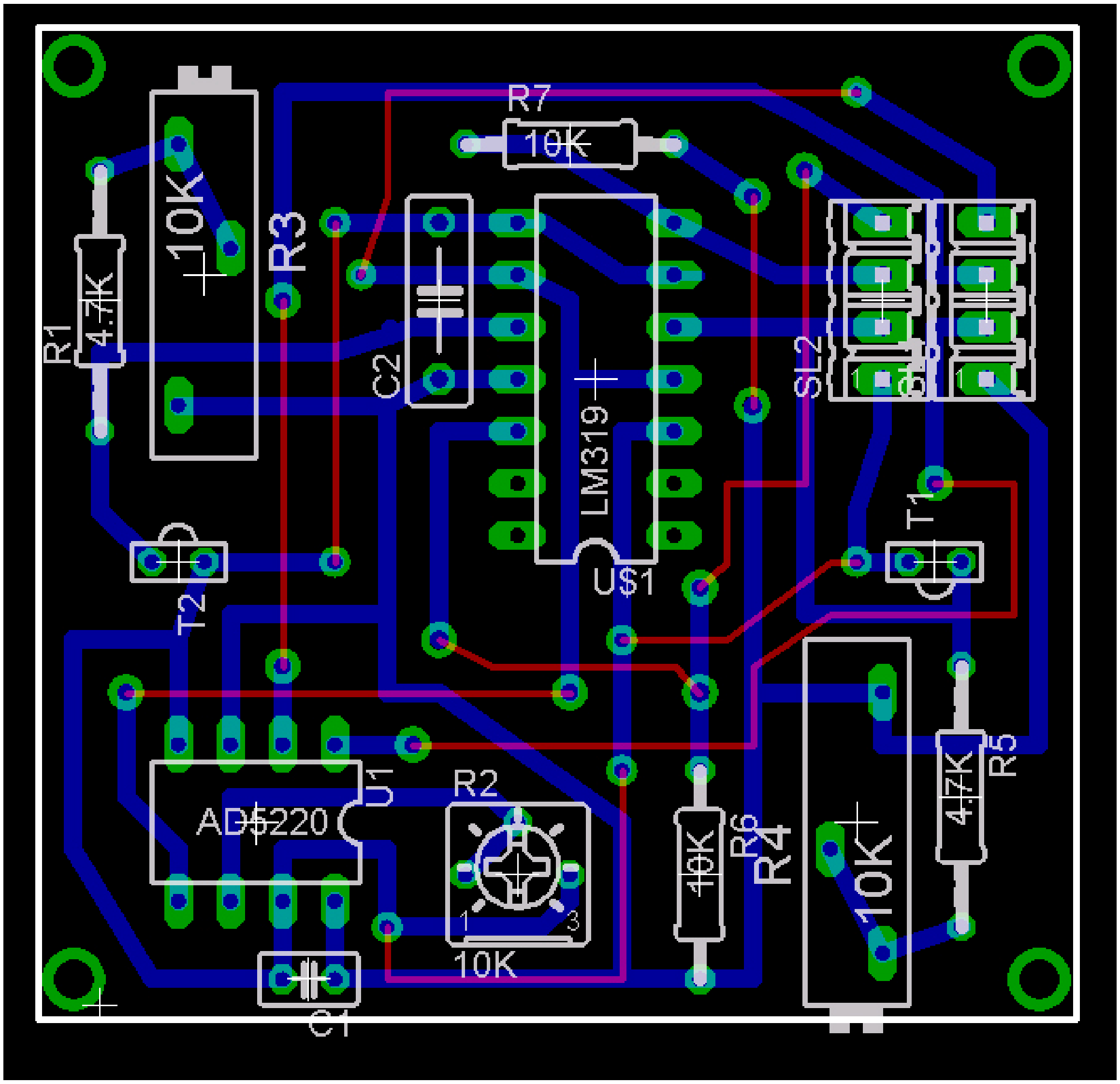
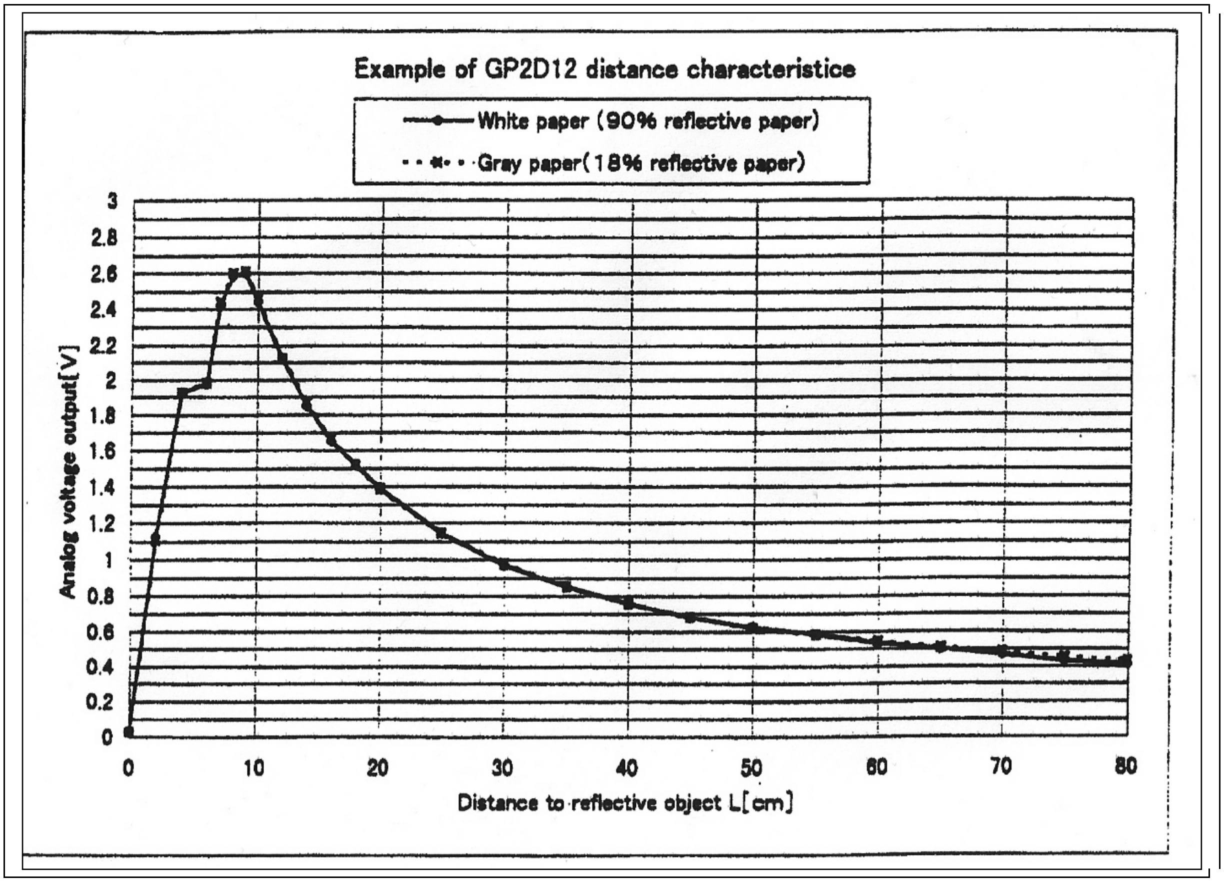


Figure B.4: The Localization sensor board layout.

Appendix C: Datasheets



© 2006 Sharp

Figure C1: A plot showing the ratio between analog output voltages to distance to an object for the GPD12 Sensors.

Appendix D: Code

*** Please refer to CD for all source code ***