Behavioral Algorithm Sensor Suite

The aim of the BASS project is to create and demonstrate a new sensor suite for behavioral robotics. This suite will allow for the configuration of the sensors it uses and behaviors that it provides to the end user. The suite should be able to be used as a base for small behaviors (such as obstacle avoidance, navigation, etc.) and future research into behavioral robotics. To demonstrate the BASS system, a small robot will be run through a maze extraction scenario. In order to accomplish this goal we will only consider sensors that work in an indoor environment, such as IR, ultrasound, and inertial measurement devices.

The basic design for the sensor suite is a three tiered architecture. Tier 0 comprises the sensors and filtering system. Tier 1 is made up of the behaviors, and receives data from the sensors (tier 0.) The behaviors then feed into the Tier 2 arbitrator. This arbitrator then decides what action the robot should take. Communication among tiers can be tailored to meet system resource constraints (i.e. lack of a certain class of sensor) via a cross layer configuration. This architecture will allow sufficient reconfigurability for the end user, while maintaining the organization necessary to implement the behaviors under different physical configurations.

12/07/2009

Jens Peter Kaps

Faculty Signature

Team Members:

Alex Behnaz
Brian Loop
Ehsan Foroudi
Sameer Dhawan

ECE 493
Behavioral Algorithm Sensor Suite (BASS)
Final Report

Team Members:
Alex Behnaz – Project Manager
Brian Loop – Technical Manger
Ehsan Foroudi
Sameer Dhawan

Faculty Supervisor:
Jens-Peter Kaps

Date: 11/30/2009

Table of Contents

1. Executive Summary

The aim of the BASS project is to create and demonstrate a new sensor suite for behavioral robotics. This suite will allow for the configuration of the sensors it uses and behaviors that it provides to the end user. The suite should be able to be used as a base for small behaviors (such as obstacle avoidance, navigation, etc.) and future research into behavioral robotics. To demonstrate the BASS system, a small Lego robot will be run through a maze extraction scenario. In order to accomplish this goal we will only consider sensors that work in an indoor environment, such as IR, Sonar, and inertial measurement devices.

The basic design for the sensor suite is a three tiered architecture. Tier 0 comprises the sensors and filtering system. Tier 1 is made up of the behaviors, and receives data from the sensors (tier 0.) The behaviors then feed into the Tier 2 arbitrator. This arbitrator then decides what action the robot should take. Communication among tiers can be tailored to meet system resource constraints (i.e. lack of a certain class of sensor) via a cross layer configuration. This architecture will allow sufficient reconfigurability for the end user, while maintaining the organization necessary to implement the behaviors under different physical configurations.

The work comprising the project can be broken up into 3 main tasks with multiple subtasks.

- **Hardware**
  - Prototyping
  - Integration/Final Build
- **Software**

  - Algorithms
  - Simulation
  - Sensor Characterization
- **Testing**

2. Approach

2.1. Overview

The mission of the BASS project is to develop a new sensor suite for behavioral robotics. This suite allows for the configuration of the on-board sensors and behaviors that it provides to the end user. The suite is configurable and expandable using simple software interfaces. This allows a person with no hardware experience to build an autonomous robot with a smaller time investment.

The suite is able to be used for basic behaviors (such as obstacle avoidance, navigation, etc.) as well as providing the computational horsepower for future research into behavioral robotics. To demonstrate the efficacy of the BASS system, a small robot platform was created to execute an extraction point scenario in a maze environment. Infrared (IR) and ultrasonic sensors were selected for proximity sensing, and inertial measurement devices were chosen for dead reckoning.

The sensor suite was designed with a three tiered architecture. Tier 0 is made up of the sensor hardware and filtering system. Tier 1 is made up of the software implemented behaviors. Tier 1 uses the data from the sensors (tier 0) to execute its tasks. Tier 2 is the uppermost level, and acts as an arbitration unit. The behavior’s outputs are gathered in Tier 2, and the arbitration algorithm decides what action the robot is to take. Communication among tiers can be tailored to meet system resource constraints (i.e. lack of a certain class of sensor) via a cross layer configuration. This architecture allows sufficient customization for the end user, while maintaining the organization necessary to implement the behaviors under different physical configurations.

The work performed on this project can be broken up into 3 main tasks with multiple subtasks.

Hardware

- Small robot (for demonstration)
- Sensor ring
- Hardware intergration
- Power supply circuitry and battery supply design

Software

- Algorithms
- Drive control system
- Sensor characterization

Testing

- Independent testing of each module
- Integration of hardware
- Inter-device communications
- Demonstration of extraction point scenario

2.2. *Specifications*

The BASS system provides spatial awareness, obstacle avoidance, and has expandability and maintainability inherent in the design. The system's standard configuration uses eight ultrasonic rangefinders. In addition, six infrared sensors are available on the platform. This allows for a 360 degree view of the obstacles around the robot

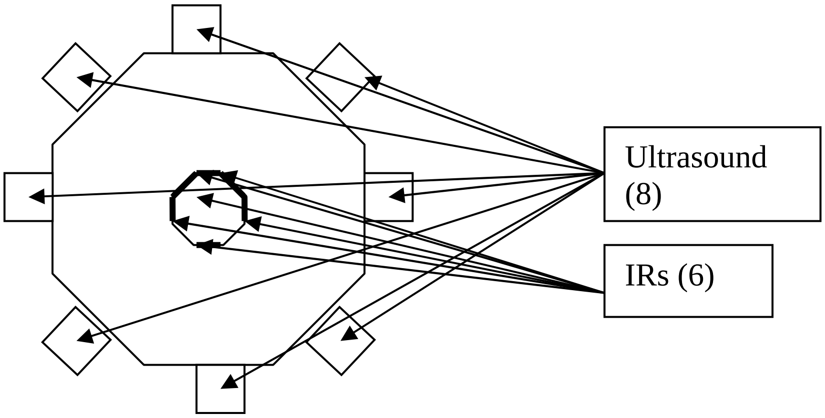


Figure 1: Diagram of demonstration system fully assembled and operational

Tier 1 collects data from the sensors using analog and digital busses. These sensor outputs are then filtered and made available to the behaviors running in software on the Gumstix. To demonstrate the action of the system, a basic set of behaviors were implemented and tested in a maze extraction scenario.

One of the main goals of the system was to allow for modification and development of new behaviors. This is accomplished by using a high-level design tool; the BASS Robot Configuration Application (BASS RCA). The RCA provides the user auto-source code generation, configuration options, and makefile generation. It allows for behaviors to be written in high level languages, including C, C++, Java, and Python. The entire process is aided by the RCA's GUI, allowing for a streamlined workflow for the end user.

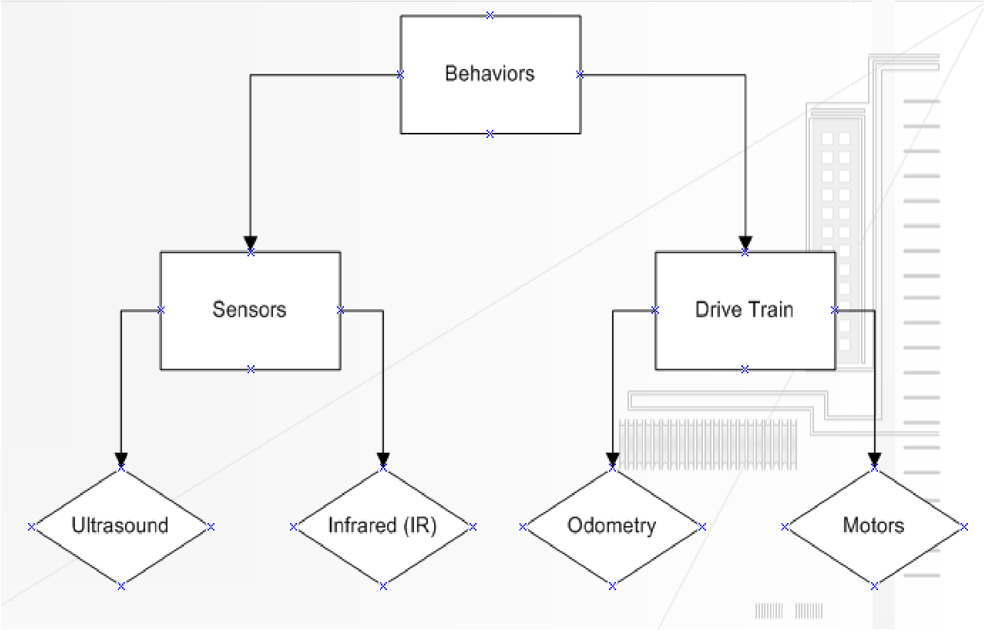


Figure 2: Overall conceptual diagram

3. Technical

3.1. Sensor Ring Overview

The sensor ring is a self-contained system that consists of sensors, glue logic, a microcontroller, and a real time operating system. This ring was designed with the goals of expandability and maintainability.

The standard sensor configuration for the ring is eight 40kHz ultrasonic rangefinders and six infrared rangefinders. The sensors selected were COTS parts. This allows the sensor ring to be both cost effective and utilize known high-quality systems. The computational and control base was provided by an Atmel ATMEGA 128 microcontroller on the Robostix breakout board provided by Gumstix. The Robostix served to do the controlling, sampling, filtering, and communication tasks associated with Tier 1 of the BASS architecture. Additional glue logic was used in the design to more efficiently use the resources available on the microcontroller.

The ring is assembled on a metal octagon for the demonstration platform. However, the ring may also be modified to be placed on any mechanical mount so long as the sensor wires reach back to the PCB.

3.2. Ultrasonic Rangefinders

Various ultrasonic sensors are available on the market at the time of this writing. Manufactures of note include Parallax, Devantech, and Maxbotix (see figure 3).

Figure 3: Commercial Off the Shelf (COTS) offerings of ultrasonic sensors at 40 kHz (left to right: Parallax PNG, Maxbotix EZ, Devantech SRF04)

Ultrasonic sensors come in various packages: double transducer element, single transducer element, 40 kHz, 235 kHz, and so forth. Research was done on the effect of the various physical construction parameters and effect of different frequencies on the sensor’s output data. A 40 kHz transducer has a different beam width than that of a 235 kHz transducer (see figure 4). This will affect the kind of obstacles the sensor will detect, as well the range at which it detects them (among other considerations). A single transducer element will weigh less than a double transducer element, but have more of a dead band.

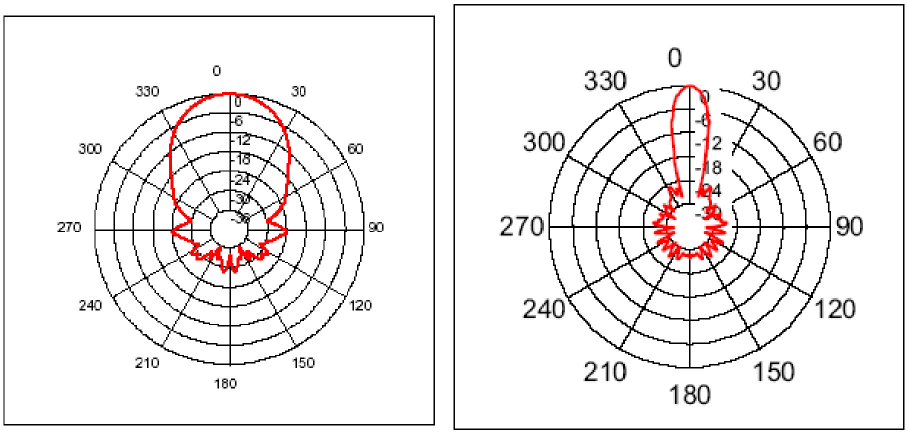


Figure 4: Relative beam width patterns of a 40 kHz (left) and 235 kHz (right) ultrasonic transducer [4]

After weighing the pros and cons of different ultrasonic products, it was determined that the Maxbotix EZ4 sensor would be optimal. This sensor provides the thinnest beam width (improved object resolution), 40 kHz operation, detects objects from 6 to 254 inches, and uses a single transducer element to reduce weight. The small form factor, low weight, and multiple interface options made it an attractive candidate.

The EZ4 provided numerous interfaces to extract data from the sensor. The two most obvious methods were the analog output, and the digital pulse width output. The digital pulse width output was selected, as it made efficient use of the relativity cheap GPIO pins on the Robostix whereas the analog output would have required use of the more precious analog to digital converter pins.

To avoid crosstalk between the various ultrasonic sensors, a time division multiplexing (TDM) circuit and control schema were devised. An external pair of glue logic chips (multiplexer and demultiplexer) were used to implement the ultrasonic range finding's TDM functionality. A diagram of the overall conceptual operation of the ultrasonic range finding system is shown in figure 5.

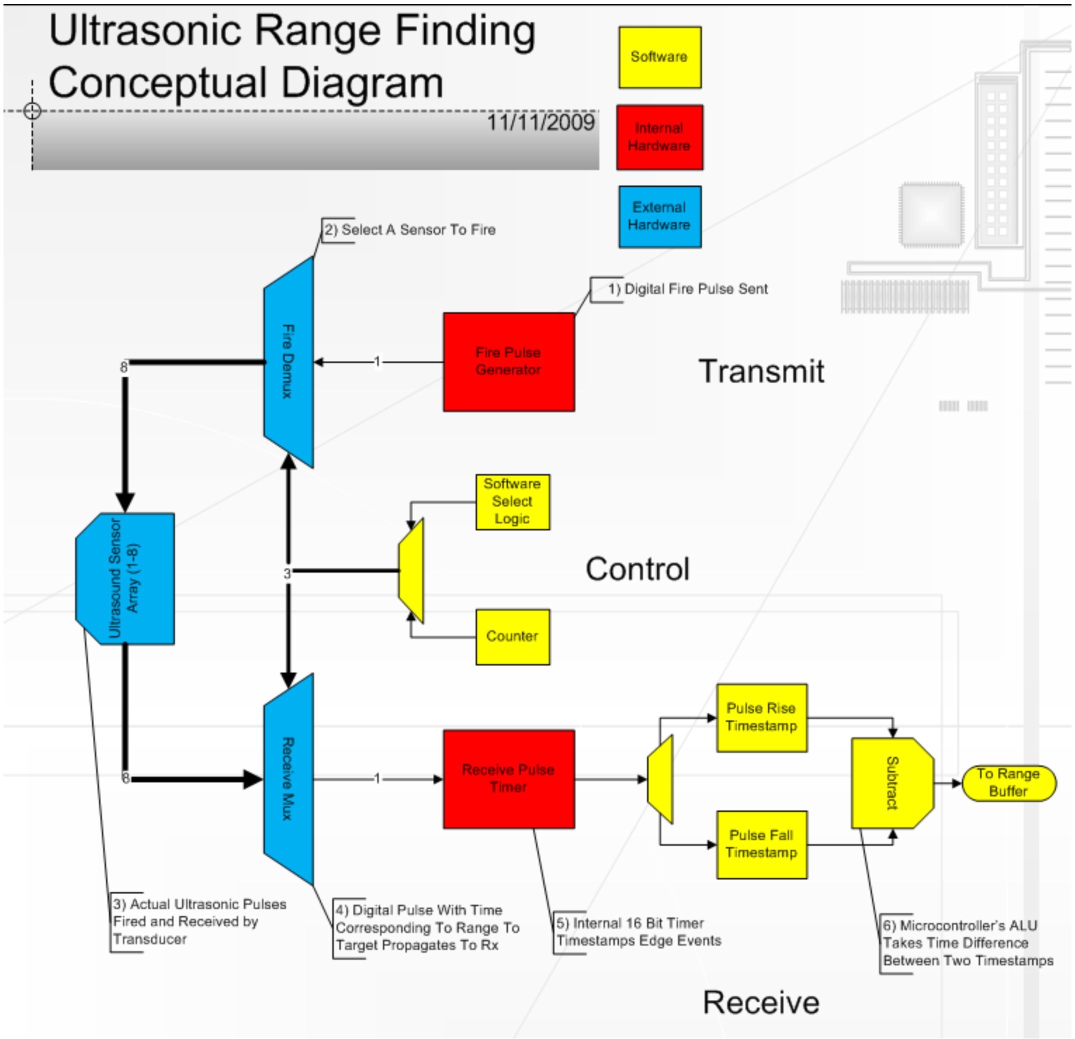


Figure 5: Conceptual diagram of the operation of the ultrasonic sensor array.

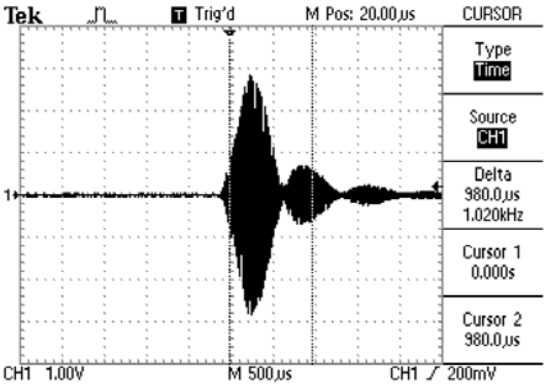


Figure 6: Ultrasonic return pulse echo for time of flight measurements

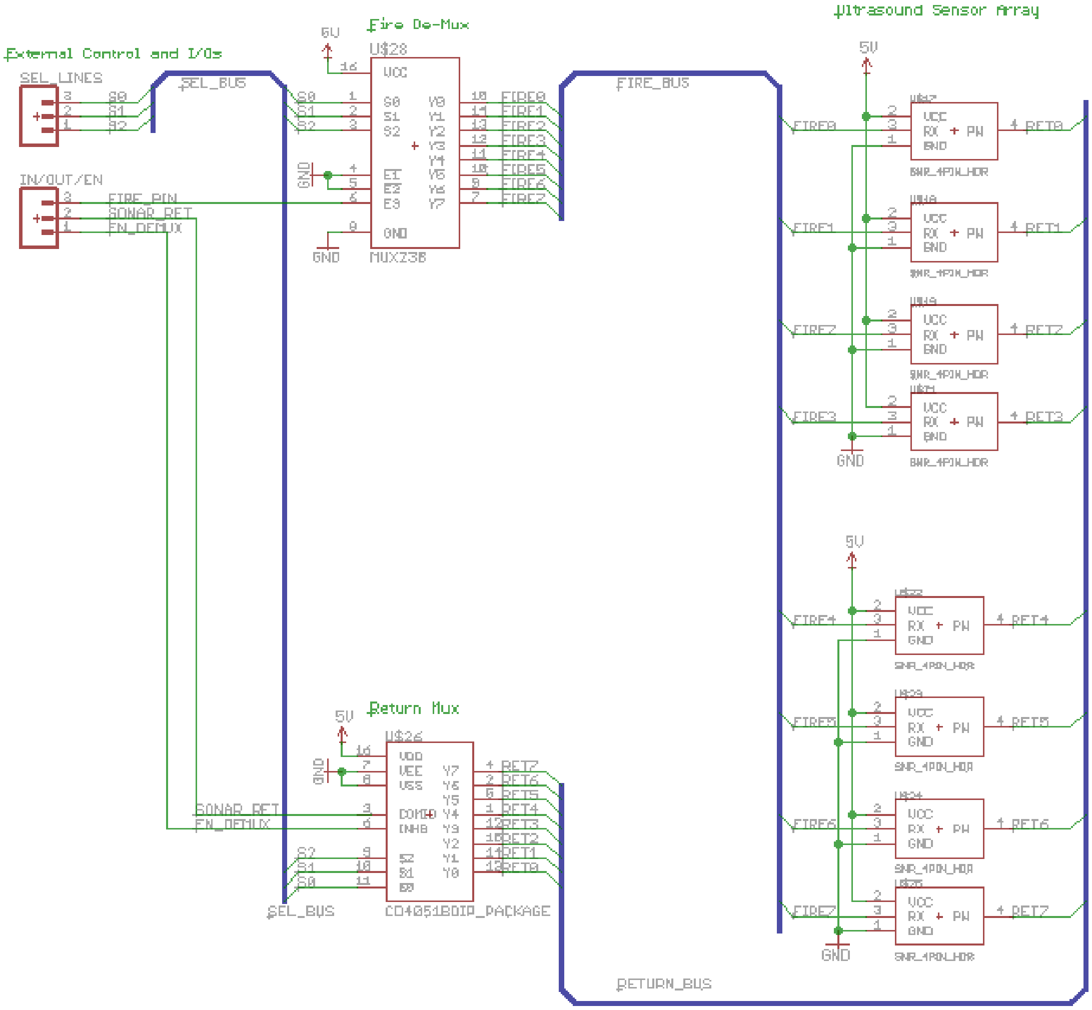


Figure 7: Schematic of PCB for TDM circuitry (ultrasound)

3.3. *Infrared Rangefinders*

The other class of rangefinder used on the BASS sensor ring was the IR sensor. Sharp is the primary manufacturer of these circuits, and the Sharp IR sensors are easily found as well as inexpensive.

The IR rangefinder class selected for the BASS system is the GP2Y0A21YK (figure 8). This sensor outputs an analog voltage proportional to the range from the target in the area of approximately 9cm to 45cm [6]. Outside of this range, the sensor outputs voltages that cannot be distinguished well. Figure 9 shows the non-linear nature of the sensor. Note that around 45 cm, the sensor values become not very distinguishable from each other. Experimentation showed that 45cm was a good cutoff range for the IR sensors.

Figure 8: Sharp IR sensor

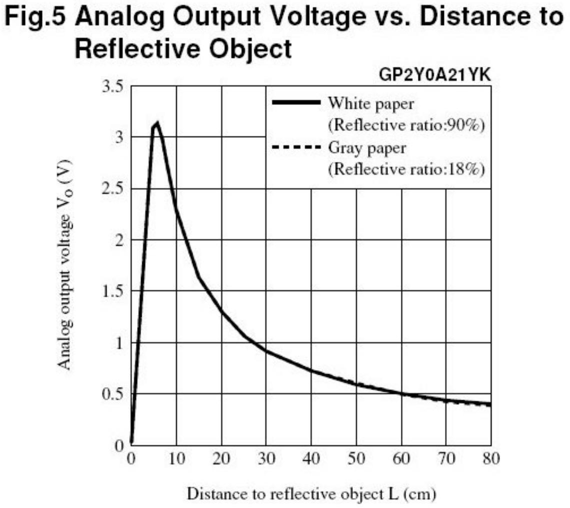


Figure 9: Non-linear transfer function of IR rangefinder

Another problem arises from the non-linear transfer characteristic. Note that when the sensor is less than 9 cm from the obstacle, it begins to produce voltages that map to much larger distances. This problem is well known, and can be solved with the following setup shown in figure 12. Figure 10 shows that by placing the IR sensors within the perimeter of the ring, the erroneous range values may be avoided. This is implemented on the demonstration platform in the form of the IR sensors being grouped in the center of the ring.

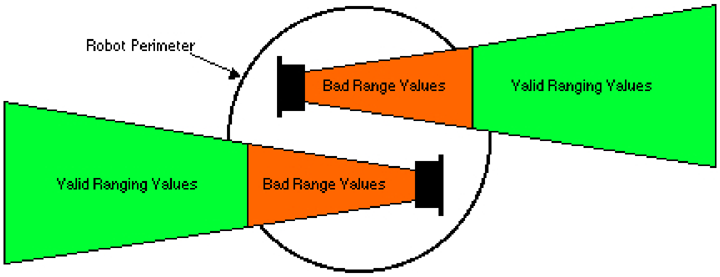


Figure 10: Sharp IR erroneous range data elimination configuration [7]

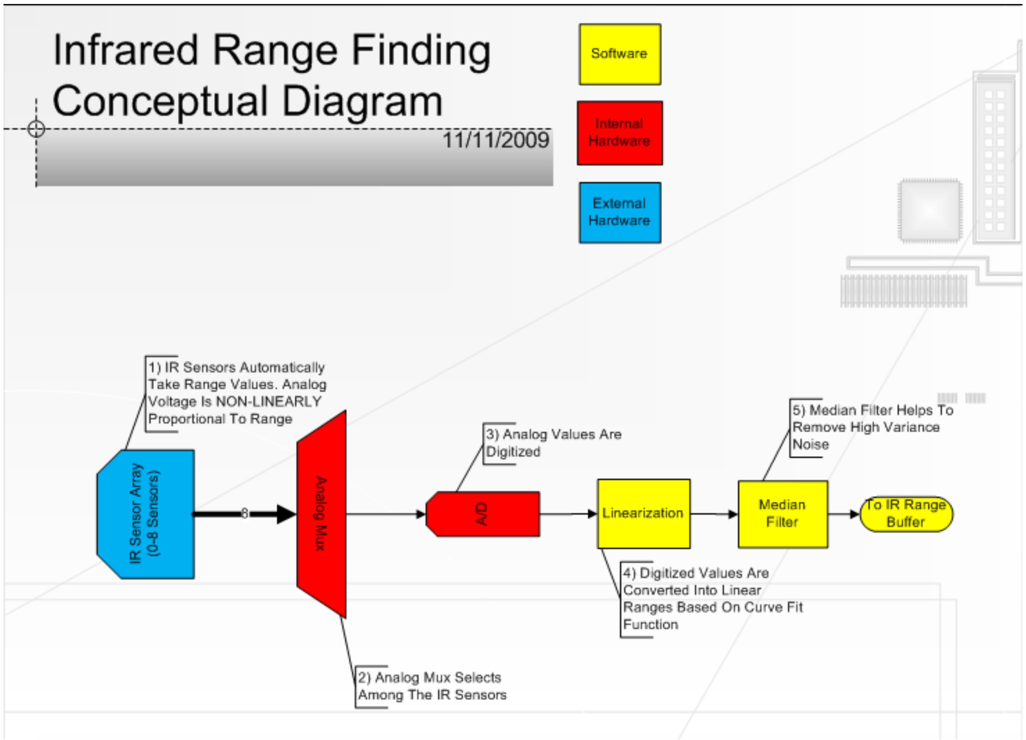


Figure 11: IR rangefinder conceptual diagram

3.4. Mechanical and Electrical Design of the Ring

The demonstration ring was designed with both electrical and mechanical components. The mechanical part of the sensor ring was designed using AutoCAD 2010. The dimensions, mounting brackets, and sensor positions were modeled. In addition, the drawing allowed for visualization of layout issues and physical spacing. Figure 12 shows the actual drawings used to prototype the mechanical sensor ring.

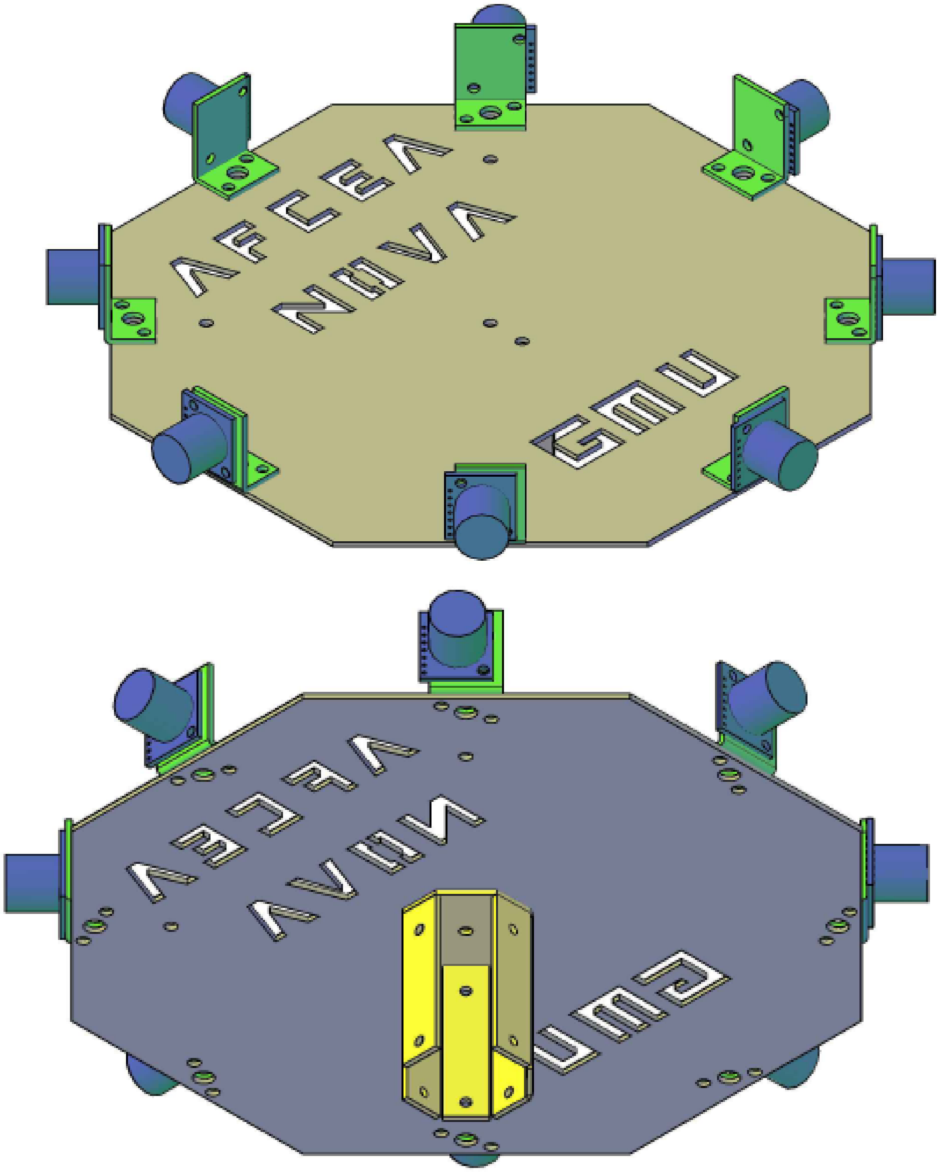


Figure 12: AutoCAD renderings of conceptual sensor ring

To implement the circuitry of the Ultrasonic TDM as well as gain access to the signals coming from the Sharp IR sensors, a PCB was constructed. This PCB also provided power to the sensors, and handled all the physical interfacing aspects. The outputs were routed to standard male headers to allow for ease of access from the robostix. The PCB was designed with Cadsoft's Eagle program, and sent to a 3rd party PCB fabrication house for physical implementation. The PCB was populated by hand.

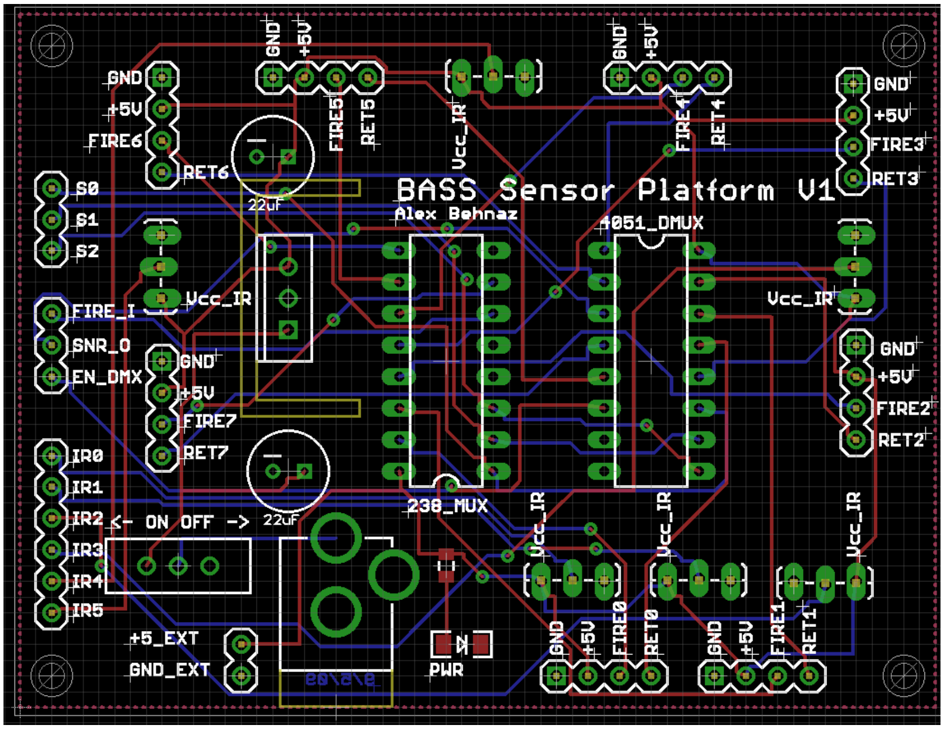


Figure 13: PCB layout for manufacturing

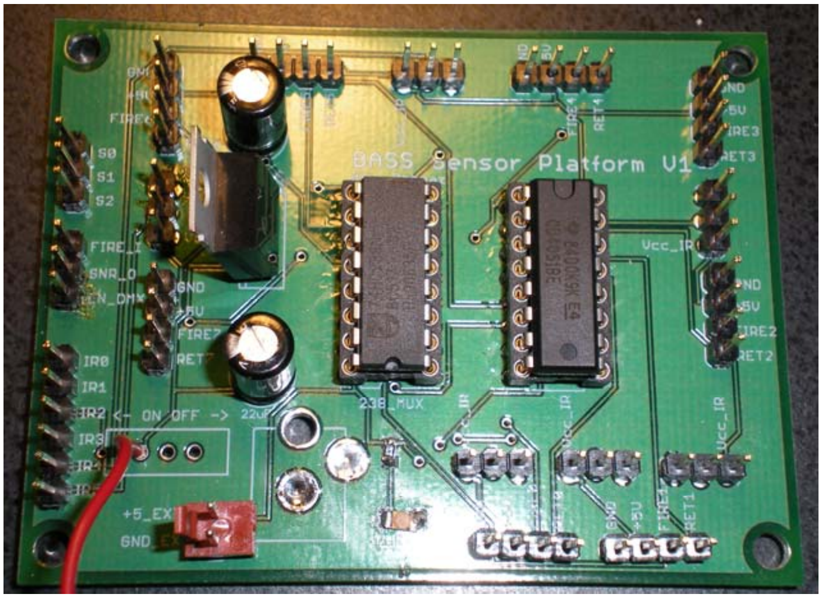


Figure 14: Manufactured PCB (partially populated)

3.5. *Software Design*

The software running on the Robostix implemented the control, range finding, TDM, filtering, and communication routines. To control all of these tasks as well as provide an extensible and easily modifiable platform for further development, the FreeRTOS light-weight operating system was used. This allowed for the tasks associated with each component of the sensor ring to be broken up logically in the code. In addition, with 23 ports to other architectures, the implementation in FreeRTOS ensures a quick turnaround if a different microcontroller was ever chosen to replace the Robostix. Figure 15 shows the hierarchal structure of the software design.

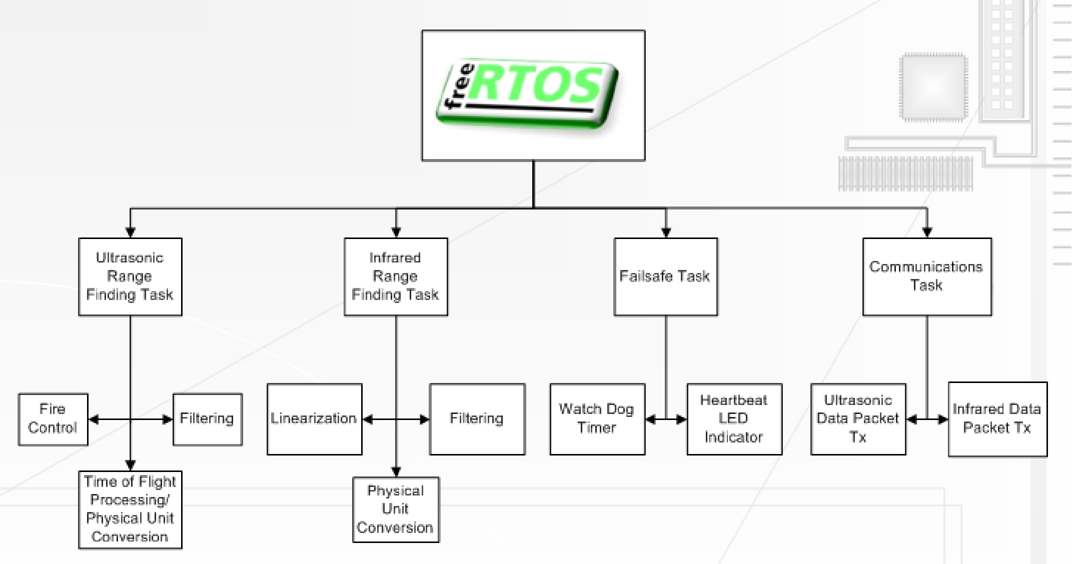


Figure 15: Hierarchal structure of tasks and software systems

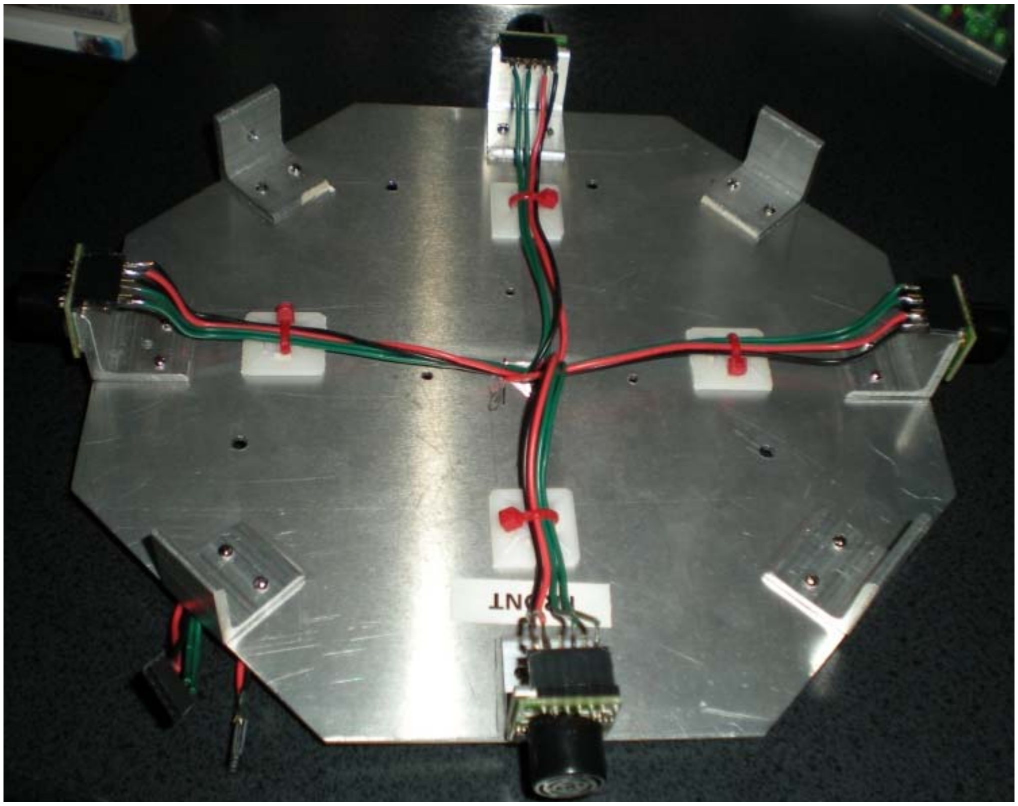


Figure 16: Final sensor ring in completed form.

3.6. Software

3.6.1. Robot Algorithm Core (RAC)

3.6.1.1. Software Architecture

The Algorithm software is an event based hybrid goal/behavioral architecture with sensor updates as the event triggers. Behaviors are activated and deactivated based on a finite state machine. When an event has occurred, all active behaviors that depend on that event are called to update their output. Below is a block diagram of the final layout.

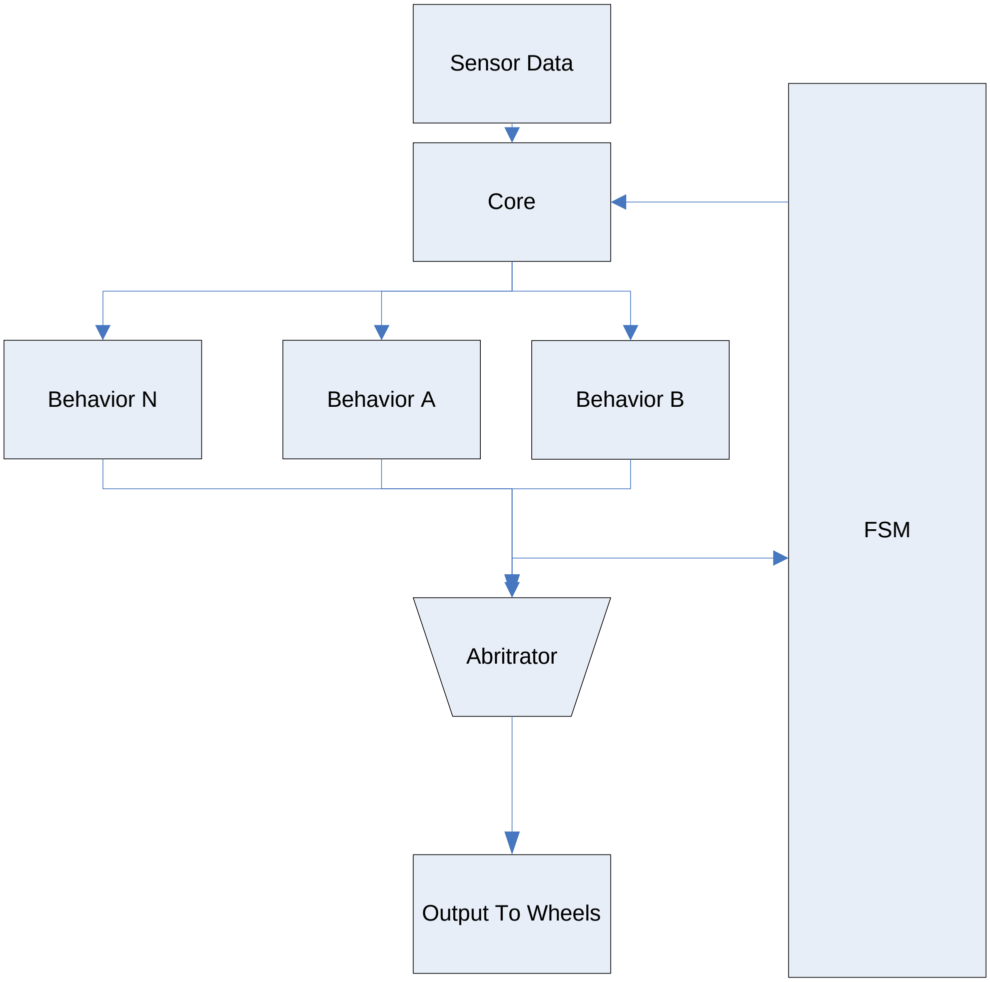


Figure 17: Tier 2 software architecture

Each behavior outputs two integers that indicate the direction of motion in vector form ie (s, angle) where s=speed and angle= the orientation based on the robots “North/Front” (0 degrees). The output are then weighted by a weight from 0 to 100 and normalized to 100. Then all vectors are added together and sent the driver controller. (Note that a behavior that has a 0 weight will not cause the robot to move in any direction, but still will cause state transitions) Behaviors are enabled by a finite state machine (FSM). Each state has a list of behaviors that are active in that state. The transitions between states are based on the output of one or more behaviors. Each behavior does not maintain any state information (any form of tracking) when it is deactivated. Each behavior is be initialized with the output value of the behavior that caused the transition that activated it, and the Behavior ID. The behaviors can be local or remote. A remote behavior resides in a different process than the RAC and communicates via sockets. Therefore a remote behavior can be written in any programming language that supports sockets (C, C++, Java, Python...). The FSM is imported as an xml file generated by the BASS Robot Configuration Application. The following default local behaviors are included in order to perform the maze demonstration.

- *Human*—Listens for console input and outputs different values base on key inputs. Used to transition from default (wait state to Go state).
- *Wall Follower*—Follows the right wall using IR's. If right IR's both read infinity then follows left wall. Uses a p-controller to follow the walls
- Move to Next Wall—Outputs an open direction for the robot to go. If the front is open ie. Range > threshold always outputs straight, else outputs the direction of most openness.
- Test Serial—prints out the sonar and IR data values to the user whenever they are updated.

3.6.1.2. Software Implementation

This software architecture was implemented using Linux POSIX libraries for serial communication and threading. The implementation uses several threads to listen for communication messages, form sensor ring, user console, and IP. The data is then passed into a set of pre-allocated threads to validate and process the incoming messages. If the message is a valid sensor update then, the appropriate event is passed to the event processing thread. The event processing thread is responsible for 1) calling all active behaviors both local and remote that are attached to that event, 2)Weighting the behavior outputs and calculating the output to send to the drive train 3) updating the FSM based on the outputs of the behaviors and performing any transitions if needed. Below is a diagram of the implementation.

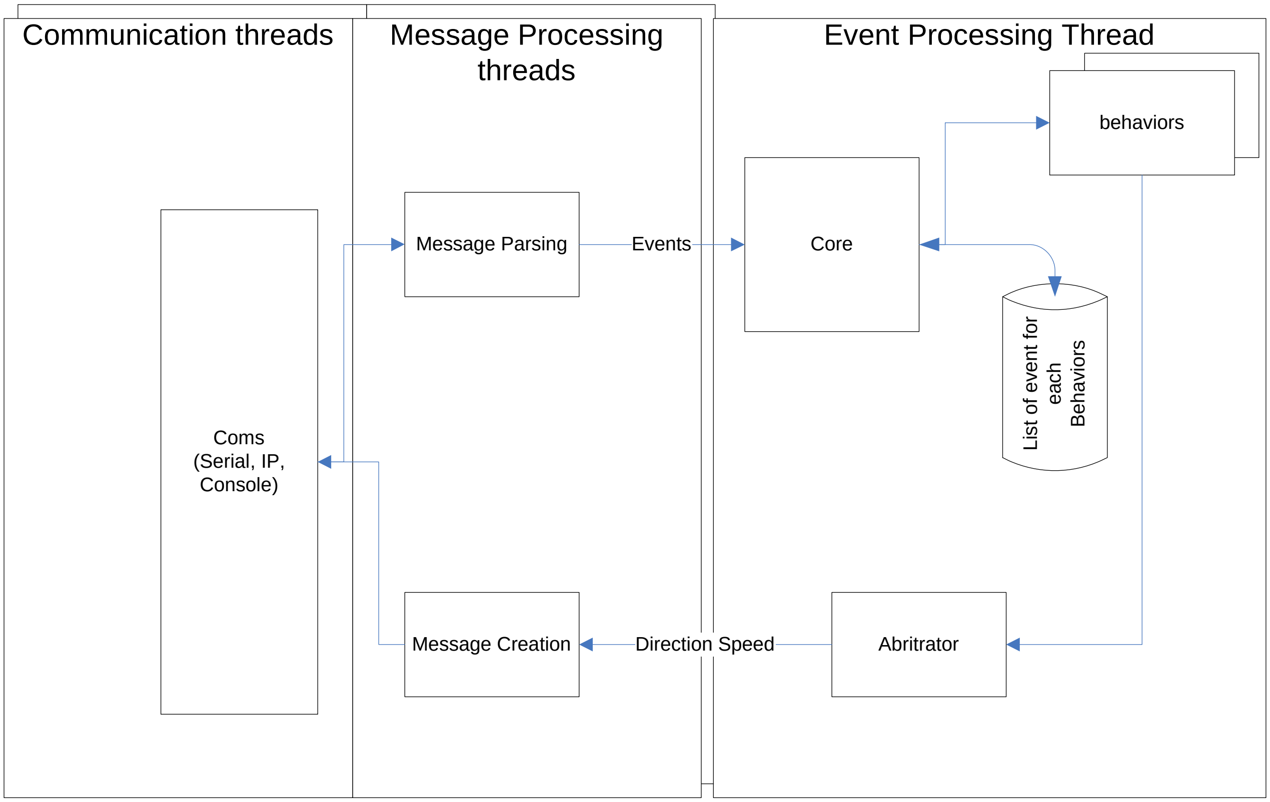


Figure 18: Multithreaded Implementation

3.6.2. Robot Configuration Application (RCA)

The Robot Configuration Application allows the user to define an FSM using a GUI. This application provides the ability to create/delete states, configure which behaviors are active in each state, which sensor information is used as a trigger to each behavior, which behavior output will cause a transition to a new state, auto-generate new behavior source code (C only), link new behaviors into the main system, and output an XML configuration file describing the FSM. In addition FSM, new behaviors, system sensors and/or other settings can be saved into project files to allow easy editing later. Below are some screenshots of the RCA and its dialog boxes.

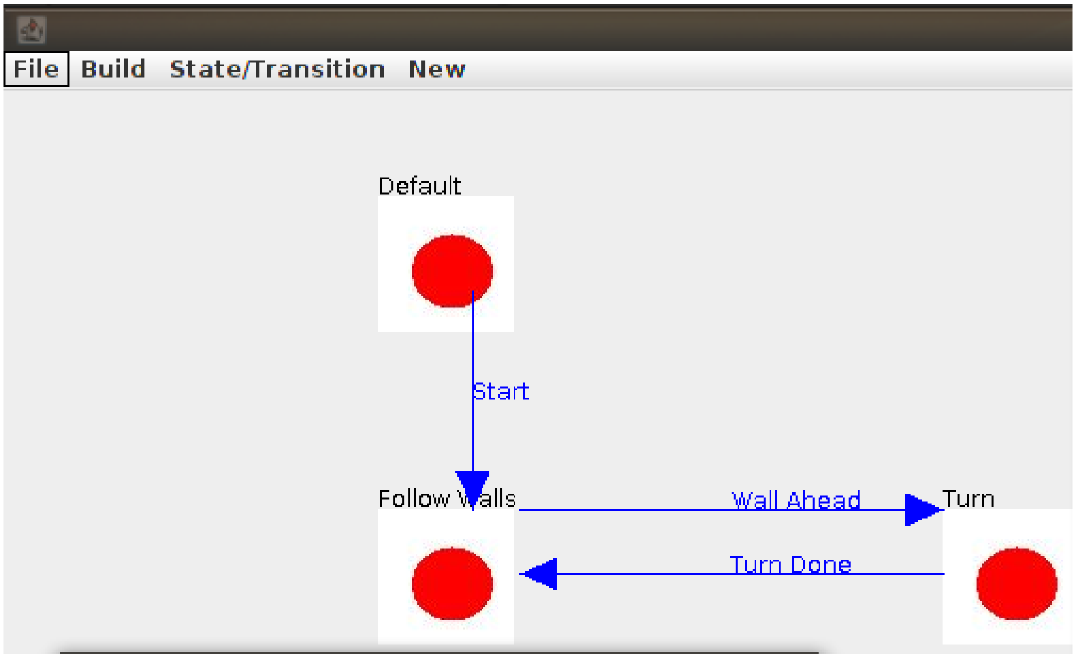
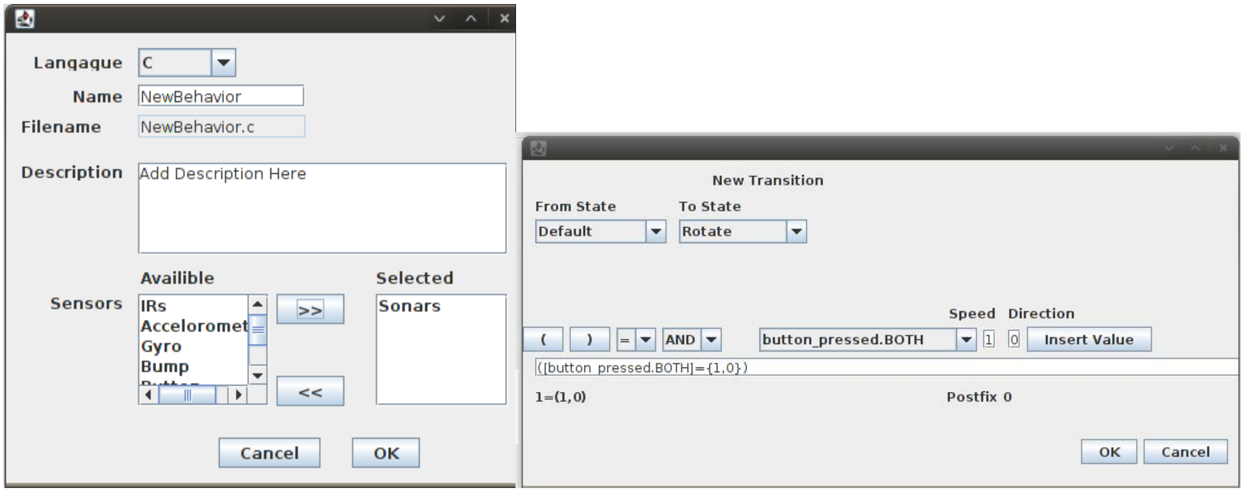


Figure 19: Screen shot of RCA GUI in action (actual FSM used in maze demo)



Langague: C
Name: NewBehavior
Filename: NewBehavior.c
Description: Add Description Here
Sensors
Availible: IRs, Acceloromet, Gyro, Bump
Selected: Sonars

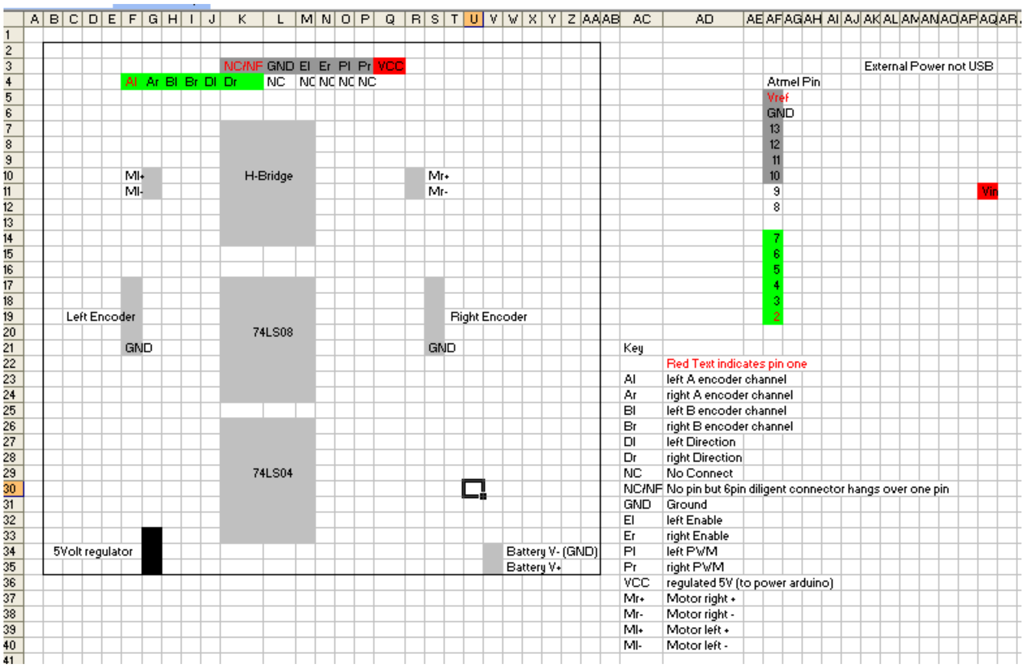
New Transition
From State: Default
To State: Rotate
Speed: 1
Direction: 0
button_pressed.BOTH
=
([button pressed.BOTH]={1,0})
1=(1,0)
Postfix 0

Figure 20: Screen shot of RCA GUI dialog boxes to create a new Behavior(left) and a new Transition (right)

3.7. Demonstration Platform

3.7.1. H-Bridge Circuit

The COTS H-Bridge circuitry that came with the robot base was not robust and was expensive at $50 per set. To provide a more efficient solution in cost and size, a custom H-Bridge circuit was implemented using 74LSXX series chips and a DIP H-Bridge housed in chip sockets. This allowed for easy replacement of damaged components, as well as ease of repair and modification for the end user. A pin diagram of the H-Bridge circuit is shown in figure 19. The H-Bridge circuit also holds the power supply circuitry. An 8V regulator sources the DC motors, and a 5V regulator sources the digital logic.



Key

Red Text indicates pin one

Al	left A encoder channel
Ar	right A encoder channel
Bl	left B encoder channel
Br	right B encoder channel
Dl	left Direction
Dr	right Direction
NC	No Connect
NC/NF	No pin but 6pin diligent connector hangs over one pin
GND	Ground
El	left Enable
Er	right Enable
Pl	left PWM
Pr	right PWM
VCC	regulated 5V (to power arduino)
Mr+	Motor right +
Mr-	Motor right -
Ml+	Motor left +
Ml-	Motor left -

Figure 19: Pin diagram of H-Bridge system

Power supply consists of two 7.2V battery packs in serieswhich being regulated in order to supply logic and motors which is shown in Figure 20, and two 3.5V Li-Po battery packs in series to supply the sensor ring.

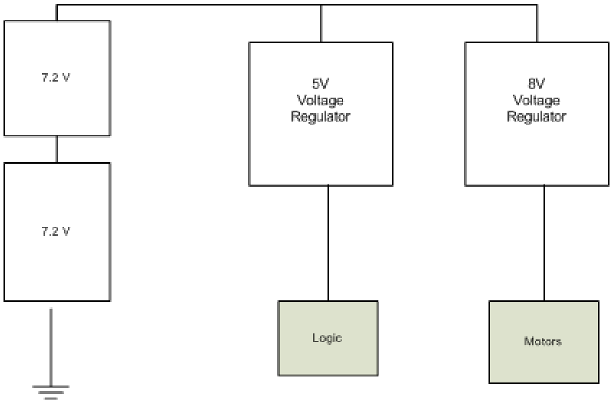


Figure 20: Top Level Design of Power Supply

3.7.2. Drive Controller

Drive controller is being supplied by 14.4V power supply which is being regulated by dual voltage regulators of 8V for DC motors and 5V for logic. Ball bearing caster wheel has been modified into a steering wheel by mounting a servo motor on top of a phenolic caster wheel which benefited the drivetrain by corrective motion inside the maze. Amt 102 Capacitive Encoder had been replaced by Optical encoder based on X and Y axis which communicates by micro controller via PS/2. Although the output data from optical encoder was valid, but experimented the higher controller is capable of handling the drivetrain more accurate inside the maze regarding to corrective motions.

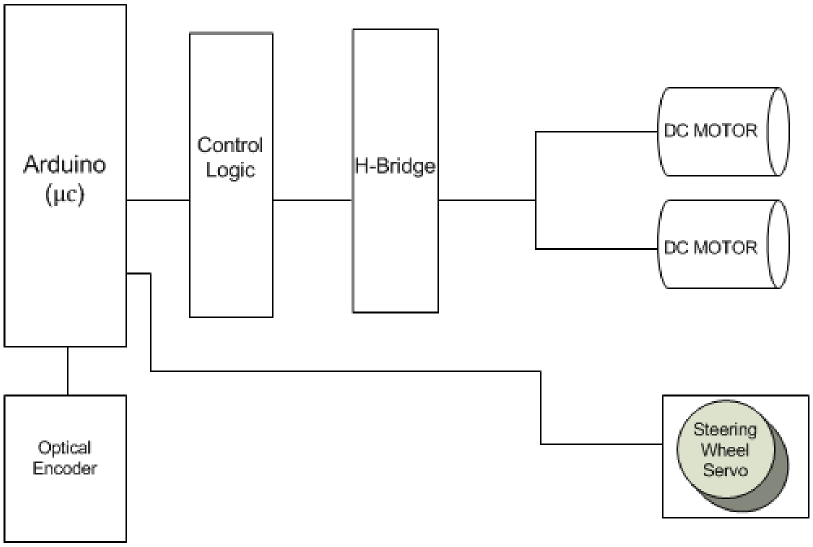


Figure 21: Top Level Design of Drive Controller

4. Experimentation

The BASS system is comprised heavily of both hardware and software. Experimentation was done on both the hardware components and the software that controls the system. This was accomplished by segmenting the platform into its functional blocks and testing each block with respect to its own functionality and its interaction with the entire system.

Initially testing was done with the Ultrasound and Infrared sensors to check for their accuracy and consistency in data.

Eight thousand points of data were taken at an increment of every centimeter. Through experimentation, it was found that the Infrared sensors had accurate data with a median filter for a range between 9-45 centimeters. The ultrasound sensors were accurate from 7- 24 inches.

After the hardware and software for individual sensors were laid out, testing for implementation and validation of Behaviors began. The platform was placed at varying angles in the maze to check for the functionality of the “Wall Follower” behavior and code changes were repeatedly made until the expected behavior was observed. Similar testing was done with other behaviors such as the “Turn”, “Move to Open Direction” and “Avoid Obstacle” behaviors.

5. Experimentation validation using evaluation criteria

Evaluation Criteria

Hardware

Test	Description	Criteria	Result
Motion Test	This test will verify that the BASS system is capable of driving servos in a differential drive configuration.	BASS system must drive two servos in both forward, backward, stop, and turn movements.	Passed
Control Test I	This test verifies the capability of the BASS system to provide controlled driving.	PID controllers will run both servo motors such that the platform will move in its intended direction with minimum error (i.e. if the system wishes to go straight ahead, the PID controllers will adjust for errors in the servo motor matching and provide for correct forward motion.)	Reworked
Control Test II	This test verifies the capability of the BASS system to provide controlled driving.	PID control was done using the IR sensors to detect the walls on either side. Correction was done using the back steering wheel. This allowed for the system to go straight ahead making necessary corrections.	Passed
Odometry Test (Encoders)	The BASS system needs to provide an odometry system to provide both distance measurements and rate feedback for the PID control.	Encoder system will provide a count of wheel movement verified by physical measurements.	Failed
Collision Test	The system must be able to detect a collision with a solid object	The system will detect when a collision has occurred with an object	Passed
Sound Based Range Finding Test	The ultrasonic sensors must be able to provide mid-long distance range finding capabilities for the system.	Ultrasound sensors will be able to detect an obstacle between 7 - 24 inches.	Passed

Infrared Range finding Test	The IR rangefinders must be able to provide short to midrange distance sensing.	The IR sensors must be able to detect an obstacle between 0 - 7 inches (0 inches being taken from the outside perimeter of the platform.)	Passed
Inertial Measurement Test	The IMU must be shown to provide information about the change in the robot's position	The IMU will demonstrate the ability to detect changes in the movement of the platform and correlate those measurements with physical units (i.e. robot has turned counter clockwise, approximately 45 degrees.)	Failed
Communications System Test	The BASS needs to provide a modern communications interface.	The BASS system will be able to implement a USB CDC slave device to communicate with a host system.	Passed

Software

Test	Description	Criteria	
Local Communications Test	To allow for the proper operation of the system, the software must provide for the ability to communicate with sub-modules.	The software will demonstrate the ability to facilitate local communications with sub modules (i.e. interacting with range finding and drive train sub modules)	Passed
Behaviors Test	The BASS system must provide different behaviors for the artificial intelligence routines (such as obstacle avoidance, goal finding, etc.)	Each behavioral algorithm is to be tested individually and proved to be operating with the expected behavior. The behaviors will not interfere with one another.	Passed
Arbitration Test	Due the conflicting nature of the behaviors implemented, the software must provide for an arbitration scheme to control which behavior is expressed.	The arbitrator will be able to control and select among the behaviors such that the proper behavior is selected for different situations. The arbitrator must be able to control which behavior is implemented at a given time, and by how much.	Passed

Goal Accomplishment Test	The system must prove its ability to solve the given task. For the demonstration purposes of this project, the maze extraction point scenario must be solved by the BASS system.	The BASS system will reach its end goal if it is capable of doing so (not blocked on all sides, or otherwise "unfairly" prevented from completing its goal.) The BASS system must be able of repeatedly accomplishing its goal. The BASS system must be capable of successfully completing its mission, even in the presence of some obstacles	Passed

6. Benefits

6.1. Portable/platform independent

The BASS System only requires a power source between 9.2V and 15V. The serial communication protocol between the algorithms and the drive controller may have to be modified. Besides this, there are no changes required the hardware or software in order to make it work on a different robotic platform.

6.2. Reconfigurable without writing code

The Algorithms make use of a finite state machine (FSM) to determine which behaviors are active. The BASS RCA allows the user to graphically create the FSM and output the configuration file to be loaded at the operating system’s boot up (in the Gumstix). For example, with the default behaviors, one FSM can be configured to provide a maze extraction, but by reconfiguring the states and their transmissions these same behaviors can cause the robot to search the walls of a room for obstacles.

6.3. *Expandable*

The BASS system allows for additional behavior to be easily added to the system. The BASS RCA will auto-generate the code for additional C or JAVA behaviors. In addition, if a behavior's processing takes too long or the system runs out of memory, behaviors can be run on a separate computer and communicate back to the robot via UDP over Bluetooth.

6.4. *Requires no knowledge of hardware to operate*

The main uses of the BASS System are software oriented. The user can move the system to another robotic platform without looking at any hardware datasheets. The only thing the user must do is provide the power and connect to the clearly marked pin connections. Most importantly, all of these changes can be accomplished without any code changes on a microcontroller.

7. Potential uses

The BASS System could be used in connection with a drive system for teaching the basics of robotics to freshmen (as is done at George Mason in CS 101). The BASS System would work extremely well for this type of application because it would provide an intro to event based programming and would not require the students to write large amounts of code in order to create a very complex robotic application. In addition with Bluetooth and Serial backup for students who do not have Bluetooth adapters, the students would be able to easily connect and test their designs.

Other possible uses include being used to rapidly prototype a larger more complicated Robot. This system could be expanded to include other sensors read in from the Gumstix.

8. Alternatives

An alternative to BASS project is the FlockBots project which has been designed at George Mason University. The FlockBots system costs approximately $800, and has less functionality and motor power than the BASS system. The FlockBots are equipped with a differential drive powered by continuous rotation servos with a caster wheel whereas the BASS platform is equipped with brushed DC motors with a caster wheel. The FlockBot system is equipped with five Sharp IR distance sensors to sense objects whereas the BASS system has six Sharp IR distance sensors along with 8 ultrasonic sensors. Power supply to the FlockBots is a removable 5-A-cell NiMH battery pack. The power supply to the BASS system is two sets of 7.2 V battery packs in series which is regulated by dual linear voltage regulators on the H-Bridge circuit board.

9. Maintainability

The BASS project is expandable and adapts to the user’s requiremnts. It can be plugged into any platform that will accommodate its constituent parts (computational modules, sensors, etc.) The user is not required to perform any software modification or development to run the BASS system. The BASS system is a true plug-and-play solution. Electronic components need to be protected. The ultrasonic sensors, Sharp IR sensors, Gumstix, Ardunio, and Robostix are the most sensitive components of this project.

10. Replacement/retirement

The custom design of the H-Bridge circuit takes into account the needs of the end user. The components used on the H-Bridge are standard 74LSXX series chips, and a standard H-Bridge IC. These parts are inexpensive and are in DIP form allowing for ease of replacement. The power solution for the BASS system makes use of standard rechargeable battery packs for economical re-usage. The batteries are on polarized connectors, preventing polarity reversal mistakes and making the connection process simple.

The Sharp IR sensors and ultrasonic sensors are standard COTS components and can be easily replaced. The modular design of the system makes replacing a part as simple as plugging a new part in.

The sensor ring uses a watch dog timer and other fail safe techniques to keep the system running, even in face of a sensor failure.

Upon the end of the useful lifetime of the Bass system, the batteries and printed circuit boards must be returned to a special center for recycling (like any digital camera or other electronic component). The remaining components are primarily metal, and can be re-used or recycled into other projects easily.

11. Administrative

Funds Spent on Parts:

Robotic Starter Kit	79	1	79
Base Plate Expansion Kit	6.99	1	6.99
Ball Caster Omni-Directional Metal	5.95	1	5.95
1100 mAH Batteries	12	4	48
SCA3000 (Accelerometer)	45	1	45
Gyro Breakout Board (MLX90609 - 150 degree/sec)	60	1	60
USB Cable A to B - 6ft	4.95	1	4.95
Arduino	35	2	70
Continous rotation servos	15	2	30
Gumstix verdex pro XM4-bt (with u.fl antenna)	159	1	159
robostix	49	2	98
Amt 102 Capacitive Encoder	30	2	60
Standoffs	0.72	10	7.2
Jumper Wires Premium F/F	3.95	1	3.95
PCB	50	1	50
Bluetooth Adapter	15.57	1	15.57
Shipping	27.85		27.85
Mux, Demux Glue Logic	0.5	2	1
H Bridge DIP Chips	1.5	4	60
Vector Board	3	1	3
H Bridge Replacement	25	2	50
Total			**$885.46**

Parts available already/borrowed parts:

Name	**Quantity**
NiCad Rechargeable Battery Packs	6
Caster Wheel	1
Zip Ties	10
Zip Tie Bases	4
180 Degree Servo	1
On/Off Switch	1
Wires	
Maze Boxes Masking Tape	
Glue Logic Chips (for drive train)	
Crimp Pins	

Time Spent:

Timeframe – 08/31/2009 – 12/03/2009

Team Member	**Hours Worked**
Brian Loop	382
Alex Behnaz	281.5
Sameer Dhawan	216
Eshan Foroudi	225
Labor Cost: **$25/Hr**	**$27,612.50**
Total Cost	**$28,497.96**

12. Lessons Learned

- A simple design is much preferable to a complicated design. Simple designs come together quickly, and are easy to debug.
- Power on mobile systems is precious. A good power design is often hard to do, and can’t be done quickly.
- Voltage regulators can get *hot*.
- Seek advices early, experience counts and people who have been in the field will have run into similar problems earlier and will provide useful information.
- It is important to have multiple backup plans.
- Do not spend too much time debugging something, think of alternatives early.

13. Problems/Surprises

- Platform has been causing problems. Non-linear transfer function of motors is causing delay on PID development. Large motor mismatch requiring use of different tactics for control algorithm.
- Time lost in experimenting with the old Lego Platform that was not suitable for the Sensor Ring. Eventually ended up ordering a new platform that allowed for greater configurability and had onboard motors better than the servo motors that were initially used.
- Encoders on new platform do not provide quadrature data accurately. Time spent making the encoders function in a desirable manner.
- Mechanical problems and inaccurate data from wheel encoders forced change of project plans to use optical mouse as encoder.
- Data from the Optical Encoder (PS2 Mouse) was not high enough resolution to run a PID control from. Eventually used a servo connected to a wheel to serve as a steering wheel.
- Power Brownout on the Robostix and the Arduino (too much current draw).
- Time lost to experiment and test alternative solutions for power supply.
- Power issues with linear regulators not supplying a constant output voltage and current. This forced change of project plans to use a Switched Mode power supply that required further time to research and construct.
- Time consumed on learning the FreeRTOS software.
- Time lost to re-designing ring for manufacturing.

14. References

[1]Parallax PNG

http://www.parallax.com/tabid/768/ProductID/92/Default.aspx

[2]Maxbotix

http://www.sparkfun.com/commerce/product_info.php?products_id=639

[3]Devantech

http://www.acroname.com/robotics/parts/R93-SRF04.html

[4]Ultrasonic frequency effects information

http://www.robot-electronics.co.uk/htm/srf235tech.htm

[5] Ultrasonic range finding implementation

Renesas "Ultrasonic Rangefinding with SLP" March 2004, AN0403006/Rev.1.00

[6] Linearization of Sharp IR sensors
http://www.acroname.com/robotics/info/articles/irlinear/irlinear.html

[7] Theory of operation – Sharp IR sensors
http://www.acroname.com/robotics/info/articles/sharp/sharp.html

[8]VTI Technologies : Digital Multi-Axis Accelerometer

http://www.vti.fi/en/products/accelerometers/

[9]Melexis: Digital Gyroscope

http://www.melexis.com/Sensor_ICs_Inertia/Sensor_ICs_Inertia/Evaluation_board_for_MLX90609_674.aspx

[10]LSI/CSI: Wheel Encoder Chip

http://www.lsicsi.com/

[11]CUI Inc: Capacitive Wheel Encoders

http://www.cui.com/adtemplate.asp?invky=762111

[12]Atmel: Microcontroller

http://www.atmel.com/

[13]Maxbotix: Ultrasonic Sensors

http://www.maxbotix.com/

[14]Sharp: Infrared Sensors

http://www.acroname.com/robotics/info/articles/sharp/sharp.html

[15]Union Battery: Lithium Polymer Battery

http://www.sparkfun.com/commerce/product_info.php?products_id=339

Special Thanks to:

William Wadley

A'amer Almujahed

Dr. Beatty

Dr. Luke, Randy, Brian, Keith

Appendix A Proposal

Executive Summary

The aim of the BASS project is to create and demonstrate a new sensor suite for behavioral robotics. This suite will allow for the configuration of the sensors it uses and behaviors that it provides to the end user. The suite should be able to be used as a base for small behaviors (such as obstacle avoidance, navigation, etc.) and future research into behavioral robotics. To demonstrate the BASS system, a small Lego robot will be run through a maze extraction scenario. In order to accomplish this goal we will only consider sensors that work in an indoor environment, such as IR, Sonar, and inertial measurement devices.

The basic design for the sensor suite is a three tiered architecture. Tier 0 comprises the sensors and filtering system. Tier 1 is made up of the behaviors, and receives data from the sensors (tier 0.) The behaviors then feed into the Tier 2 arbitrator. This arbitrator then decides what action the robot should take. Communication among tiers can be tailored to meet system resource constraints (i.e. lack of a certain class of sensor) via a cross layer configuration. This architecture will allow sufficient reconfigurability for the end user, while maintaining the organization necessary to implement the behaviors under different physical configurations.

The work comprising the project can be broken up into 3 main tasks with multiple subtasks.

- **Hardware**
  - Prototyping
  - Integration/Final Build
- **Software**
  - Algorithms
  - Simulation
  - Sensor Characterization
- **Testing**

Problem Statement

Need:

Robotics is a broad field, comprised of professionals from many backgrounds. One of the fields involved in robotics is Computer Science. The basic training for these professionals often does not include much hardware exposure. However, robotics cannot be practiced without the hardware necessary to digitize and make sense of the outside world. Therefore, there is a need in the market for an easily configurable, extendable, and robust sensor platform to facilitate behavior based robotics.

The data that is of critical importance to behavior based robotics (and mobile robotics as a whole) is:

1) What is the orientation of the robot in physical space?
2) Where is the robot with respect to its objective?
3) What obstacles impede the motion of the robot?

Based on interviews with Computer Scientists involved in robotics research and applications, the system needs to provide several off the shelf functionalities. The behaviors that the system provides must be configurable by the end user. The drive train control system needs to be transparent to the end user's application software. In addition, the platform must provide a modern communications interface. Many embedded systems make use of serial protocols that are no longer compatible with standard personal computers. The most obvious option available is the use of the USB protocol. Alternatively, a wireless communications would be preferred.

Objective:

The goal of this project is to design and implement a prototype sensor suite that enables the end user to approximate the orientation of the robot, the position of the robot in respect to its end objective, and to detect the objects that prevent the robot from achieving its goals. In addition, the BASS system will provide a modern, simple, and efficient interface to a host system. This interface will include the use of a Human-Machine Interface (HMI.) The interface will be a modern communications device, namely the USB protocol, with possible expansion to wireless protocols. Because of time and budget constraints, the initial phase of this project will focus on indoors environments, specifically a maze. The simple configuration and drive train identified in the need will be provided.

In order to demonstrate that this sensor suite operates in the manner outlined by the need, an extraction point scenario will be run. The extraction point scenario will take place in an sheltered indoor environment. This scenario will test the ability of the BASS platform to provide accurate and reliable sensor data to the end user. In addition, it will demonstrate the platform's built in behaviors capability to perform their intended task.

Problem Analysis and Requirements Definition

The project is based around the field of autonomous robotics. The robotic architecture that this unit will be designed to provide is behavior based. Our need statement identifies a requirement for spatial awareness and obstacle avoidance.

In order to provide spatial awareness, the algorithm will make use of the data from the range finding equipment, Inertial Measurement Unit (IMU), and other instrumentation equipment. The IMU will need to have multiple components in order to relay the required data to the end user. Accelerometer x and y channels along with gyroscopic data will be used to approximate the position of the robot with respect to an initial position. To further facilitate this goal, wheel encoders will be deployed to measure the kinematics of the platform. These design constraints necessitate the use of control electronics with both analog and digital interfaces.

The range finding equipment allows for obstacle avoidance. In addition, the algorithm can make use of the range finding data and pre-determined environmental knowledge to enable a more intelligent obstacle avoidance system.

In order to demonstrate the stated objectives, a demonstration platform will need to be constructed. The platform will be implemented via Legos, due to their modularity and inexpensive nature. To support the differential drive platform, two driving wheels and one caster wheel will be required. To facilitate the protection of the motor apparatus, slip gears are a good candidate to add within the drive system. Gearing ratios will also be employed to change torque and velocity parameters of the drive system. Preliminary research points towards the use of continuous rotation servo motors or stepper motors as the drive mechanism of choice. To provide servo control for the platform, a digital control algorithm has to be implemented.

Due to the autonomous nature of the intended application of the platform, the system must have its own onboard power supply with regulation. The most user friendly method of implementation is providing an interface to a rechargeable battery. Currently one of the most widely used battery chemistries in robotics is the Lithium Polymer (Li-po) battery. Therefore this will most likely be our battery of choice.

A major barrier to the implementation of the BASS system is the constraints of the power supply. In order for the micro controller to operate at sufficient clock speeds, a nominal supply of 5V must be obtained. In addition, most servo motors operate with a supply voltage of 5V. Since the micro controller is central to the BASS system and servo control is part of the deliverables, it is of paramount importance that a stable 5V supply is obtained. Also, some components such as the inertial measurement unit may require operating voltages at 3.3V.

The nominal cell voltage of a standard Li-po battery is 3.7V. This will not meet the power supply voltage requirements of either class of device mentioned above. This poses the requirement for the use of a switched mode power supply (SMPS) scheme. The SMPS power supply has the advantage of being highly efficient when compared to standard linear regulators. The SMPS will allow for the generation of the required supply voltage despite variances in the cell voltage of the Li-po battery. Even if the battery voltage dips, as long as it does not drop below a certain voltage the SMPS will be able to provide the required voltage.

The disadvantages associated with the SMPS in this project arise mainly from power supply noise that will be injected due to the high frequency switching action inherent in the nature of this kind of supply. This noise could corrupt sensor data due to variances in potentials between different analog conversions. The noise will also add the requirement of increased filtering, which will increase system complexity and expense.

An alternative method of achieving the same goal would be to place multiple batteries in series. Two li-po cells in series would result in a total voltage of approximately 7.4V. The increase in voltage due to the series connection would raise the voltage threshold above the drop out requirements for most linear regulators.

One of the advantages of this method is that the linear regulator is much simpler to implement than the SMPS, and will require less design. Also, the linear regulator will not generate nearly as much noise as the SMPS. Disadvantages arise in the fact that this mode of supply will suffer from reduced efficiency due to the dissipation of excess voltage in the form of heat. Also, if the battery voltage dips below the threshold dropout voltage, the regulator will fail to properly regulate, and failure issues are likely to arise.

Due to the needs identified, requirements were specified for an indoors environment in a generic maze structure. The environment will be a pre-specified maze. The maze walls will be uniform, smooth, and will have 90 degree angles. The minimum maze corridor width will be no less than 24 inches. The minimum maze wall height will be 18 inches.

The platform that will be used to demonstrate the sensor suite will have a differential drive. The platform will be no more than 17 inches in height. The sensor suite will be no more than 1kg in weight. The robot will nominally keep a distance of two to four inches from stationary obstacles.

A modern communications interface is absolutely necessary. Current computers are rapidly phasing out legacy interfaces such as the parallel port and rs-232 port. At the time of the writing of this document, the USB standard is the main player in communications interfaces. In order to take advantage of this standard while maintaining the simplicity of rs-232 serial communication, it is expected that the use of a virtual communications port will be used due to its easy availability and simplicity of use for the end user.

In the event that the BASS project is ahead of schedule, potential wireless communications interfaces will be explored. The 2.4 GHz band and the protocols that run off this spectrum are the best candidates due to no licensing requirements and many well developed protocols (ZigBee, Bluetooth, etc.)

Preliminary design

The overall architecture for this sensor suite is a three tiered architecture. Below is a diagram of the three tiers and what components are in which tier.

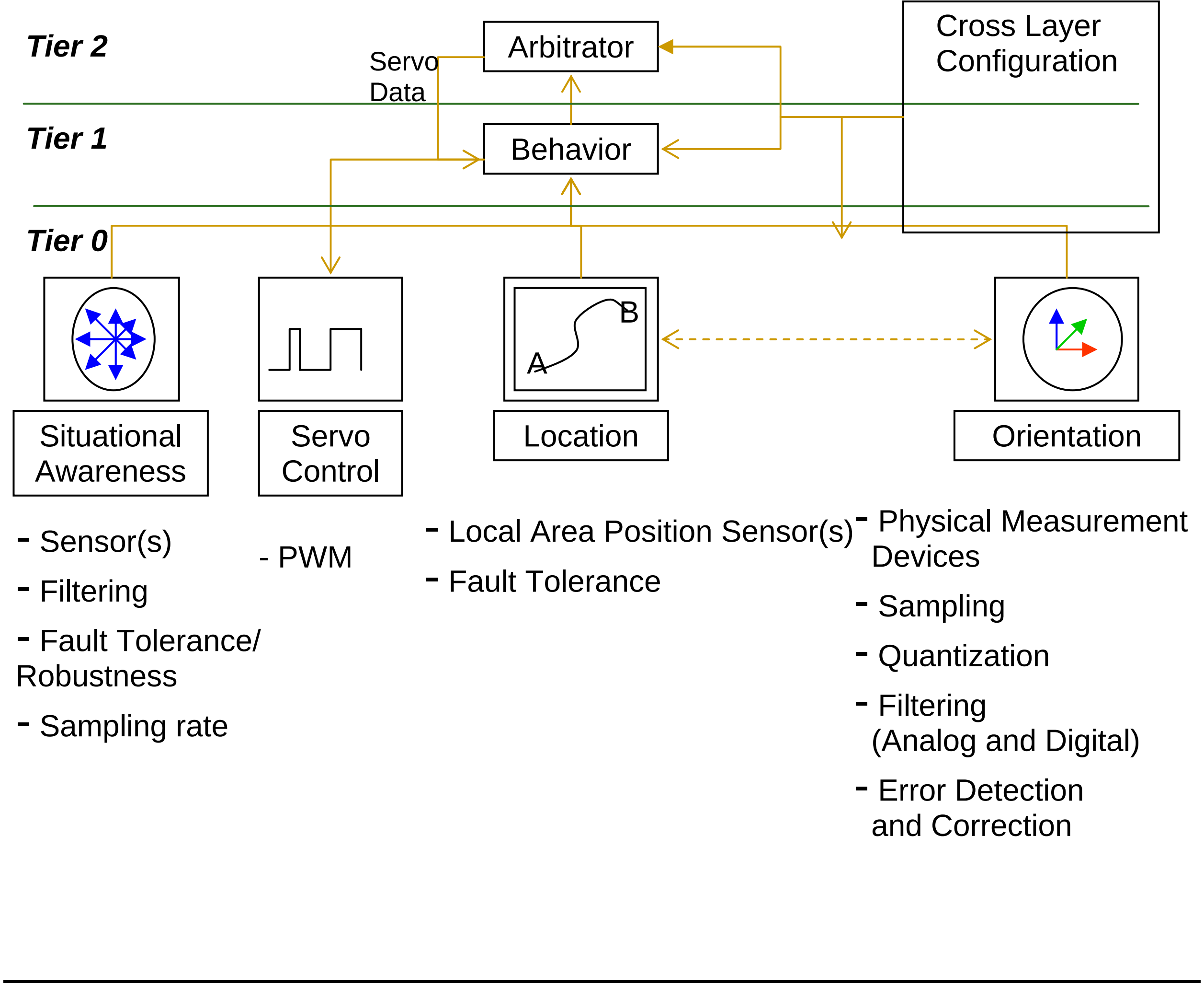


Figure 1: Overview of design

Functional Decomposition

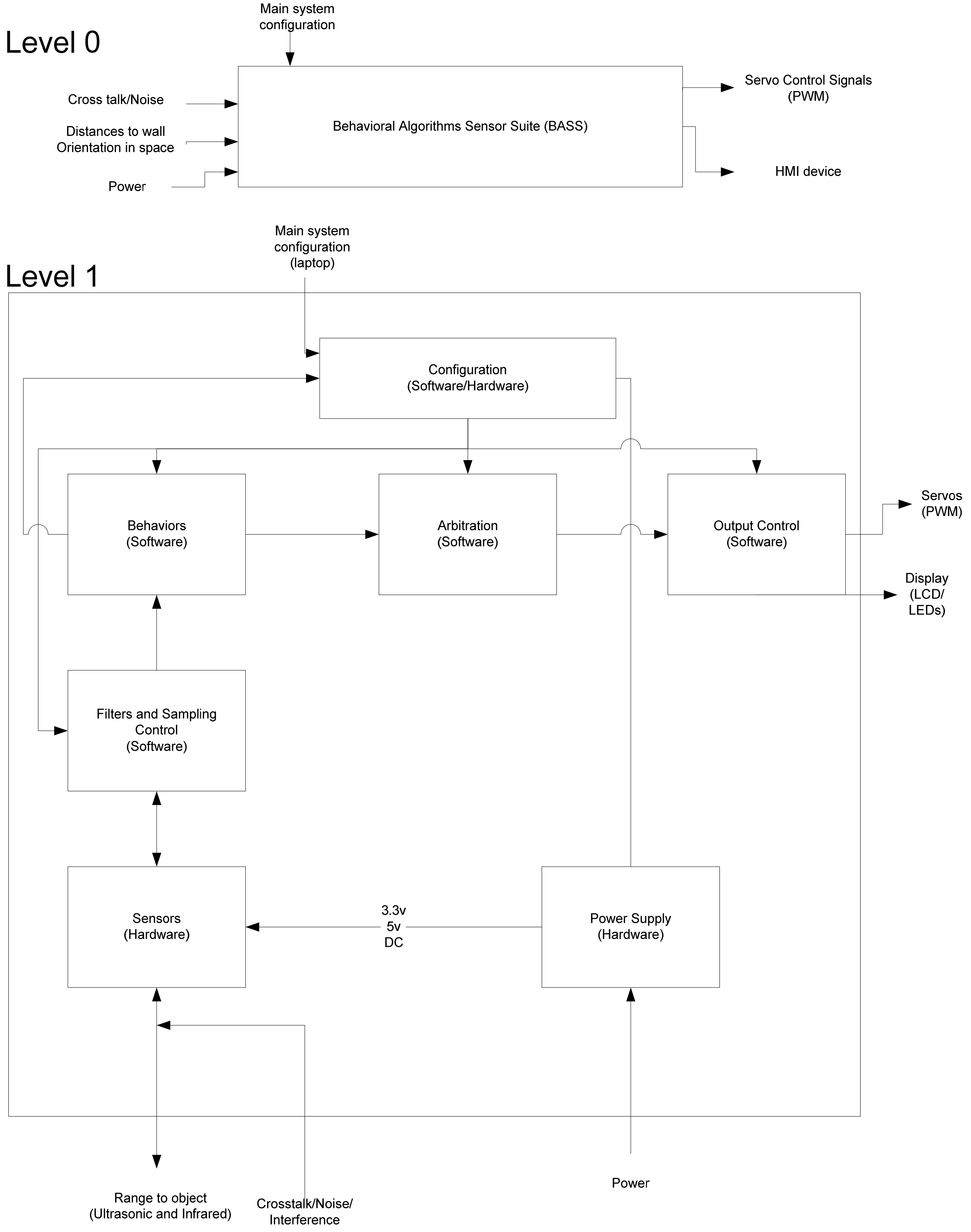


Figure 2: Functional Decomposition

Description of Level 1 Blocks

Sensors

- Inputs:
  - Sensor signals(IR, Ultrasonic, and Inertial measurement)
  - Control signals (fire, reset, and on/off) from microcontroller running Filters and Sampling Control block
- Outputs
  - Mix of analog and digital signal sent back to microcontroller
  - Sensor pings (IR, Ultrasonic)
- Responsibilities/Functions
  - All hardware required to sense ranging and position in space

Filters and Sampling Control

- Inputs:
  - Mix of analog and digital signals from the sensors
  - Configuration data (sampling rates and filter selection)
- Outputs
  - Control signals (Fire, reset, and on/off) to the sensor hardware
  - Smoothed interpreted data to the behaviors
- Responsibilities/Functions
  - Control the collection of data from the sensors
  - Smooth (filter) data so as to remove errors
  - Provide a configuration interface with the Configuration block

Behaviors

- Inputs:
  - Smoothed (filtered) data from the sensors
  - Configuration data (which behaviors are active)
- Outputs
  - The result of how each active behavior think the robot should respond
- Responsibilities/Functions
  - Determining how each behavior reacts, not which behavior the robot should follow.

Arbitration

- Inputs:
  - The output of each behavior (i.e. what that behavior thinks the robot should do)
  - Configuration data (set of rules for selecting which behavior or combination of behaviors should be followed)
- Outputs
  - The action that the robot will take
- Responsibilities/Functions
  - Determining the action the robot will take based on the outputs of the behaviors.

Output Control

- Inputs:
  - The action that the robot needs to take
  - Any information that needs to be displayed on the HMI display
- Outputs
  - Servo/motor control signals
  - HMI displays
- Responsibilities/Functions
  - Taking the action that the robot needs to take translating that action to servo and motor controls.
  - Displaying any data to the user via HMI display

Power Supply

- Inputs:
  - 5V or 12V DC power
- Outputs
  - Regulated 3.3V and 5V DC power
- Responsibilities/Functions
  - Providing power to the sensors, communication hardware, and output devices
  - Provide duel supplies (servos/motors and main system)

Configuration

- Inputs:
  - Communications interface with control system (PC)
  - Base configuration hard coding in the system
- Outputs
  - Configuration changes for all software blocks
    - Filters and Sampling Control
    - Behaviors
    - Arbitration
    - Output Control
- Responsibilities/Functions
  - Handle communications interface with control system
  - Provide all software module with the correct configurations

Preliminary Experimentation Plan and Evaluation Criteria:

The major focus of this project is the algorithms and instrumentation of the robot. In order to develop the algorithms, the sensor data must be characterized and the signal to noise (SNR) ratio must be estimated in order to determine which sensor will be used for which purposes. In addition, the sensor's reaction to the particular operating environment must be estimated.

Ultrasonic Sensor Testing:

One of the identified sensor possibilities in the budget range of this project are ultrasonic rangefinders. The ultrasonic rangefinders are linear in nature, but have the potential of cross-talk interference. In addition, they have a fairly wide sensing cone, and reflection dynamics.

The first task that must be completed with the ultrasonic rangefinders is to identify the range that they associate with corners. Due to the comparatively wide beam width of the ultrasonic rangefinders, it is possible for the corners to give many different readings, and even possibly introduce large amounts of cross-talk. As can be seen in figure 1, the wide cone that comprises the ultrasound sensor's line of sight has the possibility of returning a number of ranges. This behavior must be accounted for and statistical data on the matter needs to be compiled to understand the effect this will have on the filters.

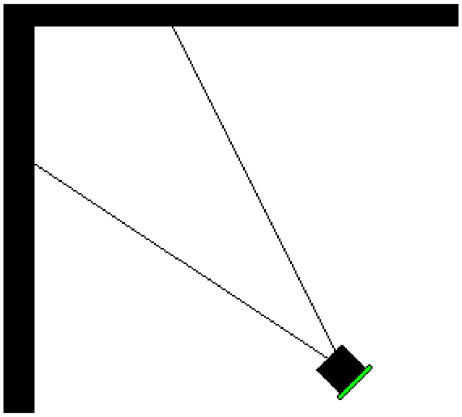


Figure 3: Ultrasonic sensor issue with large line of sight and corner

In addition, there will be a "dead band" in which the ultrasonic sensors will not return a linear range. This range must be determined so that it may be compensated for. Other issues include the distance of maximum range that the sensor can reliably range. Figure 2 demonstrates the problems associated with the dead band returning a false sense of security and the unreliable outer range of the sensor.

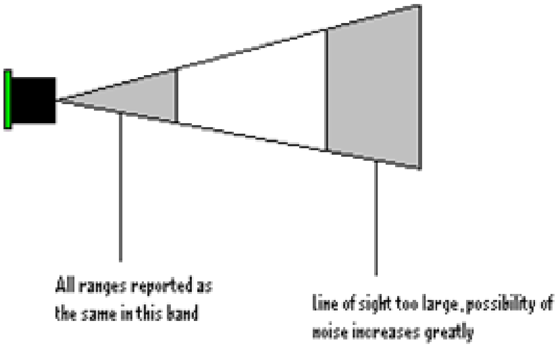


Figure 4: Ultrasonic sensor range areas

The characteristics of the noise present from the ultrasonic sensors must be determined so that a filter structure may be created to present accurate data for the algorithms. Also, the amount of cross talk from other sensors must be determined to see how much of a problem it is.

Infrared (IR) rangefinders are the second potential rangefinder identified that fit within the budget constraints. IR sensors may require analog sampling, which introduces sampling rate issues. IR sensors also are non-linear in nature, and this must be compensated for. In addition to the inherent non-linearity, other close range issues and sensor orientation parameters must be defined. Figure 3 demonstrates the severe non-linearity of the IR sensors.

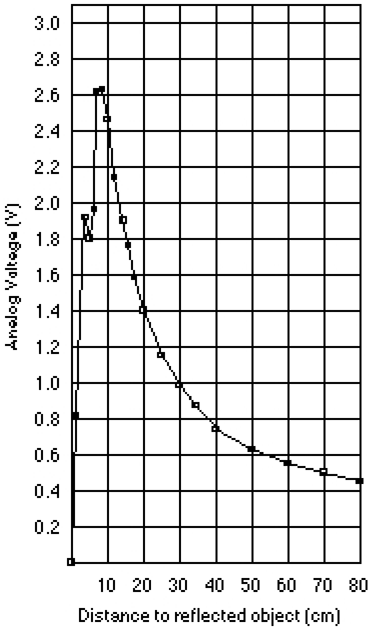


Figure 5: Non-linearity of IR sensors (GP2D12 Output Voltage to Distance Curve)

IR sensors have narrow beams as compared to ultrasonic rangefinders. Therefore it will be of importance to determine the IR sensor's parameters when ranging corners, as IR sensors might help to offset the issues created by the ultrasonic sensors. Also the distance-voltage characteristics of the IR rangefinders will need to be determined to calibrate and linearize the sensors.

IR sensors are less susceptible to cross talk as compared to ultrasonic rangefinders, but may have problems when they in the presence of other IR emitting sources. Therefore, an IR cross talk test must be performed to determine the IR sensor's reliability in the environment they will be operated in. The characteristics of the noise present on IR sensors must also be determined by testing in order to determine the filter structure that will be employed.

Device parameter tests must be run for both the IR and ultrasonic rangefinders in order to characterize the above identified issues. Therefore, the following tests will be performed: corner ranging characterization, dead-band range characterization, general ranging characterization, and cross talk characterization.

Testing must also be performed on the Inertial Measurement Unit (IMU.) The IMU will need accuracy and noise testing. In addition, drift and temperature offsets must be considered. The error and analog sampling characteristics of the device must be determined in order to evaluate the device's reliability.

One test will be to see the maximum change that the sensor can reliably report. Comparisons of the reported position and true position must be carried out in order to find out when the device must be reset. Also, the effect of quantization on the error must be measured in order to determine the required analog sampling device parameters and filtering structure.

Preliminary list/brief description of tasks and allocation of responsibilities

Hardware

Inertial Measurement Unit

- Members Responsible: Alex Behnaz and Brian Loop

- The IMU requires work in interfacing, output data stream management, and computing the data required by the host algorithm

Ultrasonic Range Finders:

- Members Responsible: Alex Behnaz and Brian Loop
- The ultrasonic rangefinders require digital bus logic development, filtering, and interrupt management

Infrared Range Finders:

- Members Responsible: Alex Behnaz and Sameer Dhawan
- The infrared sensors need digital or analog bus interfacing, and control logic development.

Bump Sensors:

- Members Responsible: Sameer Dhawan and Ehsan Foroudi
- The Bump sensors require need digital bus interfacing, and control logic development.

LCD

- Members responsible: Brian Loop
- Integrating an LCD for testing purposes and assisting in configuration.

Drive train

- Members responsible: Ehsan Foroudi, Brian Loop, and Sameer Dhawan
- The platform requires a differential drive system, including the motors, and control system.

Wheel Watchers (Encoders)

- Members Responsible: Brian Loop and Ehsan Foroudi
- The wheel watchers require microprocessor as well as hardware interfacing to measure velocity and direction.

Power

- Members Responsible: Alex Behnaz and Sameer Dhawan
- The power source requires an onboard battery and internal regulation.

Algorithms

- Members Responsible: Sameer Dhawan and Brian Loop
- The algorithms need to account for wall detection, odometry, mapping, and overall goal of the robot and control systems.

Simulation

- Members Responsible: Brian Loop
- Software simulation requires the simulation of sensors, platform and the environment.

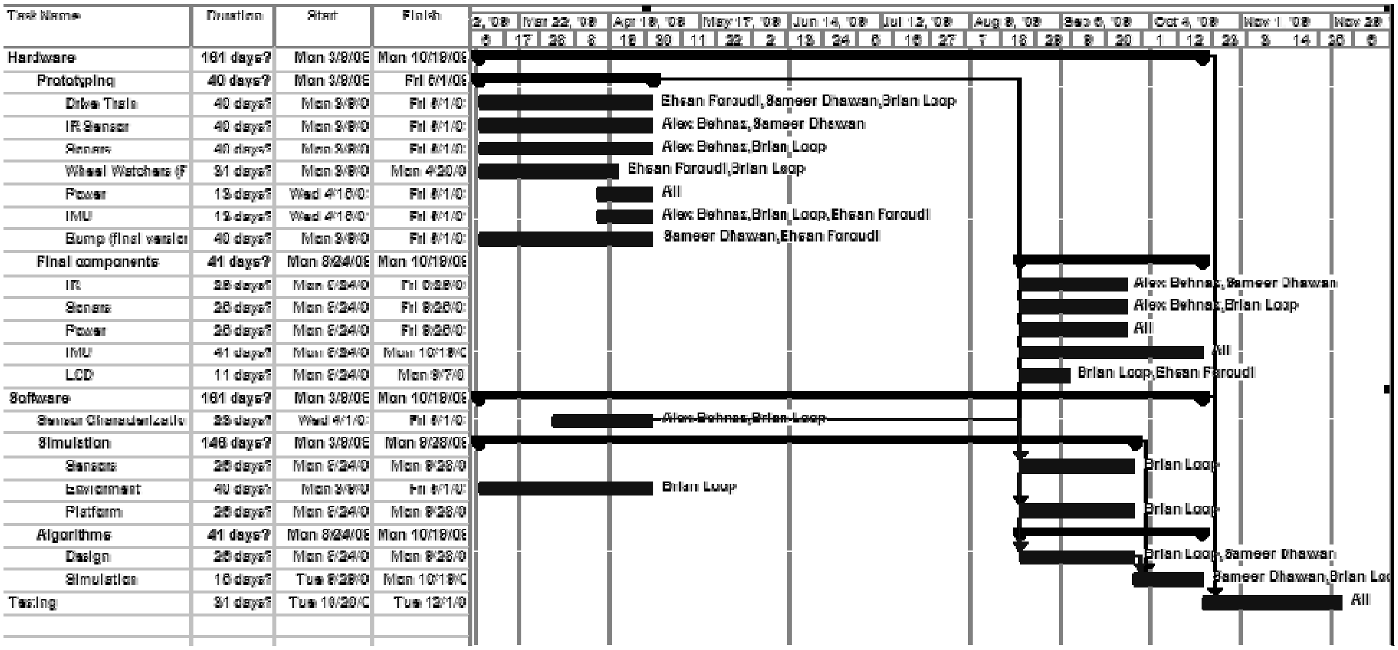
Sensor Characterization

- Members Responsible: Alex Behnaz
- Sensor characterization through empirical testing using LabVIEW.

Testing

- Members Responsible: Alex Behnaz, Brian Loop, Sameer Dhawan, and Ehsan Foroudi
- Overall System testing once hardware and software are finalized. Debugging and Fixing both the Hardware and the Software.

- **Task Schedule**



Task Name	Duration	Start	Finish
Hardware	161 days?	Mon 3/9/09	Mon 10/19/09
Prototyping	40 days?	Mon 3/9/09	Fri 5/1/09
Drive Train	40 days?	Mon 3/9/0	Fri 5/1/0
IR Sensor	40 days?	Mon 3/9/0	Fri 5/1/0
Sonars	40 days?	Mon 3/9/0	Fri 5/1/0
Wheel Watchers (F	31 days?	Mon 3/9/0	Mon 4/20/0
Power	13 days?	Wed 4/15/0	Fri 5/1/0
IMU	13 days?	Wed 4/15/0	Fri 5/1/0
Bump (final versio	40 days?	Mon 3/9/0	Fri 5/1/0
Final components	41 days?	Mon 8/24/09	Mon 10/19/09
IR	25 days?	Mon 8/24/0	Fri 9/25/0
Sonars	25 days?	Mon 8/24/0	Fri 9/25/0
Power	25 days?	Mon 8/24/0	Fri 9/25/0
IMU	41 days?	Mon 8/24/0	Mon 10/19/0
LCD	11 days?	Mon 8/24/0	Mon 9/7/0
Software	161 days?	Mon 3/9/09	Mon 10/19/09
Sensor Characterizatio	23 days?	Wed 4/1/0	Fri 5/1/0
Simulation	146 days?	Mon 3/9/09	Mon 9/28/09
Sensors	26 days?	Mon 8/24/0	Mon 9/28/0
Enviroment	40 days?	Mon 3/9/0	Fri 5/1/0
Platform	26 days?	Mon 8/24/0	Mon 9/28/0
Algorithms	41 days?	Mon 8/24/09	Mon 10/19/09
Design	26 days?	Mon 8/24/0	Mon 9/28/0
Simulation	15 days?	Tue 9/29/0	Mon 10/19/0
Testing	31 days?	Tue 10/20/0	Tue 12/1/0

Figure 6: Gantt Chart

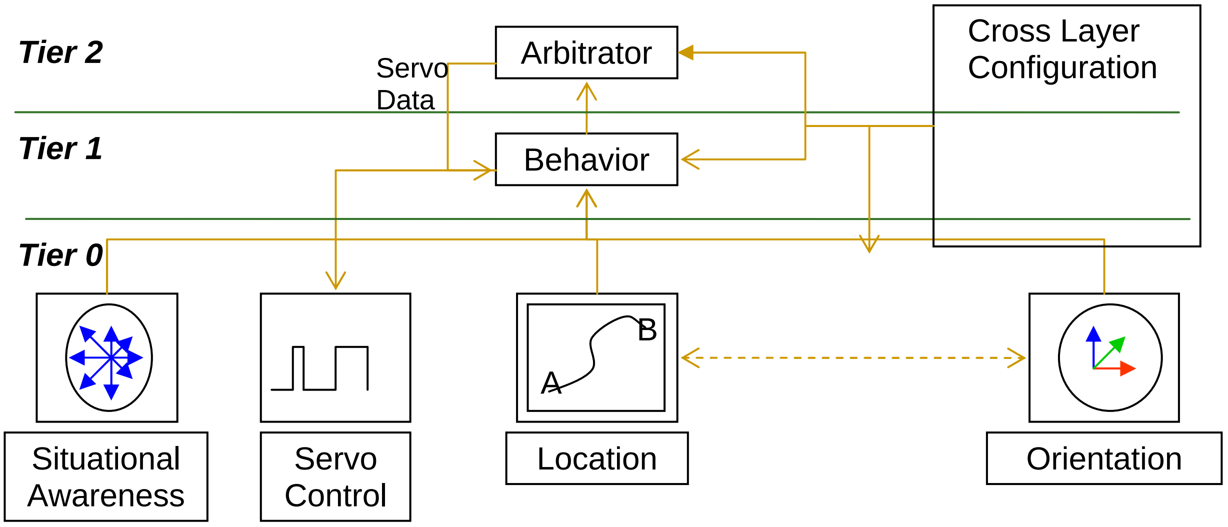
Appendix B Design Document

1. Functional Design/Architecture:

The function of the Behavioral Algorithm Sensor Suite (BASS) is to provide the orientation of the robot in physical space, the location of the robot with respect to its objective, and detect obstacles which impede the motion of the robot.

The BASS system will provide a modern, simple, and efficient interface to a host system. The interface to the host system will use a modern protocol, namely the USB Communications Device Class (CDC) protocol, with a possible expansion to a wireless protocol. This project will focus on an indoor environment.

The BASS system will provide built in control for a differential drive platform. The system will also act as a self-contained unit, with onboard power supply and regulation. In addition, it will provide built in behavioral algorithms for navigating a pre-determined maze structure.



- Sensor(s)
- Filtering
- Fault Tolerance/
Robustness
- Sampling rate

- PWM

- Local Area Position Sensor(s)
- Fault Tolerance

- Physical Measurement
Devices
- Sampling
- Quantization
- Filtering
(Analog and Digital)
- Error Detection
and Correction

2. System Design/Architecture:

Figure 1 demonstrates the level 0 functional decomposition of the BASS system. The inputs to the system are: power, ranges, orientation, and other sensor data. In addition, noise coupled to the readings will be an input into the system from the environment. A system configuration will be provided allowing the user to tailor the system's outputs to their specific needs. Outputs are to the human machine interface (HMI) and to the servo motors.

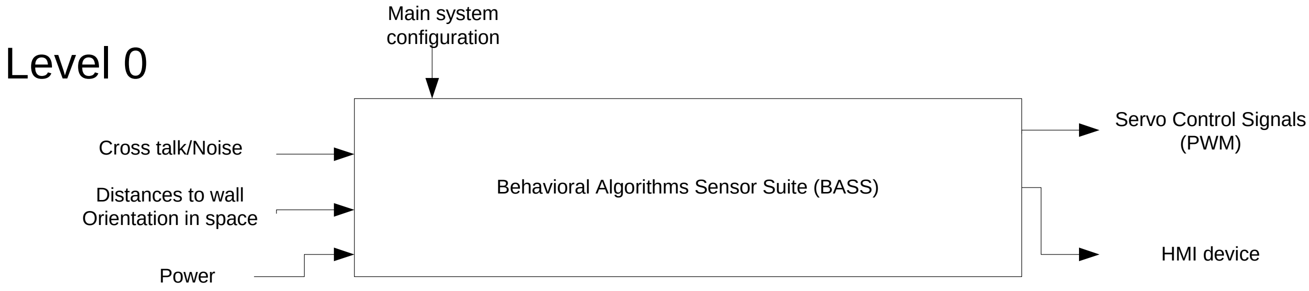


Figure 1: Functional Decomposition – Level 0

Figure 2 further breaks down the functional decomposition of the BASS system. The sensor block encompasses the ultrasonic rangefinder array, the IR rangefinder array, the IMU, and other peripheral sensors. The sensor block is implemented in hardware, and requires a power source. This power comes from the power supply block. The sensors will gather physical quantities, such as range and inertial changes along with the associated noise.

The next module is the filtering and sample control unit. This unit will be mainly implemented in the microcontroller. The filtering and sample control unit includes analog sampling routines to gather data from the raw sensor outputs. Information from the raw readings taken by the sensors will be digitally filtered. These filters will be implemented within the microcontroller software.

The filtered data is then given to the behaviors block. The behaviors block runs the behavior routines, and outputs each behavior's return value to the arbitrator. The arbitrator block takes the behavior outputs, and decides which of the behaviors should be translated into physical action.

The arbitrator mediates the behaviors, takes their information and translates it into a control signal to the drive train. This block translates the software defined actions into physical actions through the control of the motors and other actuators present on the robot. In addition, the output control will provide feedback to the user through a HMI device.

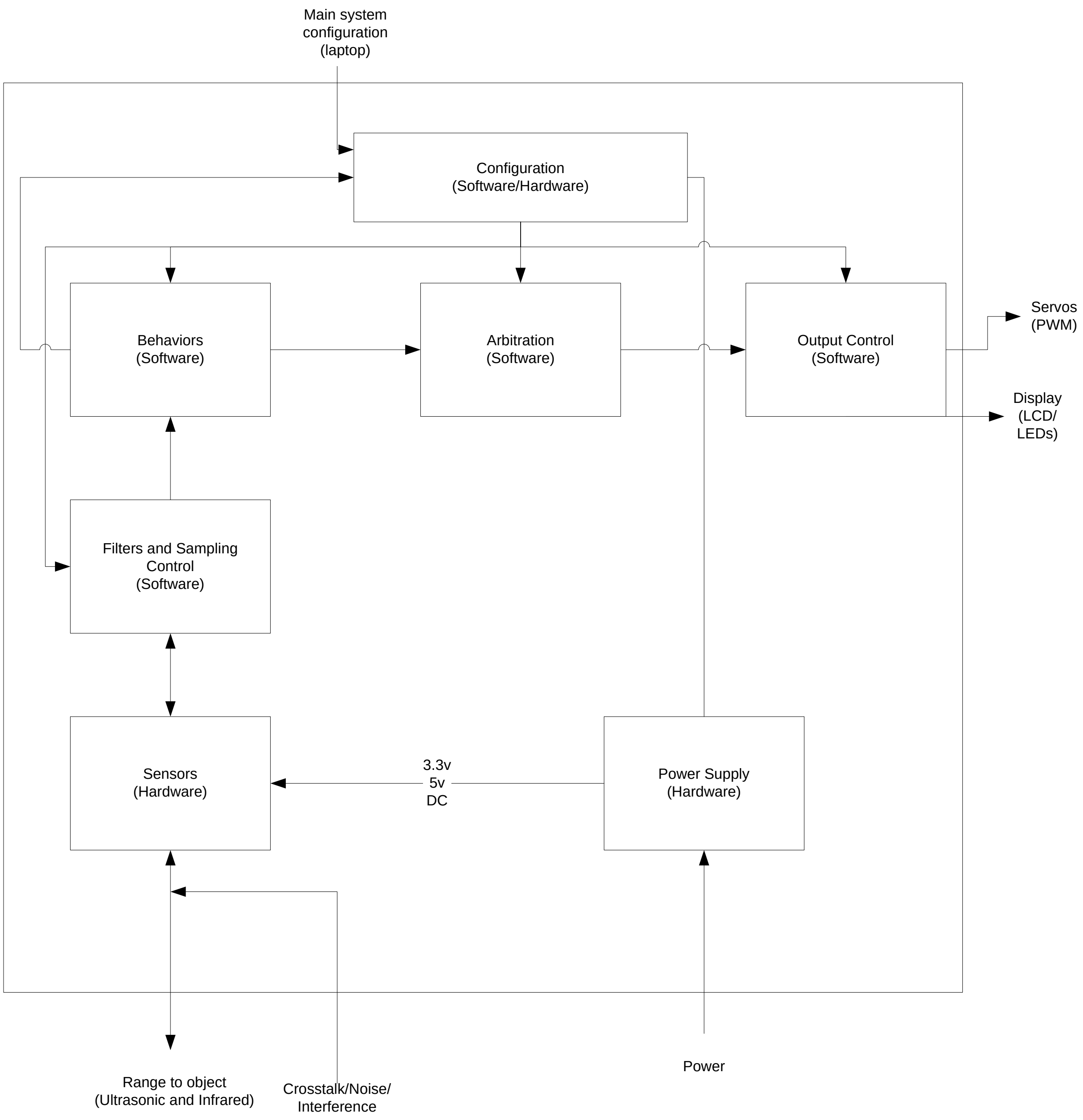


Figure 2: Functional Decomposition – Level 1

3. Detailed Design:

3.1. Overall design

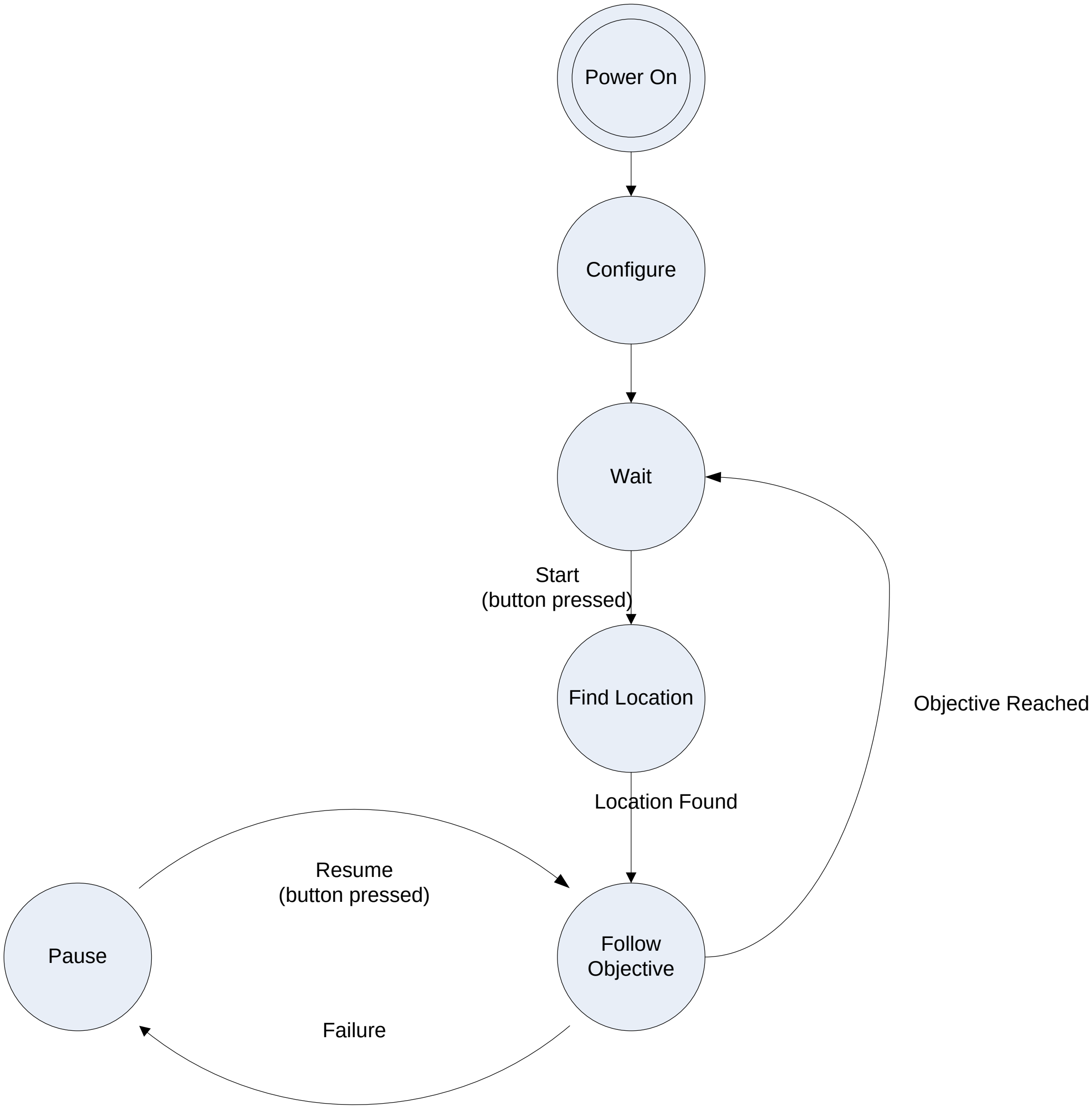


Figure 3: Overall System Behavior Flowchart

On power up the robot will enter the configure state. In this state the robot will load the sensor and filter parameters as well as configuring behaviors. When configuration is complete, the robot will enter the Wait state. The robot will not move or activate its sensors while in the Wait state. When a hardware button is pressed the robot will activate and enter the Find Location State. The action in this state is described by the following flowchart.

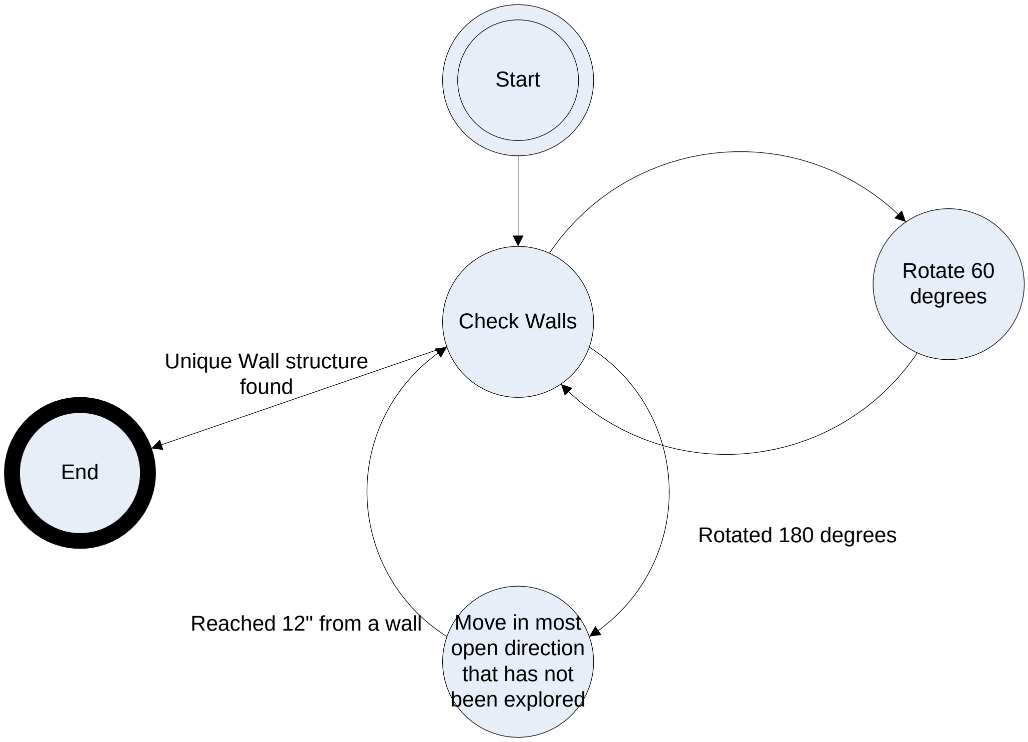


Figure 4: Find Location Process Flowchart

Since the layout of the maze is known ahead of time, the task of finding the robot's current location is reduced to finding a set of identifying walls. The algorithm will rotate 180 degrees in steps of 60 degrees. This will account for sensor blind spots. If no identifying walls are found the robot will move in the most open direction (determined by ranging data) until it reaches a wall. Then the robot will start the process again.

Once the robot has found its location, it will follow the map of the maze to reach its objective. If at any point during operation the robot encounters a poetically damaging situation, the robot will enter the pause state. In the pause state the robot will still have all the sensors and algorithms running, however the output to the wheels will be turned off. A physical button must be pressed in order to exit the pause state. After reaching the objective the robot will return to the wait state.

To provide the functionality as described in this statement of need, the following circuitry will be implemented.

3.2. Component Design

3.2.1. Ultrasound Sensors:

A ring comprised of ultrasonic sensors will be created. These sensors will be connected using their digital interface (see figure 5.) The range request control will be implemented through the use of a multiplexer (see fig 6.) In this manner, one pin will be used for firing, and three pins will be used for the select circuitry. This prevents waste of limited microcontroller I/O pins. To get the return from a specific ultrasound sensor, a de-multiplexer will be utilized (see figure 6.)

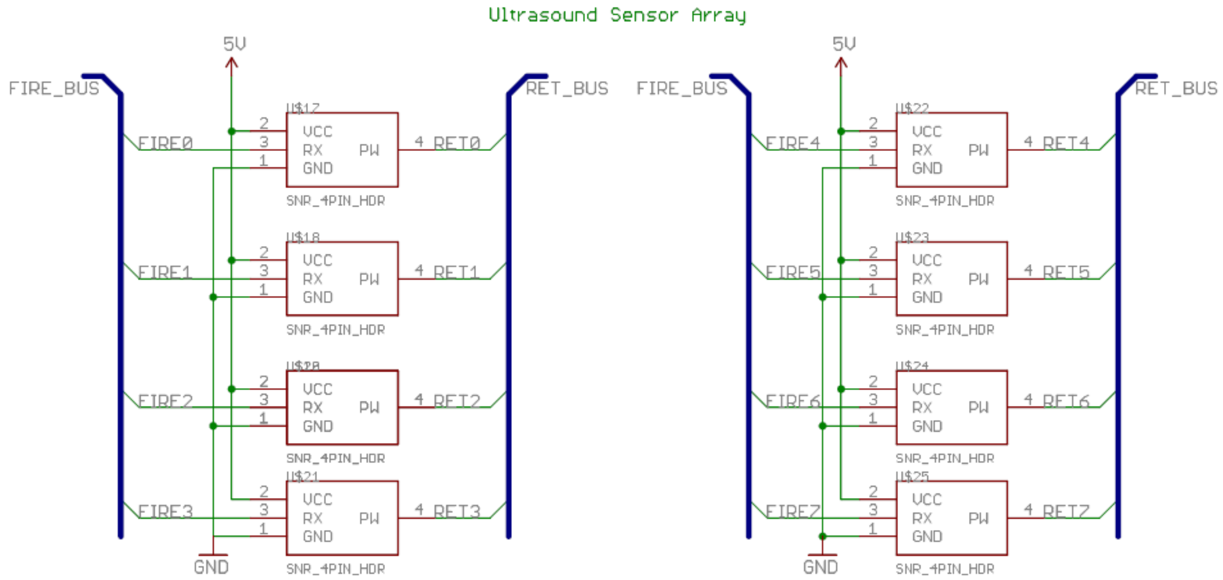


Figure 5: Ultrasound Fire and Return Circuitry

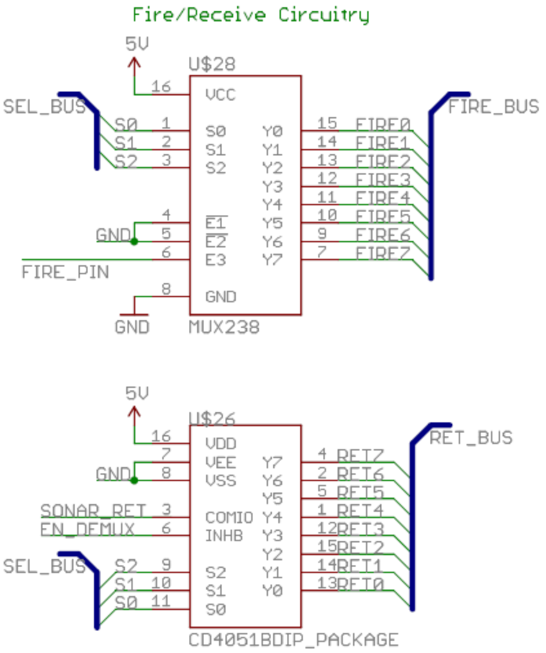


Figure 6: Ultrasound Receiver Circuitry

3.2.2. Infrared Rangefinders:

In addition to the sensing capabilities offered by the ultrasound sensors, another sensor ring will be created using IR sensors. These sensors are completely self contained and only need to be supplied power. The outputs of these rangefinders are in an analog format, and will be sampled by the microcontroller’s A/D converter (see figure 7.)

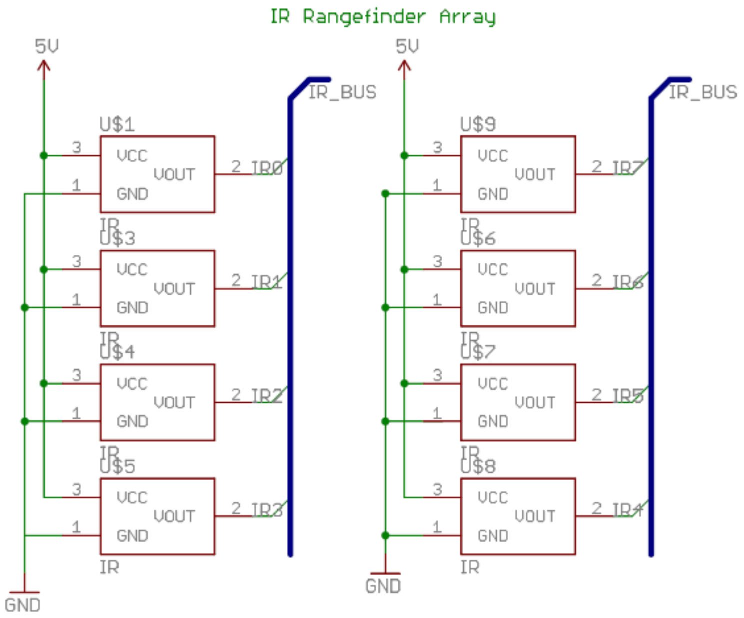


Figure 7: IR Sensors and Connections

3.2.3. Inertial Measurement Unit (IMU):

In order to use inertial data to aid in the algorithms, an IMU unit will be interfaced to the microcontroller. There are several gyroscopes and accelerometers that offer a completely digital (SPI) interface. These units will be attached to the microprocessor via the SPI bus (see figure 8.)

Figure 8: IMU Connections

3.2.4. USB Communications Circuitry:

One of the required needs identified in the problem statement is a modern communications interface. At the time of writing of this document, the Universal Serial Bus (USB) standard is a common protocol on modern equipment. In order to implement a USB slave device, certain hardware provisions are needed, and certain communications protocol must be followed. Since the scope of this communication is limited to the USB Communications Device Class (CDC), a UART to USB protocol converter chip will be used to implement this functionality. The connections to be made to this chip and the USB interface are shown in figure 9.

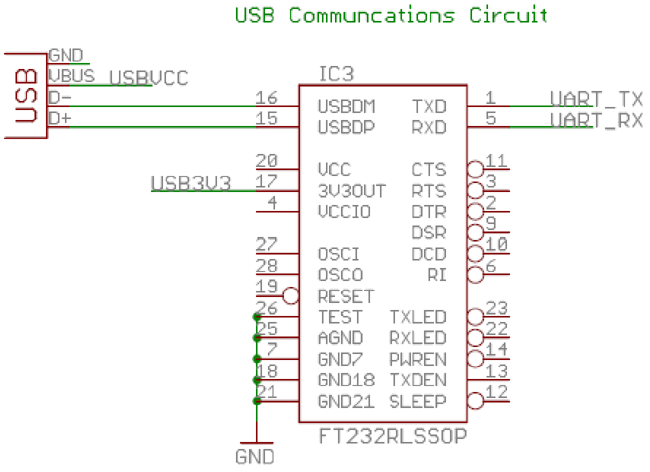


Figure 9: USB CDC Implementation Circuitry

3.2.5. Power Supply Circuitry:

To supply the required voltage and current to the circuit components, a power supply circuit will be created. Due to the specifications of the devices currently on the market and other restrictions such as operating frequency, both 3.3V and 5V need to be supplied. These voltages will be provided by linear regulators running off the battery. If need be, a switched mode power supply may be utilized to improve efficiency. The battery of choice will be a lithium polymer ion (li-po) battery pack. Also, to improve flexibility, the sensor suite will have a barrel jack style plug to allow users to plug in wall adapters or other power supplies. The details of the power supply circuit can be seen in figure 10.

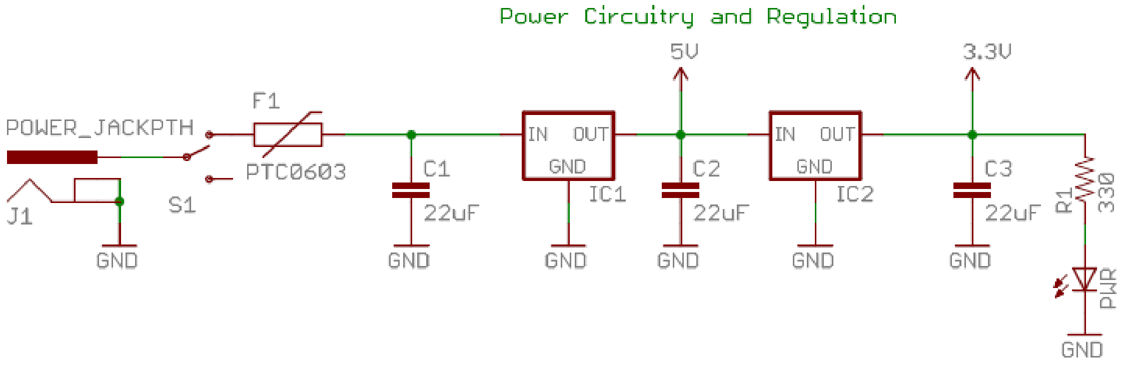


Figure 10: Power supply circuitry

3.2.6. Control, Filtering, and Data Processing:

In order to implement the control of the platform and provide the differential drive deliverable, one or more microcontrollers will be used. The design involves the use of an Atmel ATmega 168 microcontroller to deliver the motor control, sensor filtering, and data processing algorithms. A microcontroller will also hold the algorithms used to control the behaviors that the BASS system seeks to provide. Code size and/or speed constrains require another microcontroller may be added. Figure 11 below demonstrates the connections to be made to the processor.

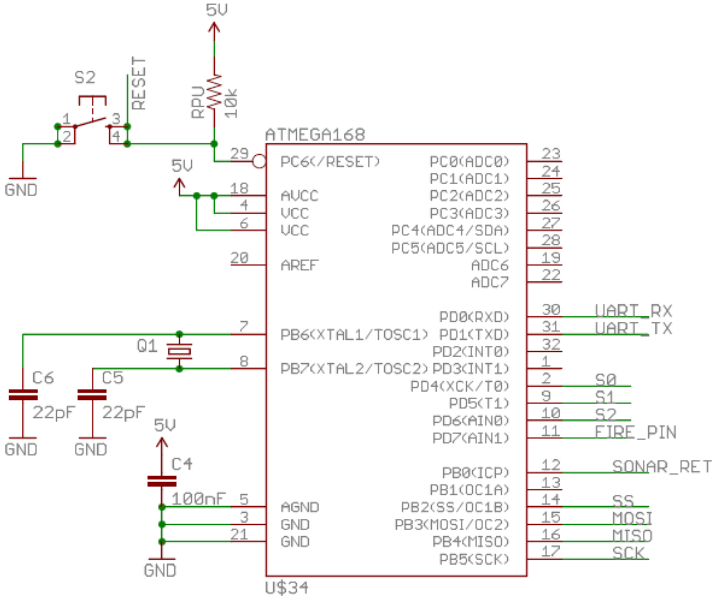


Figure 11: Atmel ATmega168 Microcontroller

For algorithm processing there will be an additional MCF51JM128 ColdFire micro-controller. Figure 12 shows the ColdFire connections.

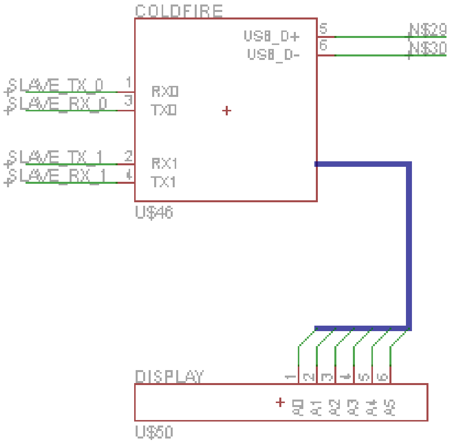


Figure 12: MCF51JM128 ColdFire Microcontroller

3.2.7. **Wheel Encoders:**

The wheel encoders output a quadrature output which is difficult to read and process directly from a microcontroller. To help simplify the processing of the encoder data, we will use a LS7266R1 quadrature counter. This chip will perform the processing and output the resulting count of wheel ticks to the microcontroller. Figure 13 demonstrates the connections from the LS7266R1 to the wheel encoders and microcontroller bus.

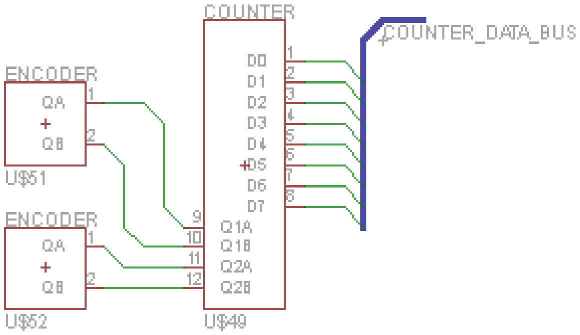


Figure 13: Quadrature Encoder Circuit

3.2.8. Motors:

To provide for the implementation of the differential drive, two continuous rotation servo motors will be employed. These servo motors are standard pulse driven continuous rotation servos. The motors are COTS parts, and are simply connected to the microprocessor on a given pin.

3.2.9. Bump Sensors:

In order to ensure that a given platform does not damage itself or its surroundings, a bump sensor array will be implemented. The bump sensor array will provide the platform knowledge of collisions, allowing the platform to cease the offending action.

4. Prototyping Progress Report:

Platform:

Our first step was to construct the demonstration platform from legos. The platform was designed with a gearing system that employed slip gears to help prevent catastrophic motor damage. However, the slip gear’s friction and the motor’s velocities were poorly matched which prevented the platform from moving smoothly. This proved to the team that the initial concept of a PID controller was a requirement.

Wheel Encoders:

Two quadrature wheel encoders (AMT 103-V) were obtained, and their characteristics were observed. A wheel encoder interface chip (LS7266R1) was acquired and interfacing to the microprocessor was completed.

The counter chip's parallel data bus was interfaced to the Atmel microcontroller. Testing was done at the lowest resolution of 48 counts per revolution.

PID Control:

The PID controller algorithm was prototyped and implemented via the Arduino platform. The PID controller implementation was then tested and shown to work in simulation.

Rotational velocity information was used from the wheel encoders to provide the input signal to the PID controller. However, issues arose from the fact that the servo control signals had very limited resolution, and therefore mapped poorly to the wheel encoder velocity information. This caused the precision of the PID control to be degraded.

In order to fix earlier problems, the slip gears were removed as they were poorly matched and caused slippage when even simple motions such as turning were performed.

Infrared Sensors:

Both a digital and analog version of the Sharp IR sensors were obtained. The digital IRs were preferred for their reduced pin requirements. However, these sensors have been discontinued by the manufacturer, and therefore are no longer a usable item. Based on this lack of digital IR sensors, the decision was made to switch to analog IR sensors. The analog sensors were tested, and seen to draw about 27 mA of current each.

Ultrasound Sensors:

A preliminary design for the ultrasonic sensor ring was completed. The driving code and the interface combinatorial logic were created and tested on a breadboard. Some initial sensor characterization was performed, mainly to gauge the expected accuracy to the sensors, and to get an idea of their noise profile. The ultrasound sensors were tested for current consumption, and found to have an average of 2mA of current draw.

Power:

Li-po batteries were chosen as the mobile power supply of choice. In order to comply with the specialized charging requirements, the Sparkfun "Fast Charger" was selected to provide for battery charging and will be obtained shortly.

Due to the lack of important specifications provided on the original charger, it was decided that a different, more reliable and documented battery charger would be bought. The "Single Cell Lipoly Chrager" was ordered via the internet.

Caster Wheel:

The original caster wheel had performed poorly and needed fixing. A few design changes were implemented, namely the reduction of the wheel thickness and caster placement which improved the end performance of the caster wheel.

Problems and Surprises Encountered:

The first problem we encountered was the fact that the digital IR sensors that we had been counting on were not available for purchase anymore. In addition, we ran out of available pins on the microprocessor very quickly, leading to a need to move to a larger microprocessor. In terms of the algorithms, it was seen that a need for more memory was a pressing force. The microprocessors available just did not have the required amount of memory onboard.

5. Final Experimentation Plan and Evaluation Criteria:

The BASS system is comprised heavily of both hardware and software. Experimentation will occur on both the hardware components and the software that controls the system. This will be accomplished by segmenting the platform into its functional blocks and testing each block with respect to its own functionality and its interaction with the entire system.

5.1. Hardware

Test	Description	Criteria
Motion Test	This test will verify that the BASS system is capable of driving servos in a differential drive configuration.	BASS system must drive two servos in both forward, backward, stop, and turn movements.
Control Test	This test verifies the capability of the BASS system to provide controlled driving.	PID controllers will run both servo motors such that the platform will move in its intended direction with minimum error (i.e. if the system wishes to go straight ahead, the PID controllers will adjust for errors in the servo motor matching and provide for correct forward motion.)
Odometry Test (Encoders)	The BASS system needs to provide an odometry system to provide both distance measurements and rate feedback for the PID control.	Encoder system will provide a count of wheel movement verified by physical measurements.
Collision Test	The system must be able to detect a collision with a solid object	The system will detect when a collision has occurred with an object
Sound Based Range Finding Test	The ultrasonic sensors must be able to provide mid-long distance range finding capabilities for the system.	ultrasound sensors will be able to detect an obstacle between 7 - 24 inches.

Infrared Range finding Test	The IR rangefinders must be able to provide short to midrange distance sensing.	The IR sensors must be able to detect an obstacle between 0 - 7 inches (0 inches being taken from the outside perimeter of the platform.)
Inertial Measurement Test	The IMU must be shown to provide information about the change in the robot's position	The IMU will demonstrate the ability to detect changes in the movement of the platform and correlate those measurements with physical units (i.e. robot has turned counter clockwise, approximately 45 degrees.)
Communications System Test	The BASS needs to provide a modern communications interface.	The BASS system will be able to implement a USB CDC slave device to communicate with a host system.

5.2. Software

Test	Description	Criteria
Local Communications Test	To allow for the proper operation of the system, the software must provide for the ability to communicate with sub-modules.	The software will demonstrate the ability to facilitate local communications with sub modules (i.e. interacting with range finding and drive train sub modules)
Behaviors Test	The BASS system must provide different behaviors for the artificial intelligence routines (such as obstacle avoidance, goal finding, etc.)	Each behavioral algorithm is to be tested individually and proved to be operating with the expected behavior. The behaviors will not interfere with one another.
Arbitration Test	Due the conflicting nature of the behaviors implemented, the software must provide for an arbitration scheme to control which behavior is expressed.	The arbitrator will be able to control and select among the behaviors such that the proper behavior is selected for different situations. The arbitrator must be able to control which behavior is implemented at a given time, and by how much.

Goal Accomplishment Test	The system must prove its ability to solve the given task. For the demonstration purposes of this project, the maze extraction point scenario must be solved by the BASS system.	The BASS system will reach its end goal if it is capable of doing so (not blocked on all sides, or otherwise "unfairly" prevented from completing its goal.) The BASS system must be able of repeatedly accomplishing its goal. The BASS system must be capable of successfully completing its mission, even in the presence of some obstacles

6. Final list/brief description of tasks and allocation of responsibilities:

Inertial Measurement Unit
- Members Responsible: Sameer Dhawan and Ehsan Foroudi
- The IMU requires work in interfacing, output data stream management, and computing the data required by the host algorithm

Ultrasonic Range Finders
- Members Responsible: Alex Behnaz
- The ultrasonic rangefinders require digital bus logic development, filtering, and interrupt management

Infrared Range Finders
- Members Responsible: Alex Behnaz
- The infrared sensors need digital or analog bus interfacing, and control logic development.

Bump Sensors
- Members Responsible: Sameer Dhawan and Ehsan Foroudi
- The Bump sensors require need digital bus interfacing, and control logic development.

LCD
- Members responsible: Brian Loop
- Integrating an LCD for testing purposes and assisting in configuration.

Drive train
- Members responsible: Ehsan Foroudi
- The platform requires a differential drive system, including the motors, and control system.

Wheel Watchers (Encoders)
- Members Responsible: Brian Loop and Ehsan Foroudi

- The wheel watchers require microprocessor as well as hardware interfacing to measure velocity and direction.

Power

- Members Responsible: Ehsan Foroudi and Sameer Dhawan
- The power source requires an onboard battery and internal regulation.

Algorithms

- Members Responsible: Sameer Dhawan and Brian Loop
- The algorithms need to account for wall detection, odometry, mapping, and overall goal of the robot and control systems.

Simulation

- Members Responsible: Brian Loop
- Software simulation requires the simulation of sensors, platform and the environment.

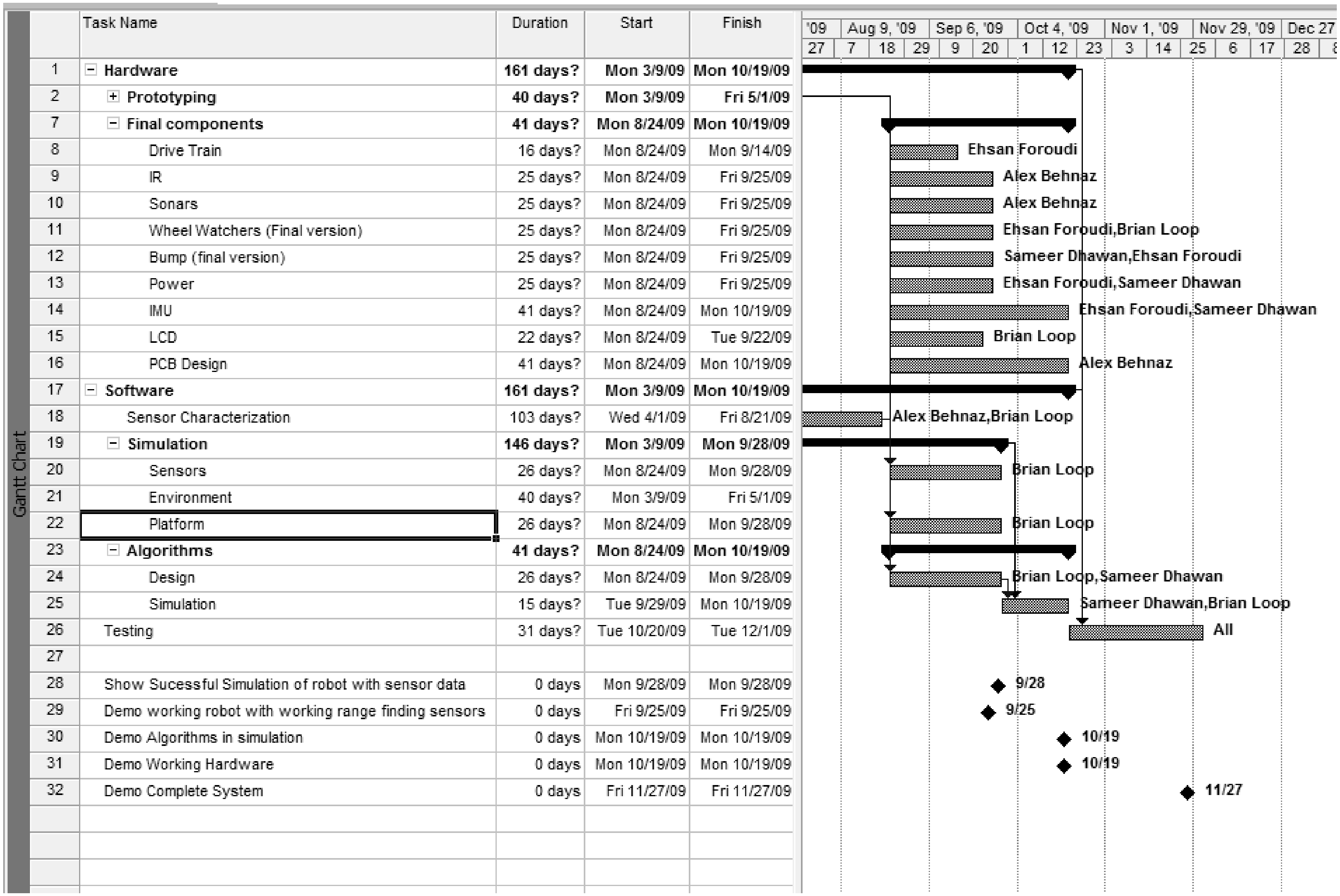
Sensor Characterization

- Members Responsible: Alex Behnaz, Brian Loop
- Sensor characterization through empirical testing using LabVIEW.

Testing

- Members Responsible: Alex Behnaz, Brian Loop, Sameer Dhawan, and Ehsan Foroudi
- Overall System testing once hardware and software are finalized. Debugging and Fixing both the Hardware and the Software.

7. Final schedule and milestones:



	Task Name	Duration	Start	Finish
1	**Hardware**	**161 days?**	**Mon 3/9/09**	**Mon 10/19/09**
2	**Prototyping**	**40 days?**	**Mon 3/9/09**	**Fri 5/1/09**
7	**Final components**	**41 days?**	**Mon 8/24/09**	**Mon 10/19/09**
8	Drive Train	16 days?	Mon 8/24/09	Mon 9/14/09
9	IR	25 days?	Mon 8/24/09	Fri 9/25/09
10	Sonars	25 days?	Mon 8/24/09	Fri 9/25/09
11	Wheel Watchers (Final version)	25 days?	Mon 8/24/09	Fri 9/25/09
12	Bump (final version)	25 days?	Mon 8/24/09	Fri 9/25/09
13	Power	25 days?	Mon 8/24/09	Fri 9/25/09
14	IMU	41 days?	Mon 8/24/09	Mon 10/19/09
15	LCD	22 days?	Mon 8/24/09	Tue 9/22/09
16	PCB Design	41 days?	Mon 8/24/09	Mon 10/19/09
17	**Software**	**161 days?**	**Mon 3/9/09**	**Mon 10/19/09**
18	Sensor Characterization	103 days?	Wed 4/1/09	Fri 8/21/09
19	**Simulation**	**146 days?**	**Mon 3/9/09**	**Mon 9/28/09**
20	Sensors	26 days?	Mon 8/24/09	Mon 9/28/09
21	Environment	40 days?	Mon 3/9/09	Fri 5/1/09
22	Platform	26 days?	Mon 8/24/09	Mon 9/28/09
23	**Algorithms**	**41 days?**	**Mon 8/24/09**	**Mon 10/19/09**
24	Design	26 days?	Mon 8/24/09	Mon 9/28/09
25	Simulation	15 days?	Tue 9/29/09	Mon 10/19/09
26	Testing	31 days?	Tue 10/20/09	Tue 12/1/09
27				
28	Show Sucessful Simulation of robot with sensor data	0 days	Mon 9/28/09	Mon 9/28/09
29	Demo working robot with working range finding sensors	0 days	Fri 9/25/09	Fri 9/25/09
30	Demo Algorithms in simulation	0 days	Mon 10/19/09	Mon 10/19/09
31	Demo Working Hardware	0 days	Mon 10/19/09	Mon 10/19/09
32	Demo Complete System	0 days	Fri 11/27/09	Fri 11/27/09

Figure 14: Final Schedule for Fall 2009