FireBug Creations
Simon Fraser University
8888 University Drive
Burnaby, BC
Canada V5A 1S6

March 11, 1999

Dr. Andrew Rawicz,
School of Engineering Science
Department of Applied Science

Dear Dr. Rawicz

The enclosed document, *FireBug Design Specifications*, outlines the design details of our 370 project. The FireBug will demonstrate the possibilities of machine assisted search and rescue operations, or indeed, any application in a hazardous environment. The FireBug's primary goal is to work in conjunction with the SFU Aerial Robotics Group airship to identify and, if possible, eliminate hazards in a simulated disaster situation.

The project goal for ENSC 370 is to have a prototype capable of tele-operation. Simple object avoidance algorithms may also be demonstrated. All features detailed in this report will be implemented for the International Aerial Robotics Competition in June 1999.

Demonstrable features for the 370 deadline on April 20 will include:
- Tele-operation of the Firebug in a simple obstacle course
- Simple object identification and avoidance
- Computer simulation

Thank you for your time and consideration. All of us in FireBug Creations are very excited about this project and its future possibilities. Many of us are considering the implications of our research as possible thesis projects and marketable products.

Sincerely,

Julie Delisle

Julie Delisle
Media Relations Prime,
FireBug Creations

Enclosure: ENSC 370 FireBug Functional Specifications

FireBug Design Specifications

FireBug Creations
Web – http://mars.ensc.sfu.ca
Email – firebug-arg@sfu.ca

Contacts:

Julie Delisle – jdelisle@sfu.ca
Greg Bolli – gbolli@sfu.ca
Ryan Sadler – rrsadler@sfu.ca
Neil Watkiss – nwatkiss@sfu.ca

Executive Summary

Currently search and rescue missions rely entirely on the expertise of skilled professionals. These trained men and women regularly put themselves at risk to neutralize hazardous situations. With the rapid increases in processing power and improvements in artificial intelligence, automating these complex tasks can now be realized.

Our team of enthusiastic third year engineering science students plans to develop an autonomous land vehicle to work in conjunction with an aerial robot to identify and neutralize hazards, specifically fire, in a simulated disaster area. This land vehicle has been dubbed the FireBug.

The FireBug will compete in the International Aerial Robotics Competition in June 1999 as a member of Simon Fraser University's team of autonomous robots. The preliminary stage of the FireBug project will be completed by April 1999, and will deliver a prototype capable of remote tele-operation through an external base station.

A successful mission will demonstrate the feasibility of automating aspects of real search and rescue missions, saving lives and cost.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

Traditionally, trained professionals perform search and rescue operations. Each year numerous search and rescue personnel are injured or killed while attempting to control hazardous situations.

The International Aerial Robotics Competition (IARC) has challenged universities to develop autonomous teams of land/air robots to perform an unmanned search and rescue mission. The FireBug will complement Simon Fraser University's team of aerial robots.

This document outlines the design specifications for the FireBug. Due to the enormity of the FireBug system, FireBug Creations has approached this project from a systems perspective. This approach entails purchasing, rather than building, the individual system components. As a result, only those components designed by FireBug Creations have schematic diagrams included in this document. The audience for this document is Mr. Steve Whitmore, Dr. Andrew Rawicz, FireBug Creations, and miscellaneous Teacher's Assistants.

2.0 System Overview

The ARG at SFU requires a ground vehicle to work in conjunction with its Airship. The FireBug is therefore only one component of the overall competition strategy. It will be expected to receive commands and then execute them, as well as communicate its current status. The interactions between the various components of the mission are shown in Figure 1.

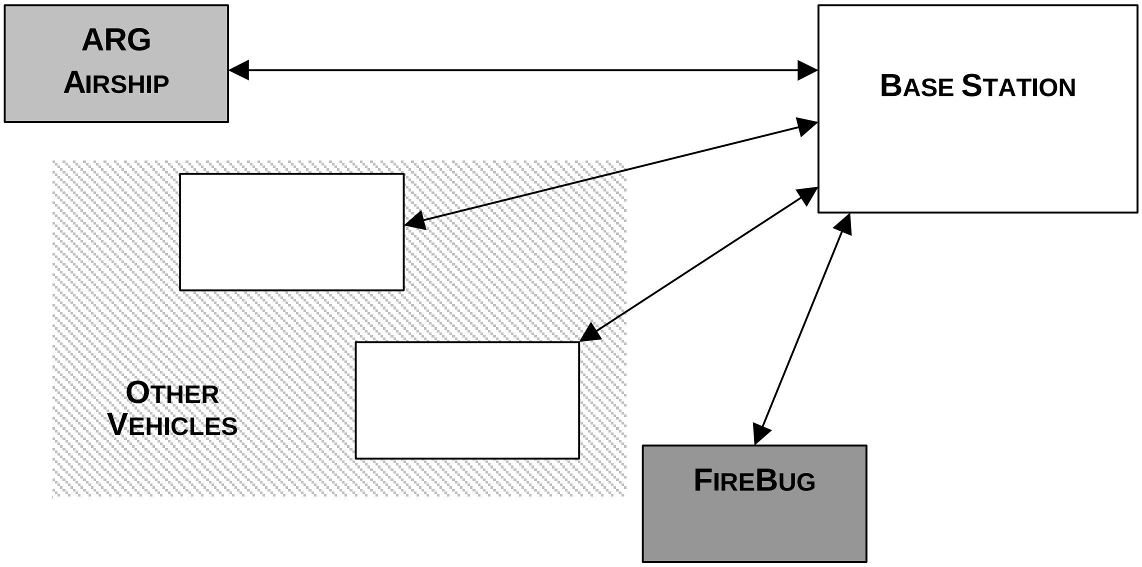

Figure 1: System Design Overview

The Airship is required to autonomously fly around the disaster zone, taking pictures of the ground. A Base Station will process these images, then issue commands to the FireBug. The FireBug will perform the following tasks:

- Navigate to a specific GPS position
- Relay video images of unknown or interesting objects to the Base Station
- Seek out and eliminate fire hazards

FireBug Creations has adopted a modular system design approach. This design strategy allows new features to be added easily, simplifying system upgrades and modifications. Several additional upgrades are under investigation for the millennial event in 2000.

3.0 Vehicle Specifictions

The interaction between system blocks is outlined in Figure 2. The sections following describe the operations of each system block in detail.

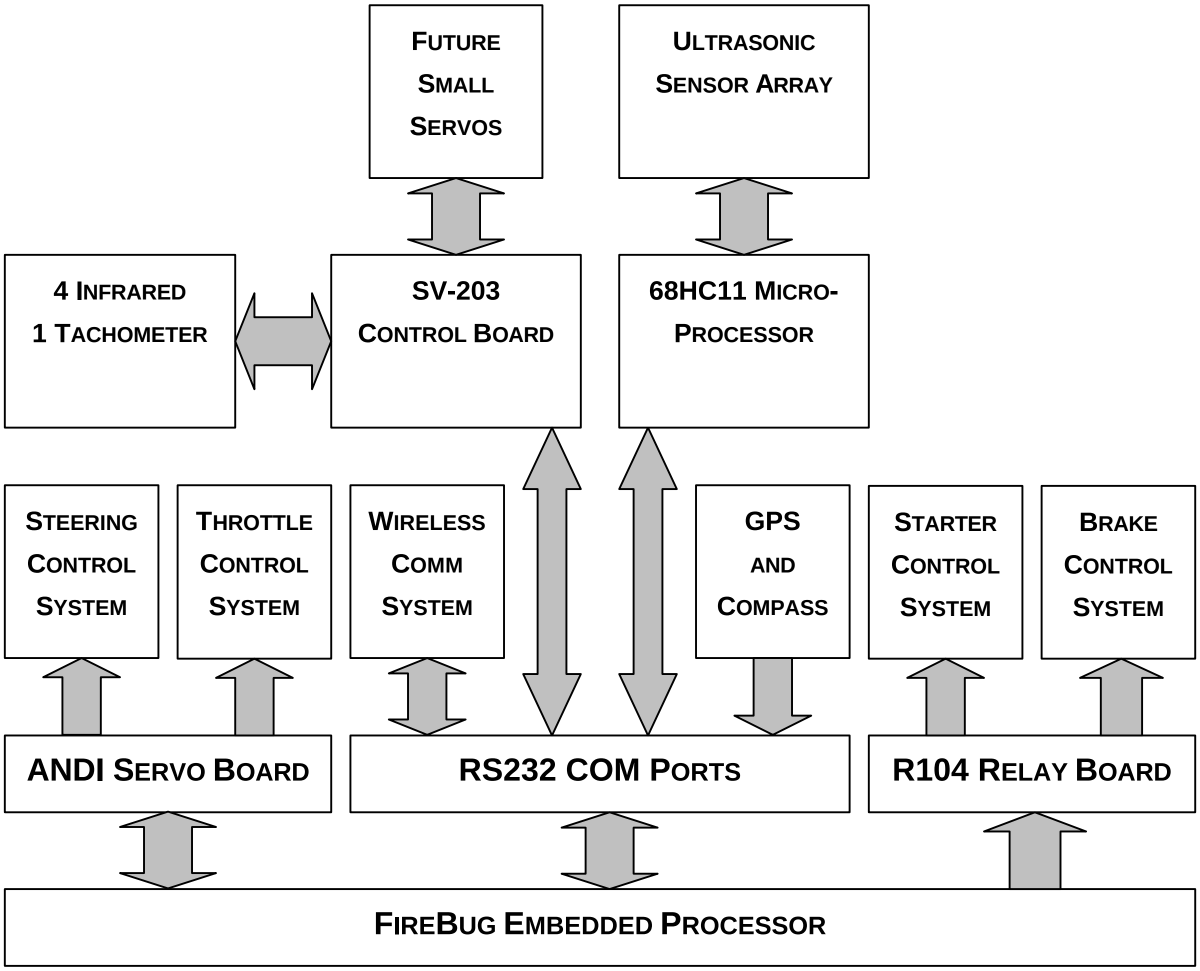

Figure 2: Vehicle Control System Block Overview

3.1 Physical Control System

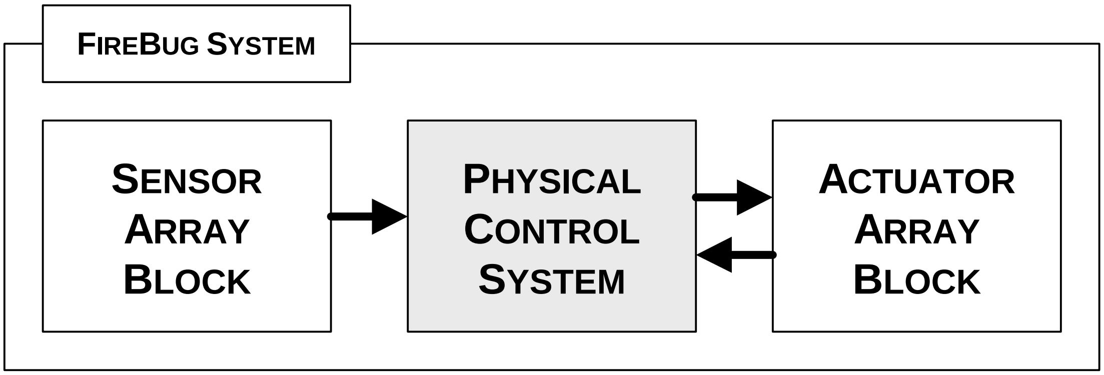

Figure 3: Vehicle Control System Context

3.1.1 Throttle Control System

The controller software sets the vehicle speed. In addition, an external PID controller improves the throttle system response and steady-state error. Figure 4 is a block diagram showing the feedback loop for the control system.

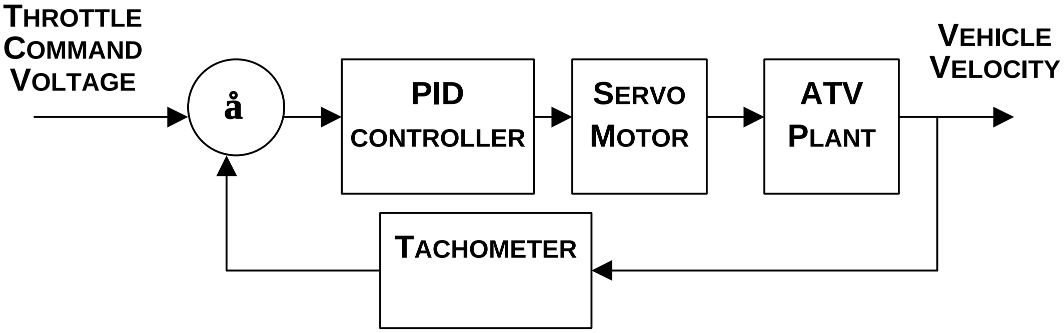

Figure 4: Throttle Control System Flow Diagram

The maximum vehicle speed is 10km/h, although the maximum speed may be adjusted after experimentation. Maximum speed is determined according to the following factors

- time to stop the vehicle from maximum speed
- turning radius achievable at maximum speed

PID control of the throttle servomotor is accomplished using the ANDI-Servo Control Board. See Section 3.1.4.1 for more information.

3.1.2 Steering Control System

The control software for the Steering Control System sets the position of the wheels. In addition, an external PID controller improves steering system response and steady-state error in heading. Figure 5 is a block diagram showing the feedback loop for the control system.

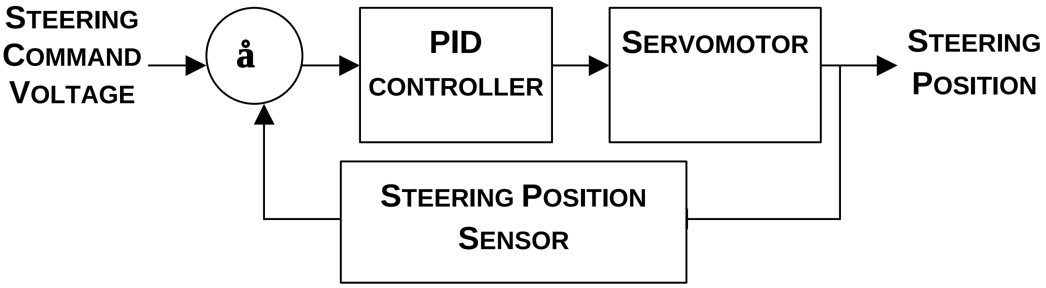

Figure 5: Steering Control System Flow Diagram

Speed will not need to be reduced during sharp turns because of the speed limit placed on the vehicle. The speed limitations put in place will safely allow full speed turning for the vehicle.

PID control of the steering servomotor is accomplished using the ANDI-Servo Control Board. See Section 3.1.4.1 for more information.

3.1.3 Braking Control System

The controller software will use the vehicle's Braking Control System in two situations:

- Rapid Deceleration
- Parking Condition

Non-critical deceleration is achieved by decreasing the vehicle speed setpoint, allowing the vehicle to slow down.

To control the degree to which the vehicle brakes, both the forward and rear brakes are connected to solenoids powered by independent relays. The tension on both of these brakes can be calibrated based on field trials.

3.1.4 Control System Hardware Components

3.1.4.1 ANDI-Servo Control Board

The ANDI-Servo Control Board is shown in Figure 6. It will control both the high torque servomotor in the steering mechanism, and the lower torque throttle control servomotor. The board is underrated for the current requirements of the steering servomotor, so an external H-Bridge circuit, Figure 7, will be constructed using SMP60N03 MOSFETs from Siliconix.

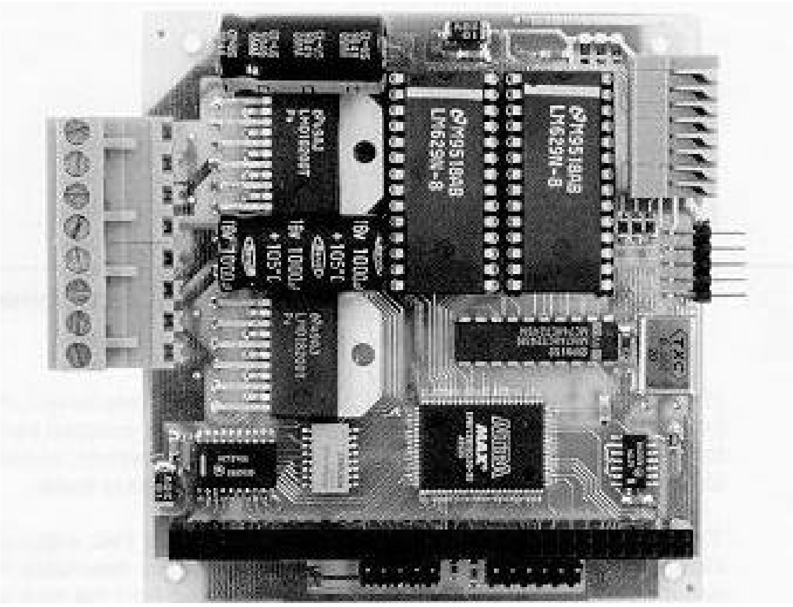

Figure 6: ANDI-Servo Controller Board

The specifications for the ANDI-Servo Controller Board are shown in Table 1.

Table 1: ANDI-Servo Controller Board Specifications	
Number of servomotors	2
Driving Circuit available	Internal, or External H-Bridge
Maximum Output	1.5A average, 6A peak, 50 VDC

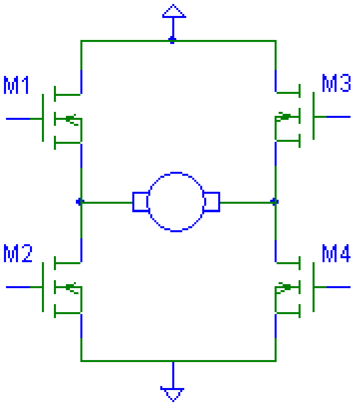

Figure 7: H-Bridge Schematic Diagram

3.1.4.2 SV-203 Servo Motor Controller Board

The SV-203 Servo Motor Control Board, shown in Figure 8, will be used for three tasks:

- Servomotor control various external servomechanisms in future expansion
- A/D Conversion of four analog inputs for IR sensors
- A/D Conversion of one analog input from the tachometer.

The SV-203 Controller Board is outlined in Table 2.

Table 2: SV-203 Controller Board Specifications
Supports control of up to eight small servomotors
RS-232 Interface
5 A/D Input – 8 bit resolution

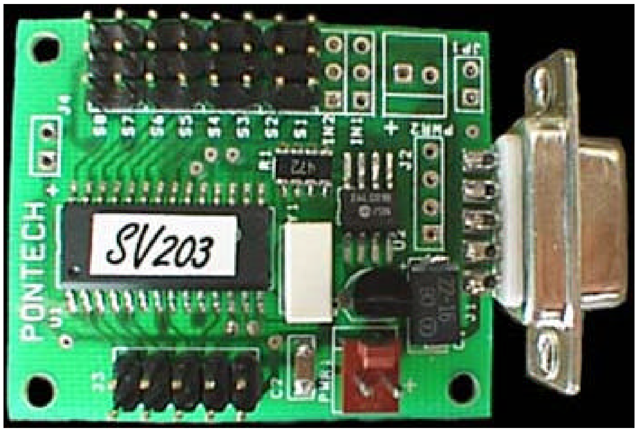

Figure 8: SV-203 Servo Motor Controller Board

3.1.4.3 R104 Relay Board

On/Off devices will be controlled by relays. The R104 Relay Board, shown in Figure 9, is a PC/104 plug-in module that can control up to 16 relays. For high current relays such as the engine starter relay, the relay board will drive an external relay rated at the appropriate current rating. The R014 Relay Board specifications are given in Table 3.

Table 3: Summary of R014 Relay Board Specifications
Up to 16 Opto-isolated Relays.
PC/104 Module
Contact Rating: 2A @30VDC

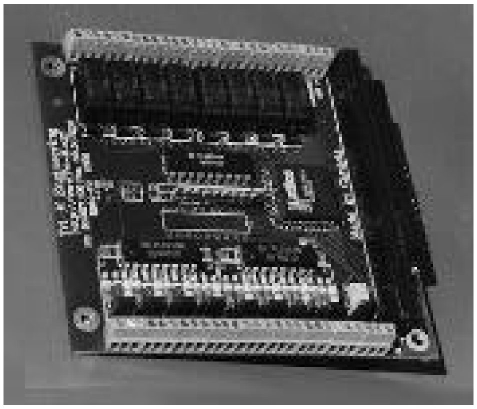

Figure 9: R104 Relay Board

3.1.4.4 Serial Interface

Eight COM ports are available for serial devices. This RS-232 interface will support the GPS, wireless radio, compass, 68HC11 microprocessor, and SV-203 IO Board. A summary of the serial port connections is given in Section 3.4.1.

3.1.4.5 OMNEX Wireless Bi-Directional Modem

The wireless modem will be used to communicate with the remote base station. The modem interfaces to the computer via RS-232, and has a data throughput of 9600 baud.

Figure 10: OMNEX DX-900 Radio Modem

3.2 Sensor Array

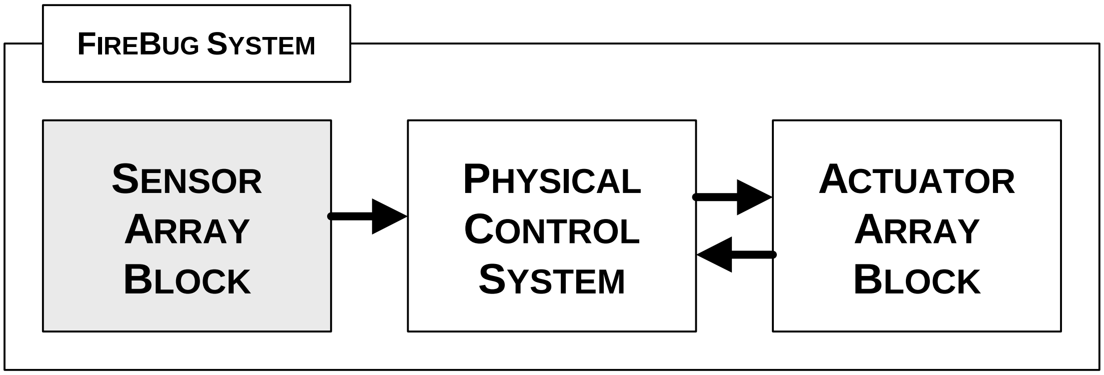

Figure 11: Sensor Array Context

The sensor array provides the Vehicle Control System (VCS) with the feedback required for trajectory planning and motion control. The sensor array block includes a position sensor, speedometer, accelerometer, heading sensor, and various sensors required for obstacle detection.

3.2.1 Position Sensor

The NovaTel MiLLenium RT-2 Differential Global Positioning System (DGPS) will act as the position sensor. An additional DGPS unit is connected to the base station to collect reference GPS information. DGPS correction is performed by the DGPS system on board the FireBug using the information received from the base station. DGPS correction will provide a position accuracy of 2cm.

The NovaTel DGPS uses string I/O via RS-232 serial port. For example the following command is sent to the DGPS to request position information:

```
log com1, prtka, ontime, 1, 0, hold
```

This log command results in the DGPS returning a long string of position information every 1 second through COM 1. The NovaTel MiLLenium RT-2 DGPS is shown in Figure 12 with two types of enclosures.

Figure 12: NovaTel MiLLenium DGPS

3.2.2 Speedometer

A surplus DC brush motor attached to the drive chain will provide speed feedback. The speed-to-voltage characteristics of this motor can be tested using available lab equipment. The chosen motor offers little mechanical resistance, and will not inhibit the motion of the vehicle.

The speedometer provides feedback needed by the throttle control system (see Section 3.1.1).

3.2.3 Heading Sensor

The Precision Navigation TCMVR-20 electronic compass will provide the VCS with heading information through an RS-232 serial port. The TCMVR-20 compass has an accuracy of ± 2 degrees. The compass will be contained within the computer enclosure for protection and the compass' reference point will be parallel with the major axis of the vehicle. The TCMVR-20 compass circuit board is shown in Figure 13.

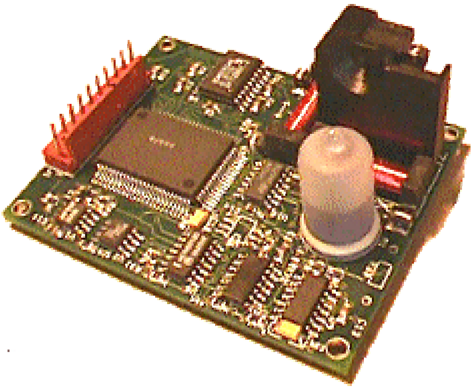

Figure 13: TCMVR-20 electronic compass

3.2.4 Object detection

The FireBug will detect objects with an array of ultrasonic and infrared sensors. The sensor array will be organized as depicted in Figure 14. Locations of the ultrasonic sensors have been chosen to maximize the field of vision and to minimize the number of sensors required for optimal area coverage. Initially, one infrared sensor will be mounted on the front of the FireBug to detect fire and hot objects straight ahead. Future prototypes will use a more complex infrared detector layout.

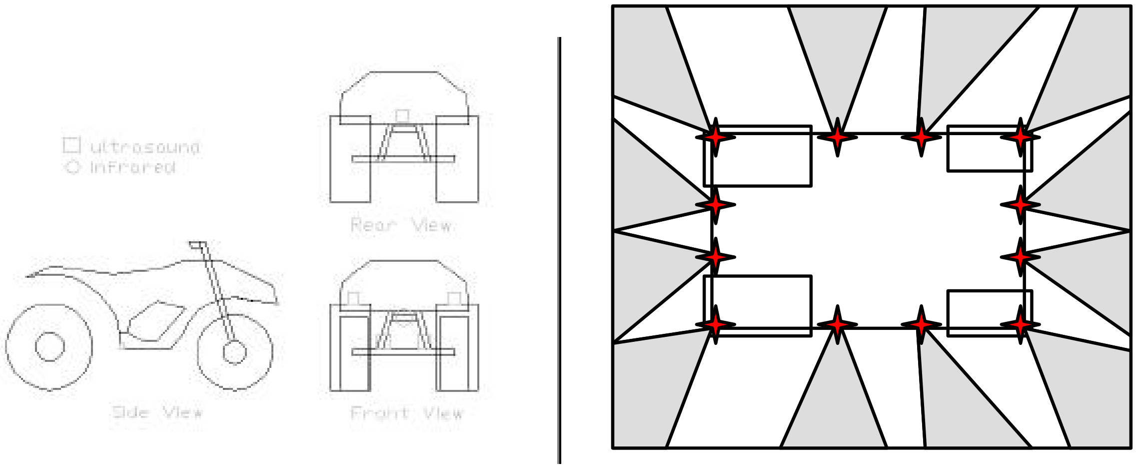

Figure 14: Vehicle Mounting Locations of Object Detecting Sensors

3.2.4.1 Ultrasonic Sensors

Ultrasonic sensors operate by emitting high frequency sound waves, and collecting the reflections of these waves to detect the presence of an object. Each sensor will operate as both an ultrasound wave emitter and receiver. Non-trivial signal processing is required for ultrasonic object detection. To avoid needless development costs, FireBug Creations has chosen to collaborate with TRAC Technology, another company using ultrasound sensors.

A block diagram depicting the conceptual operation of the ultrasonic object detection system is shown in Figure 15.

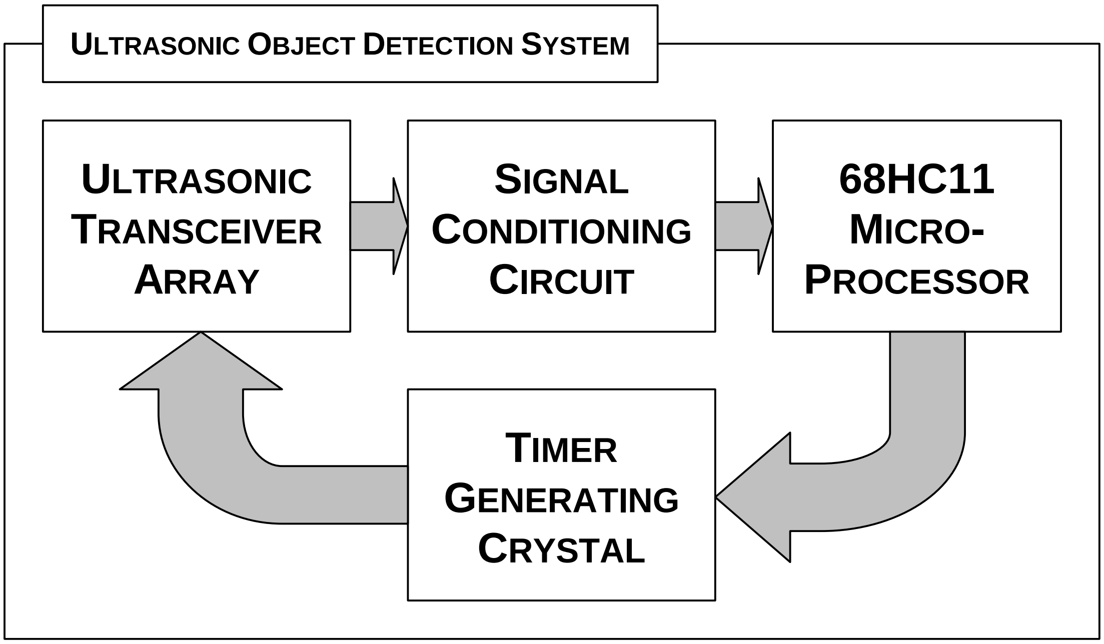

Figure 15: Ultrasonic Object Detection System Diagram

Figure 16 displays the sensor signal conditioning circuit. The input stage amplifies the sensor input and buffers the signal to avoid significant current draw from the sensor. A non-inverting amplifier configuration is used because of its high input impedance compared to an inverting amplifier. A future addition may be a differential input stage, to reduce common mode noise from the sensors. The second stage is a simple envelope detector. The 40kHz sinusoidal signal is converted to a DC signal proportional to the amplitude of the input signal. The final output stage swings between logic high and logic low, where logic high corresponds to the output of the envelope detector being higher than the reference.

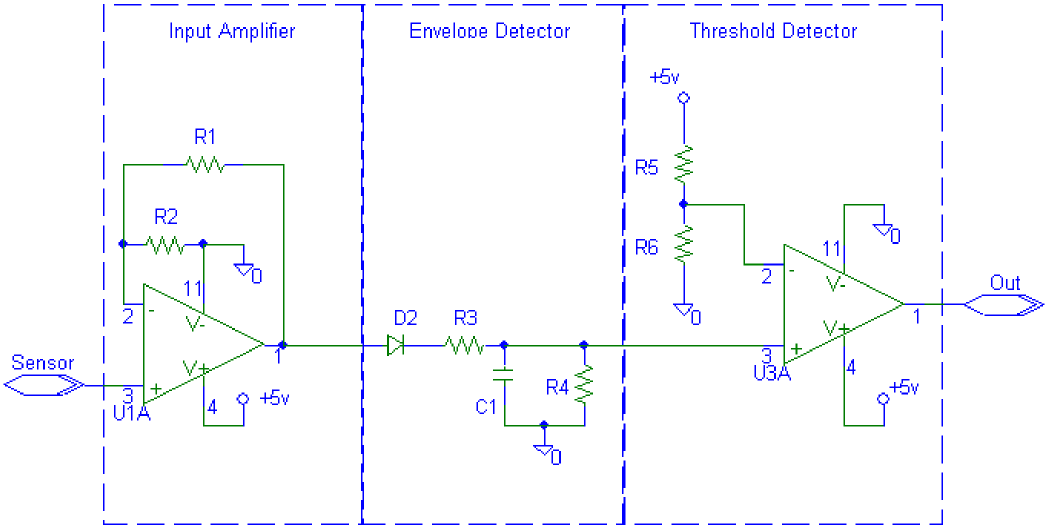

Figure 16: Sensor Signal Conditioning Circuit

The specific ultrasonic sensor chosen, the Murata MA40B8R/S, has a range of 6 meters, resolution of 9 mm and can operate normally in temperatures from –30 degrees to +85 degrees Celsius. The MA40B8R/S operates at a nominal frequency of the 40kHz. Figure 17 shows the Murata MA40B8R/S ultrasonic sensor.

Other ultrasonic sensors are under investigation by FireBug Creations. A list of all ultrasonic sensors being considered may be found in Appendix A, Parts List. The specifications for these sensors may be found on the Internet at http://www.murata.com.

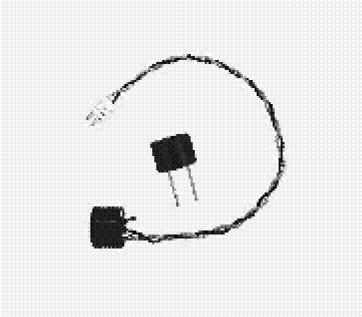

Figure 17: Murata MA40B8R/S Ultrasonic Sensors

The ultrasonic transceiver emits an ultrasonic pulse at 40kHz. The time for a wave to return to the sensor determines the distance of a given object. Table 4 provides details on the MA40B8R/S ultrasonic emitter/sensor.

Table 4: Murata MA40B8R/S Ultrasonic Sensor Specifications

Construction	Open structure
Nominal Frequency	40 kHz
Sensitivity	-63dB typical (0dB = 10V/Pa)
Direcitivity	50°
Capacitance	2000pF
Operation Temp	-30°C to 85°C
Detectable Range	0.2 m to 6 m

3.2.4.2 Infrared Sensors

Several infrared (IR) sensors are being considered for heat detection and fire identification. Initially the Murata IRA-E410QW1 pyroelectric IR sensor, mounted as shown in Figure 14, will be used to locate and identify fire and differentiate fire from other solid obstacles. Table 5 gives a summary of the IRA-E410QW1's specifications. Figure 18 provides a photo of two such sensors

Other infrared sensors are under investigation by FireBug Creations. A list of all infrared detectors being considered may be found in Appendix A, Parts List. The specifications for these sensors may be found on the Internet at http://www.murata.com.

Table 5: Murata IRA-E410QW1 Pyroelectric Sensor Specifications	
Sensitivity	1.3mVp-p
Wave Length Range maximum	4.3µm
Field of View	17°
Optical Filter	4.3µm Band Pass Silicon
Electrode	Fai 1.6mm
Supply Voltage Range	3V to 15V
Storage Temp. Range	-30°C to 100°C

The analog output of the IR sensors will be amplified sufficiently to provide adequate signal amplitude to the SV-203 Servo Board.

Figure 18: Murata IRA-E410QW1 Pyroelectric Sensor

3.3 Actuator Array

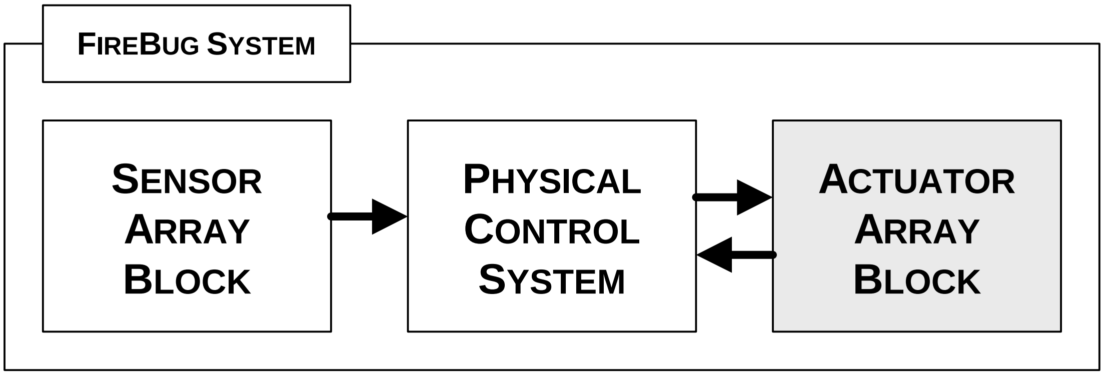

Figure 19: Actuator Array Context

The actuator array includes electric motors used for throttle and steering control, solenoid actuators used in the braking and gearing systems, and various relays. Sensors internal to a given actuator are considered to be part of the actuator array. The VCS provides all instructions to the actuator array block.

3.3.1 Throttle

To automate the control of the throttle a small high-torque servomotor will be installed. We chose the Hitec HS-805 Mega Servo, shown in Figure 20, to perform the task of throttle adjustment. The HS-805 receives input in the form of pulse width modulation (PWM). The ANDI-Servo Control Board will receive throttle instructions from the VCS and translate the digital input into a pulse train with a duty cycle representing a specific angle. The angle of rotation of the motor then corresponds to a particular throttle position. The HS-805, commonly used with remote control vehicles, will provide sufficient torque and reasonable response time for this throttle control application. The ANDI-Servo Control Board is described in Section 3.1.4.1. The Hitec HS-805 servomotor specifications are listed below in Table 6.

Table 6: Hitec HS-805 Mega Servo Specifications

Dimensions:	2.36" x 1.18" x 2.24"
Weight:	4.23oz
Torque:	224 oz-in
Transit Time:	0.20 sec/60°

Figure 20: Hitec HS-805 Mega Servo

3.3.2 Steering

Preliminary measurements indicate that the steering system will require approximately 40Nm of torque to turn the wheels. A DC motor capable of sustaining this torque is expensive, so FireBug Creations has developed a simpler and cheaper solution to the steering system.

Since the torque supplied for steering is intermittent in nature, the FireBug does not require a motor that is rated for continuous high output. The Pittman Series GM14604 Lo-Cog Brush Commutated DC Gearmotor can provide ample power. Table 7 provides the specifications of the Pittman GM14604 servomotor that will act as the steering actuator. An additional 6:1 gearbox will be connected to the output shaft of the gear-motor to amplify the torque. The final gear reduction ratio will be 144:1. This configuration will allow our steering system to provide adequate torque to turn the wheels of the vehicle.

Table 7: Pittman GM14604 Gearmotor Specifications

Max Gearbox Torque	2000 oz-in
Output Shaft No-Load Speed	154 RPM
Motor Stall Torque	204.00 oz-in
Output Shaft Stall Torque	3133 oz-in
Efficiency	73%
Dimensions (length × diameter)	6.7 in × 2.1 in
Winding Voltage	24.00 VDC
Encoder	100 CPR
Reduction Ratio	24
Rated Voltage	24.00 VDC
Stall Current	23.8 A

The ANDI–Servo Control Board servomotor driver coupled with an H-bridge will provide the control and power required to drive the GM14604 gearmotor. The ANDI board will receive feedback from the gearmotor's optical encoder and provide precise angular position control. The specifications for the ANDI-Servo Control Board are given in Section 3.1.4.1.

3.3.3 Braking

The braking system will use solenoid actuators to pull the brake cables. This will provide simple on/off control of the brakes for our first prototype. Since the FireBug's top speed is limited, there will be no loss of control with this type of brake control. The solenoids will be chosen and calibrated based on requirements determined during field tests. The ATV has two front disc brakes and one rear drum brake, all of which will be outfitted with solenoid actuators.

3.3.4 Gearing

The FireBug's automatic transmission eliminates the need to shift gears, so only a system to switch between forward, neutral and reverse is required. Two Detroit Coil Co. Model 28-462 solenoid actuators will be used for the gear switching system. The two coils are attached to the gear lever as shown in Figure 21.

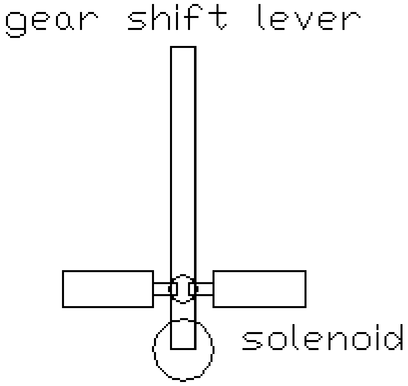

Figure 21: Solenoid Actuator Gearing Mechanism

When either of the two solenoids is excited, the gear lever will be push away from the active solenoid into either forward or reverse gear. When both solenoid actuators are energized, the opposing forces will return the gear lever to the neutral position.

3.3.5 Ignition

The ignition will be actuated by means of a relay. The R104 relay board consists of several relays and is controlled by the VCS as described in Section 3.1.4.3. The key ignition switch will remain intact as a safety measure and security feature.

3.3.6 Additional Actuators

Future features include a flashing fire-truck lamp and fire extinguishing system. Additionally, up to 8 extra small servos may be controlled through the SV-203 controller board. The R104 Relay Board has 20 relays available, so future expansion can be accommodated easily.

3.4 Embedded Processor

3.4.1 Processor Hardware

The embedded processor of choice is the PCM-5894/5892 PC/104 Pentium 200MHz compact motherboard. Information about this motherboard is supplied in Table 8.

Table 8: PCM-5894/5892 Compact Motherboard Features

CPU	Intel Pentium 200MHz
BIOS	Award FLASH BIOS
Chipset	SiS5582/5598
2nd Level Cache	On board 512KB pipeline burst
RAM	64MB 72-pin SIMM
Parallel Port	Supports SPP, EPP, ECP
Serial Ports	7 RS-232, 1 RS-232/422/485 serial ports
USB	Dual USB connectors
DMA Channels	7
Interrupt Levels	15
SSD Interface	32-pin DIP socket supports M-Systems DiskOnChip 2000
Ethernet Interface	• Chipset: Realtek8139 100-BaseT Ethernet Controller • Interface: 10-pin header supports RJ-45 jack
Expansion Slots	• PC/104 connector: 104 pin for 16-bit bus expansion • PCI slot: one PCI slot for expansion
Environmental	• Power Supply: +5V and +12V • Power Requirements: Bare Board: +5V @ 1.5A Pentium MMX: +5V @ 6A • Operating Temperature: 0 to 60ºC • Board Size: 8”(L) × 5.75”(W)

Each hardware element in the physical control system must be interfaced to the embedded processor. The PC/104 connector can be used to daisy chain multiple devices to the processor.

Other peripherals must be connected through another port, such as a serial or parallel port. Peripheral devices most commonly support the serial ports, so many COM ports are used in the FireBug system. A summary of the serial ports to be used is given in Table 9.

Table 9: Serial Port Connections Summary

Port	Connected Device
COM 1	NovalTel GPS
COM 2	TCMVR-20 Compass
COM 3	Wireless Bidirectional Modem
COM 4	SV-203 IO Board
COM 5	68HC11 Microprocessor
COM 6,7,8	Future Expansion

3.4.2 Operating System

FireBug Creations has decided to use Linux, an open source version of the UNIX operating system. While there are many other operating systems available for embedded systems, Linux combines speed and power with small size and portability to most UNIX platforms, including Solaris.

Table 10 shows a comparison of different operating systems and their features.

Table 10: Comparison of Operating Systems

OS Name	Real Time	Multitask	Multi-user	Coding Ease
Linux	Yes	Yes	Yes	Easy
QNX	Yes	Yes	No	Easy
Sun Solaris	No	Yes	Yes	Easy
MS Windows NT	No	Yes	No	Difficult
MS Windows CE	No	Yes	No	Strained
MS DOS	No	No	No	Painful

3.5 Control System Software

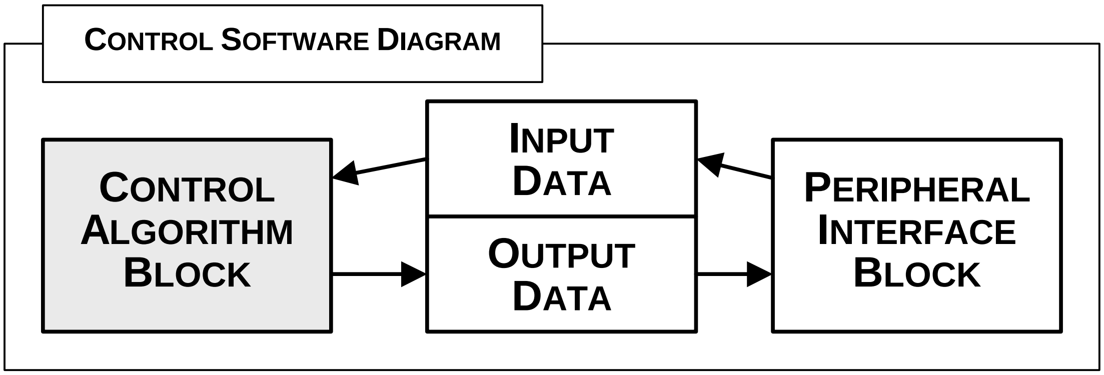

Figure 22: External Interface to Embedded Controller

The Control Algorithm Block (CAB) processes the input data from all external sensors and decides how to control the vehicle. It updates the output data based on its decisions. The Peripheral Interface Block (PIB) is responsible for supplying digital input data from external sensors, and supplying analog or digital output data for external actuators.

3.5.1 High Level Control Algorithms

3.5.1.1 Initial Trajectory Planning

Destinations are supplied by the airship. Such destinations can arrive at any time during the vehicle's autonomous execution. An initial course is planned based on the factors shown in Table 11. Avoiding known obstacles minimizes risks associated with real-time object avoidance.

Table 11: Factors Affecting Initial Trajectory

Factor	Importance and Justification	Priority
Destination Priority	High – survivors have priority over fires, for example	1
Danger Level	High – FireBug will avoid situations of extreme danger unless expressly ordered near them	2
Trajectory Length	Small – it is difficult to minimize distance and maintain a fast calculation time	3

In addition to the real destination, the initial trajectory from the current position to a destination may be composed of several sub-destinations. Figure 23 offers a conceptual picture of sub-destinations.

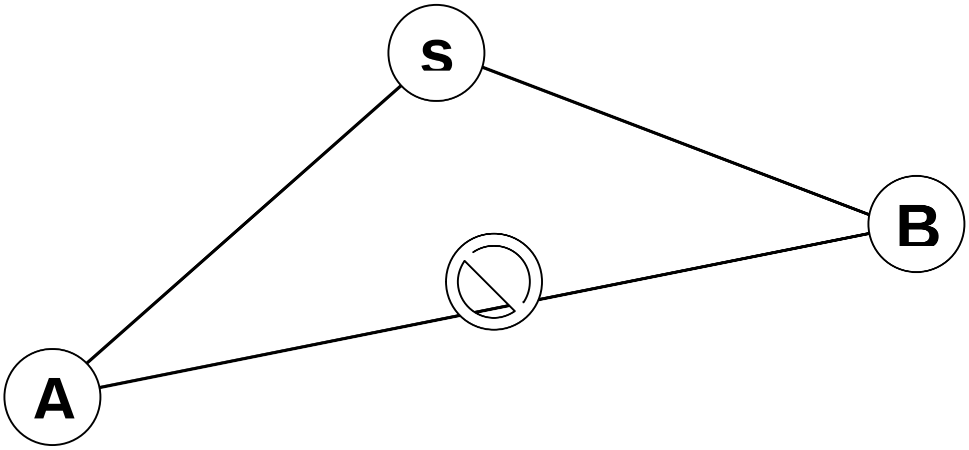

Figure 23: Conceptual Picture of Sub-Destination

3.5.1.2 *Real-time Object Avoidance and Trajectory Correction*

Given that the airship may not provide complete data, the vehicle is equipped with ultrasonic and infrared sensor arrays. The sensor arrays allow the FireBug to avoid obstacles not predicted in its initial trajectory.

The algorithm for the real-time object avoidance is not as flexible as the initial trajectory planning. The FireBug may be travelling at its top speed, and it must correct its trajectory before a collision occurs. Given the range of the sensors available and the top speed of 2.8 m/s, the FireBug must correct its trajectory within 2 seconds. The algorithm for real-time object avoidance needs to be simple and fast, but accurate. The limitations affecting real-time object avoidance are shown in Table 12.

Table 12: Limitations Affecting Trajectory Correction

Limitation	**Effect on Trajectory Correction**	**Worst Case**
Sensor range	A larger sensor range allows more reaction time.	2 m
Vehicle response time	If the FireBug responds more quickly to setpoint changes, the controller can take more time to calculate a safe trajectory	1 s
Vehicle speed	If moving faster, the FireBug has less time to react to a new obstacle	2.8 m/s

The algorithm used for real-time object avoidance is a simple scanning technique. The sensors will be laid out in a format as shown in Section 3.2.4, allowing a slight overlap of sensor range in the forward direction.

At every time slice, the vehicle will check each sensor. The vehicle will only scan the sensors that are on the critical side of the vehicle, and only the two smallest distances will be used. If the closest distance is farther than the size of the guard-band, it is ignored and no corrective action is taken. If the sensor does detect an obstacle within the guard-band, the speed is reduced, and the heading is corrected in the opposite direction. If the object is directly ahead, then the brakes are applied and the wheel is turned to avoid the obstacle.

The algorithm for trajectory correction is further defined in Table 13.

Table 13: Trajectory Correction Algorithm

Closest Sensors in	Description of Situation	Correction
Both readings are greater than guard-band	No obstacles are within critical radius.	No action taken
One reading is within guard-band; but not a tall object	There is an obstacle, but we can drive over it.	Reduced speed to get over obstacle
One reading is within guard-band; tall obstacle	There is an obstacle to avoid.	Reduced speed, and heading corrected
Both readings are within the guard-band	There is an obstacle directly ahead, or multiple objects ahead.	Stop vehicle, back up, and avoid the area.

3.5.2 Input/Output Data

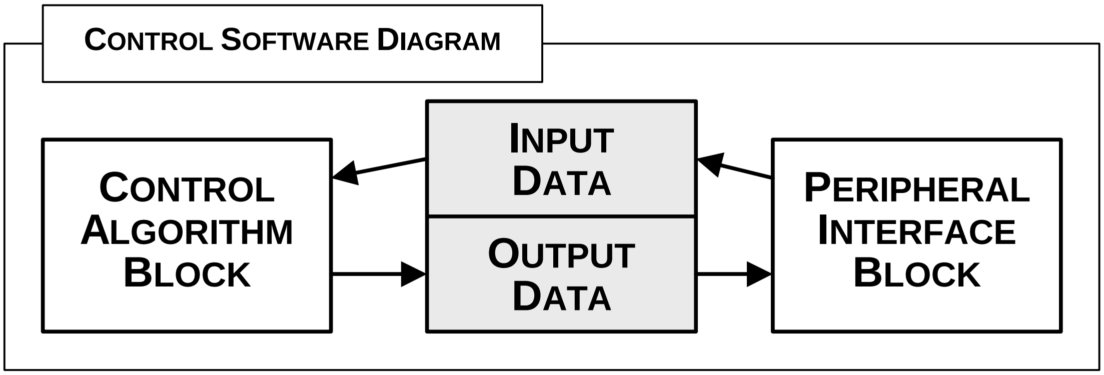

Figure 24: Conceptual Realization of Sub-Destination

The input/output data is a shared memory location used for communication between the CAB and the PIB.

3.5.3 Peripheral Interface Block (PIB)

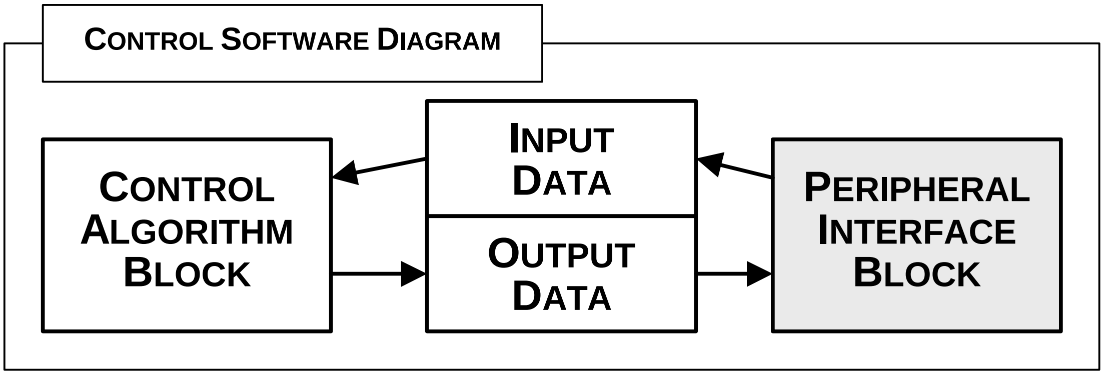

Figure 25: Conceptual Realization of Sub-Destination

The PIB is the lowest-level code in the embedded controller. It is responsible for gathering sensor data and writing it to the Input Data memory location. It also reads the Output Data memory location and writes to various hardware ports, to control the peripherals like motor controllers and relays. There are strict timing requirements for the PIB. This is the only critically real-time software block.

In addition to the primary role of a hardware interface, the PIB also has a secondary role to check the data coming from the vehicle controller for validity. If the vehicle controller crashes or is afflicted by a software bug, the PIB will correct invalid setpoints, and select the closest value within the allowed region for that setpoint. The PIB will have a table of valid regions for each setpoint.

This range of allowable setpoint values used by the PIB is given in Table 14.

Table 14: Range of Allowed Setpoint Values

Set Point	Allowable Range
Speed	0-10km/h
Brakes	On/Off
Steering	-45° to 45°
Gears	Forward/Neutral/Reverse
Ignition	On/Off

3.6 Base Station

The base station is responsible for providing the FireBug with destinations and external obstacle information. The base station operates using communication protocols transmitting over a wireless modem. Therefore, the FireBug must have the necessary hardware and software to communicate with the base station. To accurately respond to the destinations supplied, the FireBug will use the same coordinate system used by the base station.

The FireBug will also respond to the base station with information regarding map updates, vehicle status (e.g. position, speed, etc.), possible emergency situations and any permission requests. A permission request will occur if the FireBug identifies a local hazard that it could neutralize.

4.0 Physical Specifications

An all terrain vehicle (ATV) is well suited to our project. The *Yamaha Breeze,* shown in Figure 26, has been chosen as a suitable vehicle to convert into an autonomous land vehicle.

Figure 26: Yamaha Breeze All Terrain Vehicle

The ATV has sufficient stability for our application and can carry a 100-kg payload. All electronics will be enclosed in onboard containers. Information about the physical specifications of the Yamaha Breeze is summarized in Table 15.

Table 15: FireBug Vehicle Information

Dimensions (l × w × h)	1.6 m × 1 m × 1 m
Weight	135 kg
Maximum Payload	100 kg
Wheelbase	1.1 m
Ground Clearance	0.15 m
Engine	Air/fan-cooled , 4-stroke SOHC, 124 cc
Transmission	V-belt automatic (forward, neutral, and reverse)
Batteries	12 V Lead acid (starter) 2 × PS1270 High Power Batteries
Fuel Capacity	7 L
Maximum Engine Torque	8.14Nm @ 6,000 rpm

The onboard electronics of the FireBug will be in a waterproof enclosure made of metal and covered with a high-grade flexible polymer derivative. The physical dimensions of the enclosure are found in Table 16.

Table 16: FireBug Electronic Enclosure

Dimensions (l × w × h)	0.3m × 0.2m × 0.1m
Weight	3 kg

4.1 Environmental Specifications

The operating temperature the FireBug system is between -20°C to 70°C. This allows for both winter and summer operation and near sources of extreme heat. The FireBug is capable of operating in both wet and dry conditions.

4.2 Electrical Specifications

The FireBug system will be design to meet certain electrical requirements summarized in Table 17.

Table 17: FireBug Electrical Specifications

Computer Power Dissipated	200 W
Peak Servo Current	40 A (@ 24 V)
Average Total Current	15 A (@ 24 V)
Peak Power Dissipation	960 W
Average Power Dissipation	360 W

5.0 Safety Specifications

The FireBug system will meet the safety specifications outlined in the following sections.

5.1 Throttle Limiter

Both a software control (see Section 3.1.1 – Throttle Control System) and a mechanical constraint will limit the vehicle speed. The mechanical constraint will consist of a clamp placed on the throttle cable to prevent excessively high speeds from being reached. The clamp position will be determined by the throttle position that corresponds to a 10km/h vehicle speed on level ground.

5.2 Emergency Kill

The vehicle will contain a remote emergency kill switch accessible from the base station. This kill switch is separate from the computer and is activated by a relay. The vehicle will also have an onboard secondary physical kill mechanism. This will be a disconnection of a vehicle power by removing a metallic pin in the main power circuit.

6.0 Reliability Specifications

The FireBug will meet the following reliability requirements.

6.1 Accuracy

The error of the position equipment onboard the vehicle will be well within the guard band of the vehicle. The guard-band is 3m, chosen due to the time slice of the computer (0.1s – 0.25s), the turning time of the steering system (1s), and the maximum allowable speed (10 km/h).

6.2 Durability

The FireBug system will be robust enough so that it can traverse all types of terrain for up to two hours per mission. The expected lifetime of the vehicle under heavy use is five years. The system will be able to endure a noisy environment and vibrations characteristic of off-road situations.

7.0 Testing

7.1 Simulation and Modelling

To avoid risk of damage to the vehicle during testing, the FireBug Creations team is developing advanced simulation and modeling software. This will allow many control algorithms to be tested before the FireBug is ever built.

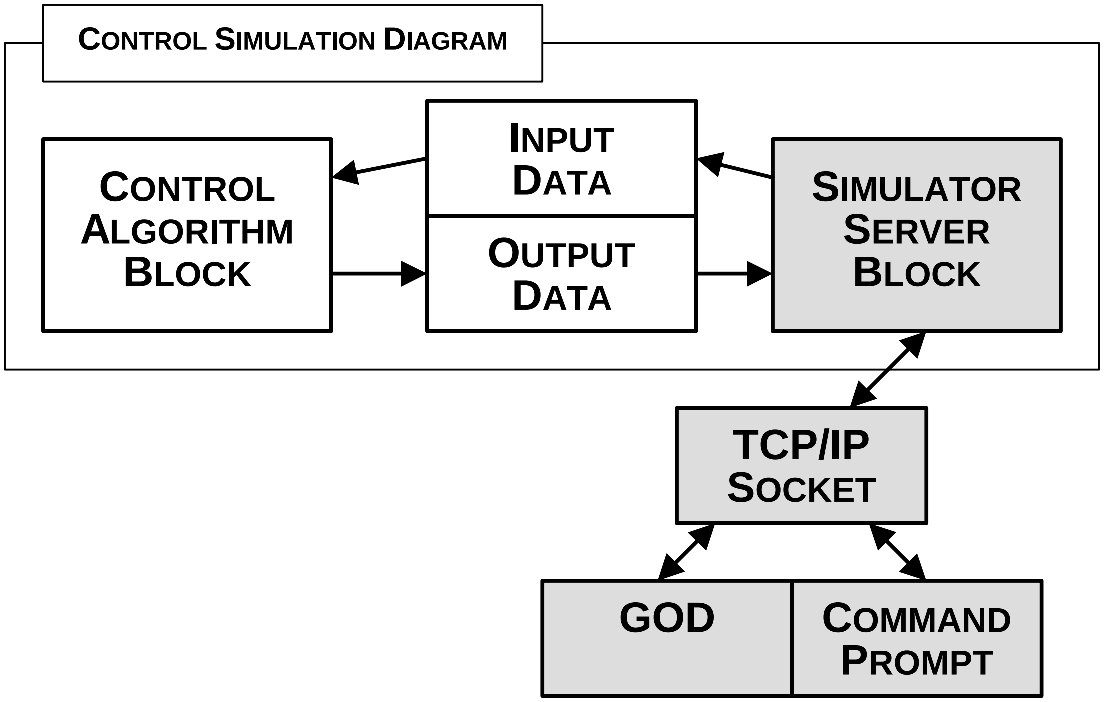

Figure 27: FireBug Simulation Diagram

When solid and reliable algorithms have been developed and simulated successfully, they will be tested in the physical FireBug under similar scenarios. The testing scenarios will be designed to stress the system, and to examine its performance.

7.1.1 Graphical Output Display (GOD)

To simplify the simulation results, the simulator will produce a graphical output format. To decrease the load on the embedded processors, the GOD will be run on an external processor. The GOD retrieves its data from the simulator via a TCP/IP socket connection. Each different type of data is sent in a uniquely marked packet. Table 18 shows the different types of data the GOD displays, and the unique packet label associated with each.

Table 18: GOD Interface to Simulator

Packet Type	**`Packet Identifier`**
Vehicle position	`SIMULATOR_VEHICLE_POSN`
Vehicle heading	`SIMULATOR_VEHICLE_HEADING`
Obstacle	`SIMULATOR_OBST`
Destination	`SIMULATOR_DEST`
Control timeslice	`SIMULATOR_CONTROL_TIMESLICE`
Number of destinations	`SIMULATOR_NUM_DEST`
Number of obstacles	`SIMULATOR_NUM_OBST`
Data read request	`CLIENT_READ_REQUEST`
Server response	`SERVER_RESPONSE`
Start simulation	`SIMULATOR_STARTSIM`
Stop simulation	`SIMULATOR_STOPSIM`

7.1.2 Command-Line Prompt Input

The Command-Line Prompt will be used to request information. In addition, the prompt will be used to activate the GOD. The commands available at the prompt are listed in Table 19.

Table 19: CLP Commands and Descriptions

Command	**Args**	**Description**
`.help`		Displays all command with a brief description of each
`.dtslice`		Displays timeslices of vehicle and simulator
`.stslice`		Sets simulator timeslice
`.vtslice`		Sets vehicle timeslice
`.isim`		Starts interactive simulation
`.dest`	`(x,y,r)`	Adds a new destination
`.obst`	`(x,y,r)`	Adds a new obstacle
`.posn`	`(x,y)`	Set vehicle position
`.gogod`		Activates the graphical output display
`.nogod`		Deactivates the graphical output display
`.dsens`		Displays the values of all sensors
`.dacts`		Displays the values of all actuators
`.dests`		Displays the list of all destinations
`.obsts`		Displays the list of all obstacles
`.prn`		Displays all sensors, actuators, and lists

7.1.3 Accelerated Simulator

For fast verification of algorithmic changes, it is important that the simulator be able to speed up the simulation. Since the simulation scenarios may involve large distances, the actual time involved to complete a test run may be several hours. The Accelerated Simulator will:

- decrease the simulation timeslice to the smallest time possible
- enforce ratios of simulator timeslice to vehicle timeslice
- allow a pause at any time during its execution
- disable the GOD for the duration of the simulation

7.1.4 Real Time Simulator (RTS)

Since the software developed for the vehicle controller will be tested directly with the simulator, it is important that it be tested at real-time for final tests. In addition, the RTS is necessary if the user wishes to watch the GOD. The RTS will:

- execute at user-specified (or default) simulator timeslice and vehicle timeslice
- allow a pause at any time during its execution
- update the GOD (if active) during the simulation

7.1.5 FireBug Physical Model

For the simulation results to be reasonable, accurate physical models of the FireBug's control system (including servos, engine, brakes, etc) must be developed. Since the specifications of most part are available from the manufacturer's data books, this task will be aided. The FireBug Physical Model, used by the FireBug to control itself, will take into account these factors:

- response time of any servo motors used
- engine response time
- brake response time
- steering response time
- current state of all sensors
- current state of all actuators
- vehicle's guard-band
- list of current destinations
- list of current obstacles

7.1.6 Environmental Models

In addition to accurate models of the FireBug, a good Environmental Model is important. The Environmental Model is used by the Simulator to determine the next state of the vehicle's actuators (i.e. position, velocity, acceleration) and sensors (objects, their height etc.). The Environmental Model will take into account the following factors:

- coefficient of friction of the ground
- gravity
- the current state of the vehicle's actuators

7.2 Physical Test Plan

Obviously, no amount of simulation can entirely substitute field tests. Therefore, FireBug Creations has developed several important field tests.

7.2.1 Safety Requirements

Before any simulated scenarios are tested in the field, the Safety Requirements specified in Section 5.2 must be tested

7.2.1.1 Software Kill Flag

Table 20: Software Kill Flag Test	
Test 1: Stationary Test	
Setup	Turn on the FireBug. Allow the vehicle to stabilize.
Action	From the Base Station, send the software kill command.
Verify	Ensure that the engine stops within a reasonable time
Test 2: Moving Test	
Setup	Turn on the FireBug. Allow the vehicle to stabilize. Program a destination in front of the vehicle: 100m
Test	From the Base Station, send the software kill command
Verify	Ensure that the vehicle stops within a reasonable time Ensure that the engine stops within a reasonable time

7.2.1.2 Manual Remote Kill Switch Test

Table 21: Manual Remote Kill Switch Test	
Test 1: Stationary Test	
Setup	Turn on the FireBug. Allow the vehicle to stabilize.
Action	Hit the manual kill switch
Verify	Ensure that the engine stops within a reasonable time
Test 2: Moving Test	
Setup	Turn on the FireBug. Allow the vehicle to stabilize. Program a destination in front of the vehicle: 100m
Test	Hit the manual kill switch
Verify	Ensure that vehicle/engine stop within a reasonable time

7.2.2 Simulated Scenarios Confirmation

In addition to safety requirements, the FireBug's performance under simulated conditions must be tested. These tests are summarized below:

Table 22: Simulated Scenarios Test Expectations	
Test 1: Close or Crowded Quarters	
Setup	Arrange a series of obstacles in a closely packed arrangement, as shown
Action	Specify a destination in an area near in the centre of such an arrangement
Verify	Ensure that the FireBug's actual behaviour is close to its simulated behaviour. The desired response should include: • decreased speed • smaller (but still safe) guard-band • no collisions with obstacles • successful arrival at destination
Test 2: Destination is a Fire	
Setup	Build a small and controlled fire
Test	Specify a destination close to the fire
Verify	Ensure the FireBug's behaviour is close to its simulated behaviour. The desired response should include: • decreased speed near fire • confirmation of a fire found • activating fire extinguishing system
Test 3: Unexpected Obstacle Found	
Setup	Place an obstacle 25m in front of the FireBug
Action	Specify a destination beyond the obstacle, so that the obstacle will pass within the FireBug's guard-band if it does not alter its trajectory
Verify	Ensure the FireBug alters its trajectory appropriately. The response should include: • decreased speed upon recognition of object • altered course to avoid object • successful arrival at the destination

7.2.3 Un-simulated Scenarios Confirmation

As well as testing simulated scenarios, there are additional scenarios that should be tested that are difficult to simulate. These scenarios are summarized below:

Table 23: Un-simulated Scenarios Test Expectations

Test 1: Foggy Environment	
Setup	Place an object some distance in front of the FireBug on a foggy day.
Action	Specify a destination beyond the object, so that the object will pass within the FireBug's guard-band if it does not alter its trajectory.
Verify	Ensure that the FireBug alters its direction in the same fashion as on a clear day.
Test 2: Extinguishing Fires	
Setup	Set up a small and controlled fire
Test	Specify a destination near the fire
Verify	Evaluate the success of the FireBug's attempts to extinguish the fire. A successful attempt should include: • activating the fire extinguishing system • a minimum time spent at the fire • a significantly quenched fire; if not completely extinguished, at least much smaller • no more than the FireBug's maximum time spent at the fire

8.0 References

More information regarding this project and related projects can be found on the following web locations.

SFU Aerial Robotics Group
http://www.ensc.sfu.ca/research/mirosot/ARG/

Association for Unmanned Vehicle Systems (AUVS), International Aerial Robotics Competition
http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html
Details on the Aerial Robotics Competition can be found on the AUVS web site.

TRAC Technology
Ultrasonics processing board

Chris Watkiss, Professor, Capilano College
Miscellaneous utility source code

Ajeco
ANDI-Servo Controller Board
http://www.ajeco.fi

Pittman
Steering Servo Motor
http://www.pittmannet.com/

MuRata
Ultrasonic, pyroelectric, infrared sensors
http://www.murata.com/

Detroit Coil Co.
Solenoids
http://www.detroitcoil.com

Pontech
Servo motor controller board, A/D board
http://www.pontech.com/

Hitec
Throttle servo motor
http://www.hitecrcd.com/

International Rectifier
MOSFET drivers
http://www.irf.com/

9.0 Sponsors

These companies and individuals have generously offered sponsorship

Tri-M Engineering
http://www.tri-m.com/

Precision Navigation
http://www.precisionnav.com/

AAEON
http://www.aaeon.com

OMNEX
http://www.omnex.com

NovaTel
http://www.novatel.com

BC Advanced Systems Institute
http://www.asi.bc.ca

10.0 Glossary of Terms and Acronyms

Glossary	
ARG:	Aerial Robotics Group
ATV:	All-Terrain Vehicle
BAUD:	Bite All Undergraduate Dogs
CAB:	Control Algorithm Block
Destination:	Objects of interest, including bodies (dead or alive) and fires.
DGPS:	Differential Global Positioning System
GOD:	Graphical Output Display
Guard-Band:	The guard-band is a safety distance and is a function of the vehicle's speed and the response time of the VCS.
IARC:	International Aerial Robotics Competition
IR:	Infrared
Linux	A free, open-source version of the UNIX operating system
MOSFET:	Metal Oxide Semiconductor Field Effect Transistor
PIB:	Peripheral Interface Block
PID:	Proportional/Integrator/Derivative Controller
RTS:	Real Time Simulator
Setpoint:	Desired output value of a system
Slice Time:	See time slice
Sub-Destination:	An intermediate goal of the vehicle used to accomplish the overall goal of arriving at a point near a destination. Sub Destinations are also used to evade objects.
VCS:	Vehicle Control System
TCP/IP:	Transmission Control Protocol / Internet Protocol

Time Slice:	The time interval used in timing applications (i.e. Δt).
TPA:	Trajectory Planning Algorithms
US:	Ultrasound

Appendix A: Parts List

Table 24: Parts List

Part Number	*Manufacturer*	*No*	*Description/Use*	*Section*
ANDI Servo Board	Ajeco	1	Controls 2 high power servo motors	3.1.4.1
SMP60N03	Siliconix	8	High Power MOSFETS: 60A at 30V V_{DSS}, max continuous power 105W	3.1.4.1
SV-203 Servo Board	Pontech	1	Controls 8 low power servos, A/D conversion capabilities	3.1.4.2
R104 Relay Board	Tri-M Engineering	1	Opto-isolation of high power devices	3.1.4.3
DX-900	OMNEX	1	Wireless RS-232 Radio Modem	3.1.4.5
MiLLenium RT-2 DGPS	NovaTel	1	Position feedback using DGPS	3.2.1
TCMVR-20 Compass	Precision Navigation	1	Directional sensor used for heading information and navigation	3.2.3
MA40S4R/S MA40E7R/S MA40B8R/S MA40B7	MuRata	4 4 2 2	Ultrasonic Sensors	3.2.4.1
IRA-E410QW1	MuRata	2	Pyroelectric IR Sensor	3.2.4.2
IMD-B101-01 IMD-B102-01	MuRata	2 2	IR Detectors	3.2.4.2
HS-805 Mega-Servo	Hitec	2	Low power R/C servo motor	3.3.1
GM14604	Pittman	1	Lo-Cog Brush Commutated DC Gearmotor	3.3.2
28-462	Detroit Coil	3	Solenoid Actuator	3.3.4
PCM-5894/5892	Tri-M Engineering	1	PC/104 Pentium 200MHz compact motherboard	3.4.1
Breeze	Yamaha	1	All Terrain Vehicle	4.0
PS 1270	Power Sonic	2	High Power Battery, 12V, 7Ah	4.0