Firefighting Robot

Oklahoma State University Engineering
Stillwater, OK
Team 9

Project Engineers:
Shoaib Afandi
Jaime Austin
Asim Behera
John Porterfield, Team Leader

Abstract

This paper demonstrates the requirements, specifications, design problems and solutions for the Fire Fighting Robot project fulfilling the requirements for Senior Design II at Oklahoma State University. The robot is proven to follow all rules and regulations for the Trinity College competition which can be found online at the Trinity College Website http://www.trincoll.edu/events/robot/Rules/Rules2004.pdf, and can be re-constructed using the ideas from this document.

The contest takes place in Hartford, CT annually, and its purpose can be summed up as follows:

> "The specific challenge of this contest is to build a computer controlled Robot that can move through a model floor plan structure of a house, find a lit candle and then extinguish it in the shortest time subject to a few operating factors [...]. This is meant to simulate the real-world operation of a Robot performing a fire protection function in an actual home. The candle represents a fire which has started in the home which the Robot must first find and then extinguish." [1]

Table of Contents

Introduction

Team 9's final project submitted for Senior Design II is a fire-fighting robot designed to compete in the Trinity College Fire Fighting Robot Contest. "The specific challenge of this contest is to build a computer controlled Robot that can move through a model floor plan structure of a house, find a lit candle and then extinguish it in the shortest time subject to a few operating factors [...]. This is meant to simulate the real-world operation of a Robot performing a fire protection function in an actual home. The candle represents a fire which has started in the home which the Robot must first find and then extinguish." [1]

To successfully win the contest a robot must completely navigate the maze while accruing the least amount of points possible. By completing various extra tasks along the way, it can also get a percentage reduction in final score (known in the contest as an "operating mode"). As a specific example, our group chose to compete in the "Sound Activation Mode," which required our robot to be started by an external alarm at a frequency of our choosing as opposed to a start button or switch. This adds a multiplier of .95 to our final score, thereby awarding us with a reduction of final score.

Penalties are assessed during the competition which can increase your score as well. These penalties are levied when the robot touches a wall, slides against a wall, or touches a lit candle. The website's rules [1] do not place much emphasis on worrying about the penalties, stating "These penalties are generally a small price to pay for a Robot that manages to accomplish the task." [1]

The specifics of how the robot's final score is tallied and how penalty points are assessed is discussed in Appendix A of this paper.

Mechanical (Asim Behera)

Asim Behera was responsible for the mechanical aspect of the robot. This included:

1. Manufacturing the robot.
2. Integrating the motor and wheels after modification.
3. Installing and mounting all of the sensors to the chassis.
4. Supplying adequate power to the robot.
5. Design of the extinguishing circuit and its installation.
6. Painting the robot.

Design

The initial challenge we faced was to choose a specific design for the chassis. Our philosophy was to keep everything simple. Through research on past projects, we found numerous options. After a lot of research and weighing the pros and cons of several past designs we decided to make our robot round. The major technical consideration of this design was, the robot would not have any sharp edges so it would not get stuck in any corner of the floor plan. Keeping the above philosophy in mind, we also mounted the wheels for the robot within the circumference of the chassis, so that nothing sticks out. The specification laid down by the Trinity College Fire Fighting Robot Contest is that the physical dimension of the robot should fit in a 31 cm cube box. Adhering to this judiciously we decided on a two level design. The upper deck is mounted on the lower deck via long screws. This enabled us to adjust the final height of the robot after everything was installed on it. The final dimensions of our robot are (H x W x L) = (28 x 25 x 25 cm). For greater balance and agility we decided to mount the motor and wheels across the center of the chassis(See Appendix B). This mounting gives us a tighter turning radius than models with one caster. The chassis is supported by two casters: one in front, and one at the back. The casters are mounted in such a way that the whole robot has a see-saw effect to it. This is purposefully done to accommodate any uneven section in the floor plan which is a design requirement. Trinity College Fire Fighting Robot Contest asks to accommodate discontinuities in the floor plan no greater than 5 mm.

Motors

We considered using a standard DC motor or a Servo. After a lot of consideration we decided to go for the Servo (See appendix C). One of our prime concerns was to make the robot as light as possible, even though this is not a design requirement. Keeping this in mind, choosing servos was the best alternative because they are extremely light and small as compared to a DC motor. In adherence to our philosophy we picked the servos because they are very easy to control. It comes with a built in control circuitry, unlike the DC motor for which you need an H bridge (See Appendix D). The operation of servos can be found in Appendix C. Most servos are not fully rotational see Appendix C, to make them fully rotational we have to "hack" or modify them. This can be tricky and needs some time and patience. It is also different depending on which brand and even model of servo you are using. It took us considerable time as this was our first experience in hacking and we had to deal with miniature parts, so extra precautions were needed.

Power

Power was a critical part of our design, because we had to make sure that we were not overloading or supplying too little power to any component. Either case would have been dangerous to the components. To keep each major component independent of the other, we have three sets of power supplies: one for the OOPIC rated at 5V, the second for the servos rated at 10V, and the third for the extinguishing motor at 18V. We have 18 AA batteries on the robot. Six powering the OOPIC and another set of 12 for the servos. We are using two 9V batteries for the extinguishing motor. This division enabled us to run one component without worrying about its effect on the other. To keep costs within budget and to conserve we decided to use rechargeable batteries. We are using the 15 minute rechargeable AA NimH batteries. This saves us a lot of time. We can charge all 18 of them in less than 45 minutes. Since we rotate in other batteries, we are able to keep them constantly charged during testing. We used two 9V batteries in series for the extinguishing motor as it only turns on for a little while and it pulls a lot of current. This may have loaded down the supply too much for other more important components to also be attached.

Extinguishing Circuit

Initially we decided to use two CPU power case fans to blow out the candle. It seemed to be a good idea because it was simple and we could use the power rails on the OOPIC. However, after installing them we found they were not powerful enough to blow out the candle instantly. After some research we decided to use a DC motor to blow out the candle. We pulled out a DC motor from a remote control car, and the blades from a CPU case fan we used previously and modified them suitably. We had to use a switching circuit to drive the motor from another power supply, because the fan draws a lot of current now that it is attached to a DC motor (See Appendix H).

Initial Problems

There were several problems initially:

1. Chassis Design and Fabrication – It was difficult to make the chassis round using the tools in the machine shop.
2. Hacking the Servos – The gears were so small and easy to break.
3. Wheels – The wheels were thin, so the robot would have had trouble traveling in a straight line. Cost was also a major concern when buying wheels. The wheels were $25/piece.
4. Tires - The tires were too thin and flimsy. They did not grip the surface well, and broke easily.
5. Casters – Originally, we used a regular wheel caster (like one on a desk chair) which caused problems when starting from a full stop. The high torque in the motors causes a caster like this to rotate, which turns the robot so it is no longer aligned correctly.
6. Mounting the sonar sensors – The top deck of the robot interfered with the sonar, giving incorrect readings.
7. Flame Sensor – Not able to hone in on the flame of the candle.

Solutions

1. Chassis Design and Fabrication –We used a band-saw to roughly cut out the chassis into a round shape, and then used a file to refine the edges. We wasted some plexiglass in the process.
2. Hacking the Servos – The servo gears were small and there were two small protrusions that had to be removed to make them fully rotational. Extra care had to be taken to etch out the protrusions, because it would have damaged the gears. One tooth on a gear was slightly damaged but eventually we were able to notch out the damaged gear.

3. Wheels – We bought four cheaper wheels and then put two of them together using a one inch plexiglass spacer. This gave us a wider wheel.
4. Tires – After the original tires broke, we bought vacuum cleaner belts to replace them. These were much stronger, and were longer lasting. They also provided a good grip on the floor surface.
5. Casters –We managed to get casters which were like ball bearings, so they would not turn the robot in an undesired direction when starting or turning.
6. Mounting the sonar sensors – The sonar sensors were initially mounted directly on the top deck of the robot, which interfered with its functionality. The sound wave would hit the chassis and give incorrect readings, this interfered with the control algorithm. We resolved this by lowering the sonar sensor using a bracket.
7. Flame Sensor – Built housing for the flame sensor and painted it with flat black paint all over. This made it opaque. We have a small opening where the actual sensor is, so it only triggers when it is facing the flame.

Sensors (Jaime Austin)

Jaime Austin was responsible for the selection or building of all of the sensors used on the firefighting robot. In order for the firefighting robot to interact with its environment, several different sensors are necessary. Distance sensors are needed so the robot can properly navigate through the floor plan without touching the walls. A sensor to detect a flame is used to locate the flame so that it can be extinguished. White line sensors are used identify white surfaces in the floor plan. For an advantage in points, a tone decoder is used to start the robot with a 3.5 kHz buzzer that simulates a fire alarm.

Distance Sensor

Sensors for measuring distance are necessary for the robots navigation through the floor plan in order to identify its location and to avoid hitting the walls of the floor plan. Using three sensors (front, left, and right), the robot can get an accurate depiction of its surroundings. Two different sensors were considered for measuring distances in the floor plan: the Sharp GP2D12, and the Devantech SRF04 Ultra-Sonic Ranger. The Sharp sensor has a range of 10 cm to 80 cm and the distance is indicated by an analog voltage. The Devantech sensor has a range of 3 cm to 3 m and the distance is indicated by the width of an echo pulse. The Sharp sensor provides continuous readings while the Devantech must be pulsed to take a reading [2].

The Devantech SRF04 Ranger is the sensor that we used in the firefighting robot. The Devantech ranger was chosen for two main reasons:

1. The Devantech has a wider detection range, especially at the near end.
2. Interfacing with a microcontroller is simpler because an analog to digital converter is not needed.

The Devantech Ranger is operated by providing a 10 us pulse to the trigger line and reading a pulse from the echo line. The width of the pulse on the echo line is proportional to the distance being measured. Using an OOPic for the microcontroller simplified operation of the Devantech Ranger because the OOPic already has an object to control the Ranger. Another advantage of the Devantech sensor is that it uses sonar instead of IR which is more prone to external interference due to ambient lighting. More technical data on the Devantech SRF04 Ranger can be found in Appendix G.

Flame Sensor

The flame sensor is a very important sensor for the firefighting robot.   This sensor is used to locate the flame so that it can be extinguished, which is the ultimate goal of our robot. The sensor that we used for this is the Hammamatsu UVTron Flame Detector. The intensity of the flame is indicated by the frequency of pulses that are continuously read from the UVTron. The UVTron is capable of identifying a flame that is 5 m away in a sunny room [2]. Since this is a likely situation for the robot, the UVTron was an obvious choice. The UVTron is also advantageous because the OOPic has a predefined object for this sensor.

Due to the high sensitivity of the UVTron, a flame can be detected from almost any direction. In order to identify the general direction of the flame, a box was constructed around the UVTron to filter out the light coming from the periphery. The box has an opening directly in front of the bulb that senses the flame. By adding this box, the robot only sees the flame when it is directly in front of the robot. This enables the robot to use the UVTron to identify the flame and then to locate and extinguish the flame. Additional technical data on the Hammamatsu UVTron can be found in Appendix G.

White Line Sensor

Line detectors are used to identify the various parts of the floor plan that are indicated by a white line or circle. The doorways of the rooms are marked by a white line and the candle is surrounded by a white circle. By using two sensors (right and left) the robot is able not only able to identify the white areas of the floor plan, but to align itself. This gives the robot the ability to ensure that it is headed straight into or out of a room or that it is pointed at the candle when it is being extinguished. This sensor was implemented using an Optek PB745 reflective object sensor which consists of an infrared LED and phototransistor Darlington pair. The reflective object sensor has been integrated into a comparator circuit to provide a high output when it is over a white surface and low output when over a black surface. The reference voltage is adjusted by changing a potentiometer [3]. The circuit diagram for this sensor and technical data for the reflective object sensor are located in Appendix G.

Two of theses sensors were built and mounted on to the bottom of the robot.   The ideal distance for the sensors is about 1/8 in from the floor. The sensors can be calibrated by adjusting both the distance from the floor and the potentiometer. This circuit is available pre-built from Lynxmotion, but was built by hand due to its simplicity and to save money. As can be seen in the budget, we only spent $12.00, and buying the circuit pre-built would have cost $30.00. This gave us a savings of $18.00 for only 2 hours of work.

Tone Decoder

A bonus is received in competition if the robot is started using a 3.5 kHz tone. This was implemented by integrating a tone decoder with our start button so that either option could be used to start the robot. A National Semiconductor LM567 tone decoder is used to identify the 3.5 kHz tone. The frequency that the tone decoder activates on is set by external resistors and capacitors. The signal is input into the tone decoder using an electret condenser microphone. The 3.5 kHz tone is produced by a piezo buzzer hooked to a 9 V battery through a switch. The starter circuit, which incorporates the tone decoder with the start button, and technical data on the LM567 is located in Appendix G.

Firmware and Integration (John Porterfield)

John Porterfield led the firmware and integration issues on the firebot. This involved a general knowledge of the issue of coding everything onto the microprocessor, integrating all of the hardware with the microprocessor, and research involving which microprocessor to use.

Microprocessor choice – The OOPic

There were many choices for the microprocessor of our robot, and we had a hard time initially deciding which to choose. It seemed after an initial assessment of the previous winners of the contest, that the most popular microprocessors to use were the basic stamp processor from parallax, and the OOPic microprocessor, which is a newcomer to the robotics world, and does not yet have much documentation.

The key factors in deciding to use the OOPic rather than another comparable microcontroller were:

1. Less Hardware – A compatible microcontroller board kit was also being sold which would simplify connections in design,
2. More choices for programming language – We could choose to program in either C++, Basic, or Java, because the compiler supported all of these,
3. Rapid Prototyping – We felt that the OOPic would allow for more rapid development of the robot because pre-written objects were available to interface all components with the OOPic quickly. More specifically, objects were already written to provide a building block for control of the servos, the Devantech Sonar, and the Hamamatsu UVTron sensor.

The reason that we did not choose the Basic Stamp microcontroller was because we felt much more confident programming a PIC type microcontroller due to our experience programming in Senior Design I. Although three of us had experience using the Basic Stamp as well, one of us had not ever programmed this microcontroller and did not feel that they could contribute to the firmware section at all if we did not use the OOPic.

Firmware

In order to subdivide the coding responsibility between team members, we have split the issue of firmware into two separate sections.

1. Traversing the floor plan, and
2. Extinguishing the flame.

Traversing the floor plan can be further defined as all code which will allow the robot to travel to each room in the maze and point toward the room. Once it has pointed toward the room, the flame extinguishing half of the code takes over.

The flame extinguishing code allows the robot to slowly move forward until it is lined up with the doorway line, and then move into the room and search for the flame. Once it is done looking for the flame, it then either finds it and extinguishes it, or lines back up with the doorway line pointing out of the room so that the floor plan traversing algorithm can then continue.

Floor Plan

The floor plan is fixed in the division that we designed our robot to run in (See Senior Division in Appendix A). This floor plan is fixed, but was warned to perhaps be a few cm or mm off of the specification. Only the Expert division of the competition has a floor plan which is not set beforehand. In order to simplify expression of which room we were in front of, we assigned a letter to each room in a counter-clockwise fashion starting from the top right when the robot is oriented pointing toward the two smaller rooms from the home circle. Our robot travels to each room in this order, to avoid visiting two rooms twice. We noticed in the rules that just passing by the doorway of a room counts as searching it. This means that searching the two larger rooms first would be illegal, because we would have to backtrack past room C twice in order to search the two smaller rooms. This would disqualify our robot because it would have searched the same room twice. (See Appendix A)

Logic of traversing the floor plan

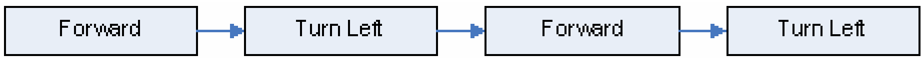
Simplifying a logic flow is the most efficient way to write any structure of code which must accomplish a large task by managing many smaller tasks. In order to simplify the logic, we have to divide the code into separate methods which all work independently of each other. This prevents smaller tasks from interfering with each other because each module works based only on preconditions and postconditions for the code.

In order to traverse the floor plan, we wrote separate methods for the following particular actions:

1. Moving forward until a stop condition (a distance) was met on one or more of the sonar devices,
2. Turning Left, and
3. Turning Right.

Because each of these modules were written separately and worked independently, we are able re-use code when writing the firmware. This allows for a more efficient debug process because it is easy to find and fix errors in logic, and it also allows more complex algorithms within a small space because there is no need to write the same snippet of code twice anywhere.

The pseudo code is simplified by this modularity to look like this:




```
// Begin

Move forward until sensors read “Left > 100”, “Front < 64”, and “Right
> 0”
Turn left for 50 cycles
Move forward until sensors read “Left > 100”, “Front < 100”, and “Right
> 0”
Turn left for 50 cycles

// We are now in front of room A.
```

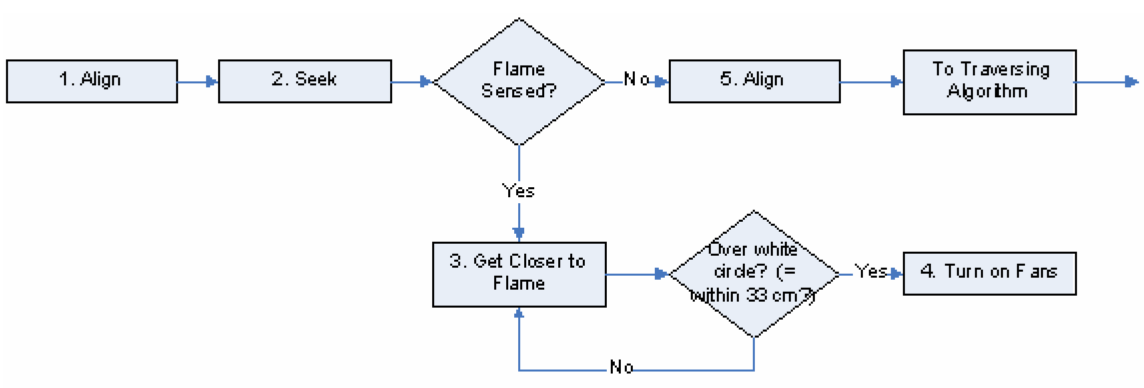
In both of the forward algorithms, there are different conditions for exiting the forward method, but the same forward method is used because it is not necessary to write an entirely new control algorithm for the different sections. Likewise, the turns are uniform, although imperfections in the maze can be corrected for by “tuning” each individual turn for maximum performance.

The final code implemented for the OOPic microcontroller has the advantage of being very readable because the actions are so modularized that the final code looks a lot like the pseudo code. This is self-evident from the code, which can be found in Appendix J.

Extinguishing the flame

Method

The steps that our robot goes through in the second half of the code are:



1. Move forward until wheels are aligned with doorway line,
2. Move into the room a distance where it is possible to see the entire room, and sweep the flame sensor toward the furthest wall until rotated 180 degrees (to point out of the room),
3. If the flame is found, get closer to it, until over the white candle circle, and
4. Extinguish.
5. Otherwise, move forward until wheels are aligned with the doorway line.

Initial Problems

There were 3 main problems with the extinguishing algorithm after the first implementation of the code. These problems were

1. Lack of Control – the robot had no control algorithm implemented within the rooms in order to keep from hitting the walls while searching for the flame.
2. Fans not effective – the fans we originally had were not sufficient to put out the flame on the first try, and the robot made no effort to get closer to the flame, or check to make sure it was being effective.
3. UVTron Flame sensor not effective – the UVTron sensor originally triggered a maximum reading of ultraviolet light when it was anywhere near the room with the candle in it. It was not necessary to be near the candle or even pointed toward it for a maximum reading to occur, so there was no way to tell if the robot was moving toward the candle or away from it, making it hard to determine the location of the flame.

Solutions

The solutions to the problems will be discussed in order:

1. Lack of Control – The robot has been given an algorithm which allows it to determine if it is too close to the wall using the sonar sensors. If it determines that it is too close to the wall, it turns away from the wall, and moves outward to attempt its approach from a more optimal direction. This allows for a better success rate when attempting to extinguish a candle which is placed very close to the intersection of two walls.
2. Fans not effective – In addition to replacing the two case fans with only one, more powerful fan, we also changed the extinguishing algorithm. Originally, the extinguishing of the flame was dependent on how well the control oriented the fan toward the flame. If the fan was not pointed directly at the flame, for instance, the indirect blast of air was not sufficient to extinguish the flame. To solve this problem, we made the fan move back and forth while trying to extinguish the flame. Because our chassis can make a zero radius turn, it is possible to move the fan back and forth without hitting the wall. The fans continue blowing on the candle and sweeping back and forth until a timer reaches a certain limit. Once this limit is reached, if the candle has not yet been extinguished, the robot moves forward and attempts the extinguishing again.
3. UVTron Flame sensor not effective – Because the UVTron Flame sensor did not determine the position of the flame, we built a casing for the sensor to filter out any light which was reflected off of the walls and only focus on making sure that the flame was directly in front of us. The casing is discussed in the mechanical

section of our report. This new casing allowed us to receive a range of values based on how close our robot was to pointing directly at the flame, effectively solving our problem of location.

Interfacing components

Integration was also the responsibility of John Porterfield. Although this is a very broad category, it can be narrowed down to the relation of hardware other than the wheels (because this involves control) that needs software (firmware) in order to operate correctly. This includes correct operation of all sensors.

In order to receive a signal from all the components, we made wires to connect each sensor to a port on the OOPic. All wires between the OOPic and the sensors are handmade in order to be closer to the correct length for each particular sensor, and are braided together to allow for quick debugging. (For example, if a wire comes loose it is easy to see that this wire affects a certain sensor).

The components which are connected to the OOPic that required firmware to be written are as follows:

1. White Line Sensors
2. Sonar Sensors
3. UVTron Sensor
4. Microphone Sensor and Start Button
5. Debug LCD Screen
6. Fans

White Line Sensors

The white line sensors are integrated into the OOPic by making a 4 bit input/output line to 4 pins on the OOPic. The sensors were designed to give a high logic signal if the white line (or any other reflective surface) is sensed, and keep a low logic signal if they do not sense a white line. These 4 bits create a number which is read by the microprocessor, and decisions for which way to turn to align with the line are made using the value of this number.

Sonar Sensors

The sonar sensors are integrated into the OOPic by reading the number of time divisions that happen between when the ping signal is sent from the sonar and when it is echoed back to the OOPic. These values are directly proportional to the distance that the sensor is from an object, so these values are used when making decisions to avoid walls, or align walls.

UVTron Sensor

The UVTron sensor takes a continuous time reading within finite time periods. Because a more intense flame will make the sensor fire more rapidly, a more intense flame (due to pointing more directly at it), will cause a higher number to be read by the OOPic. The magnitude of this number is proportional to the angle of the robot. (More direct orientation means higher numbers). These numbers are used to orient the robot toward the flame.

Microphone Sensor and Start Button

The microphone sensor is coupled on the same one bit line as the start button. When either the signal is sensed or the start button is pressed, it sends a high signal to the one bit line on the OOPic. When the OOPic is turned on, it immediately begins an infinite loop in the code to wait for this signal to start.

Debug LCD Screen

An LCD Screen was originally added to the design of the robot, to allow for a more efficient debug process. Having an LCD screen attached to the robot allowed us to make changes and discover errors onsite (while running the floor plan) instead of requiring a computer to find errors. There were some problems with keeping the LCD after most of the code was written, but it did serve its purpose of giving us information during the integration of the different pieces of hardware.

Fan

The fan was implemented using a simple 1 bit output line which sent a high signal to switch on the fan's power supply.

Looking back – The OOPic in review

The consensus in our group now that the project is completed is that choosing the OOPic in the first place was perhaps not the best decision for good development. There are many small reasons for this consensus, but they can mostly be summed up by one statement.

Because the OOPic is so new, and there is not much support for the microcontroller, not everything works as you expect it to.

On the internet, the OOPic looked like a novel concept because it simplified programming the normally linear microcontroller code into a modular, object oriented structure. However, the "object loop" presented in the documentation for the OOPic does not work as indicated. We were under the assumption that any robust microprocessor would not skip instructions based on how many instructions there were. However, the OOPic skips instructions when there are too many objects in the object list loop. This means that the more components you add and must control using the OOPic, the less likely the OOPic is to perform what you ask it to.

This phenomenon was most apparent on the LCD display. When the two servos and the sonars and one other object was plugged in to the OOPic and being controlled, anything output to the LCD would skip every other character. If more than three other objects were attached to the OOPic, then the LCD would only print every third character.

We can only assume that this means that every third sonar reading, or every third servo pulse was also skipping during execution, providing results that could not easily be interpreted. To fix this, we had to take out the LCD for the final testing of logic for the robot. A sacrifice which proved to take up much more time than necessary with any other microcontroller. The final implementation of the robot will also not have an LCD screen to show any type of information that the robot senses that would further explain its behavior.

Furthermore, the OOPic company is not very honest in its production and sale of the OOPic. The "kit form" of the OOPic which we bought came with extremely poor instructions which did not even include a layout, but only a schematic of the OOPic board. Parts were missing from the kit which the sales representative said were not needed, but we later found out were absolutely necessary for proper operation of an entire 8 bit port on our OOPic. After repeated attempts at contacting the company, we were not successful in obtaining the parts, which were not readily available at any parts store here in Stillwater.

Control (Shoaib Afandi)

One of the requirements stated in the Trinity College website is that the robot should travel the floor plan without touching the walls. Also, the robot needs to be autonomous, so in order to make it more intelligent we needed more control. For this reason we implemented a two-wall following algorithm and one-wall following algorithm.

One-Wall Following

The One-Wall Following algorithm is used when there is only one wall on either side of the hallway.

While there is a wall only on the right hand side and no wall on the left hand side the right wall following algorithm is executed as follows:

1. If the right sonar value is less than the defined reference (optimal distance the robot must be from wall) then
2. Move away from the wall. Otherwise,
3. Move towards the wall.

Pseudo Code

```
Right Wall Following Function ()
      While (No Wall on Left) AND (A Wall on Right)
            If (Right Sonar Value is Less Than Defined Reference) Then
                  Turn Left ()
            Else If (Right Sonar Value is Greater Than Defined
Reference) Then
                  Turn Right ()
      End While Loop
End Function
```

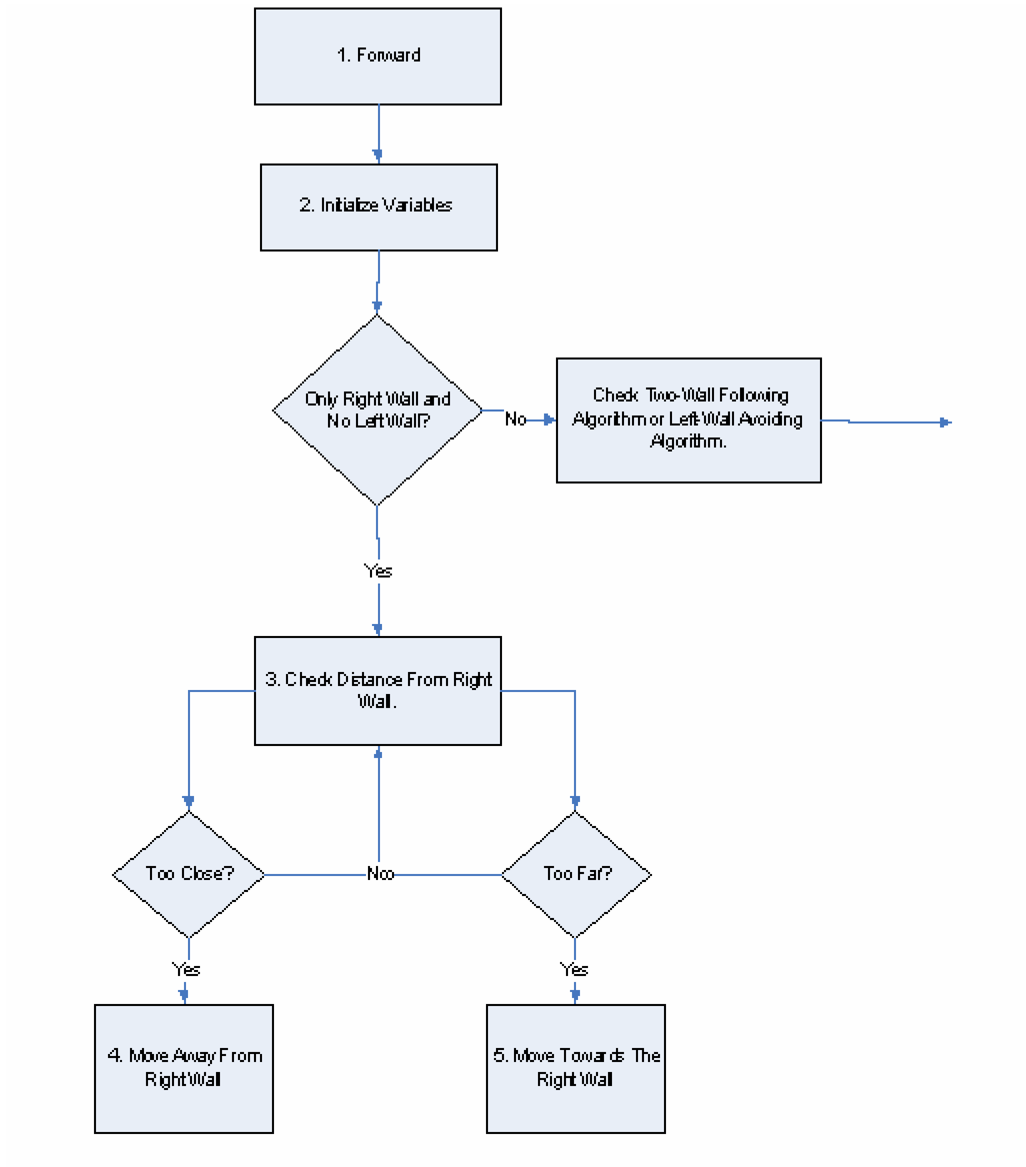


Figure: Right Wall Following Flowchart.

In the floor plan there are only a couple of small patches where the robot sees only the left wall and this occurs for a very short time. So instead of implementing an algorithm that is similar to the right-wall following, the left-wall algorithm only avoids the left wall.

While there is no wall on the right hand side and there is a wall on the left hand side the algorithm executes as follows:

1. Avoid the left wall by moving away.

Pseudo Code

```
Left Wall Avoiding Function
   While (No Wall on Right) AND (A Wall on Left)
               Avoid Wall ()
   End While Loop
   End Function
```

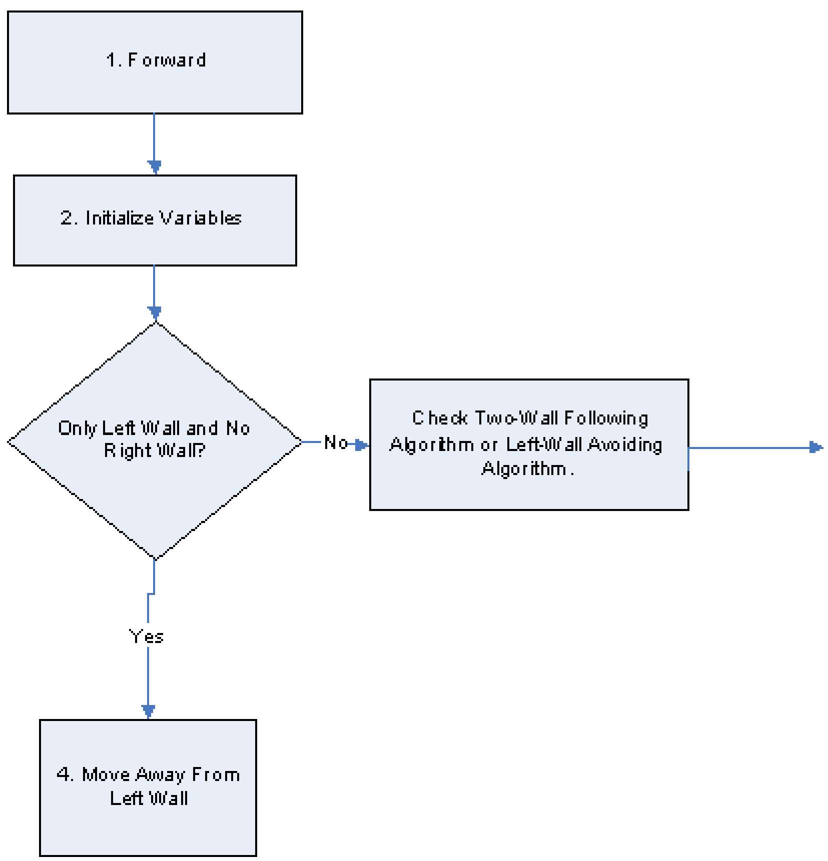


Figure: Left Wall Avoiding Flowchart.

Two-Wall Following

The one-wall following algorithm alone is not enough to control the robot for the whole floor plan. More complicated control is necessary to make sure the robot does not touch the walls and stays aligned with respect to the two walls. For this reason we used an algorithm which keeps the robot aligned with respect to the two walls. This algorithm involves a proportional control (See Appendix I) to keep the robot aligned, and works as follows:

If the right sonar value is greater than the last right sonar value then it means that the robot is moving away from the right wall and close to the left wall. So we turn to correct the alignment proportional to the amount the robot is misaligned. Similarly, if the left sonar value is greater than the last left sonar value it means the robot is moving away from the left wall. Here, we use proportional control again to align with respect to the walls by turning left.

We know that the robot is aligned with respect to the two walls by placing the robot anywhere between the two walls and reading the values from the left sonar sensor and the right sonar sensor. The sum of those values is used as a constant align value which indicates that the robot is aligned with respect to the walls. There is no other possible orientation that the robot could be placed at which would yeild this unique sum of the two sonar sensors. This constant value is then subtracted from sum of the right sonar value and left sonar value. If the absolute difference is zero the robot is aligned and should move straight ahead. Otherwise, if the absolute difference is not zero the robot is not aligned correctly and we multiply this absolute difference with a constant which moves the value into a usable range (determined by trial and error) and subtracted from the servo values.

Pseudo Code

```
Two Wall Following
      While (Wall on the Left) AND (Wall on the Right)
            If (Right Sonar Value is Greater Than Last Right Sonar 
Value)
                  Proportional Control ()
            Else If (Left Sonar Value is Greater Than Last Left Sonar 
Value)
                  Proportional Control ()
            Else
                  Go Straight ()
      End While Loop
End Function
```

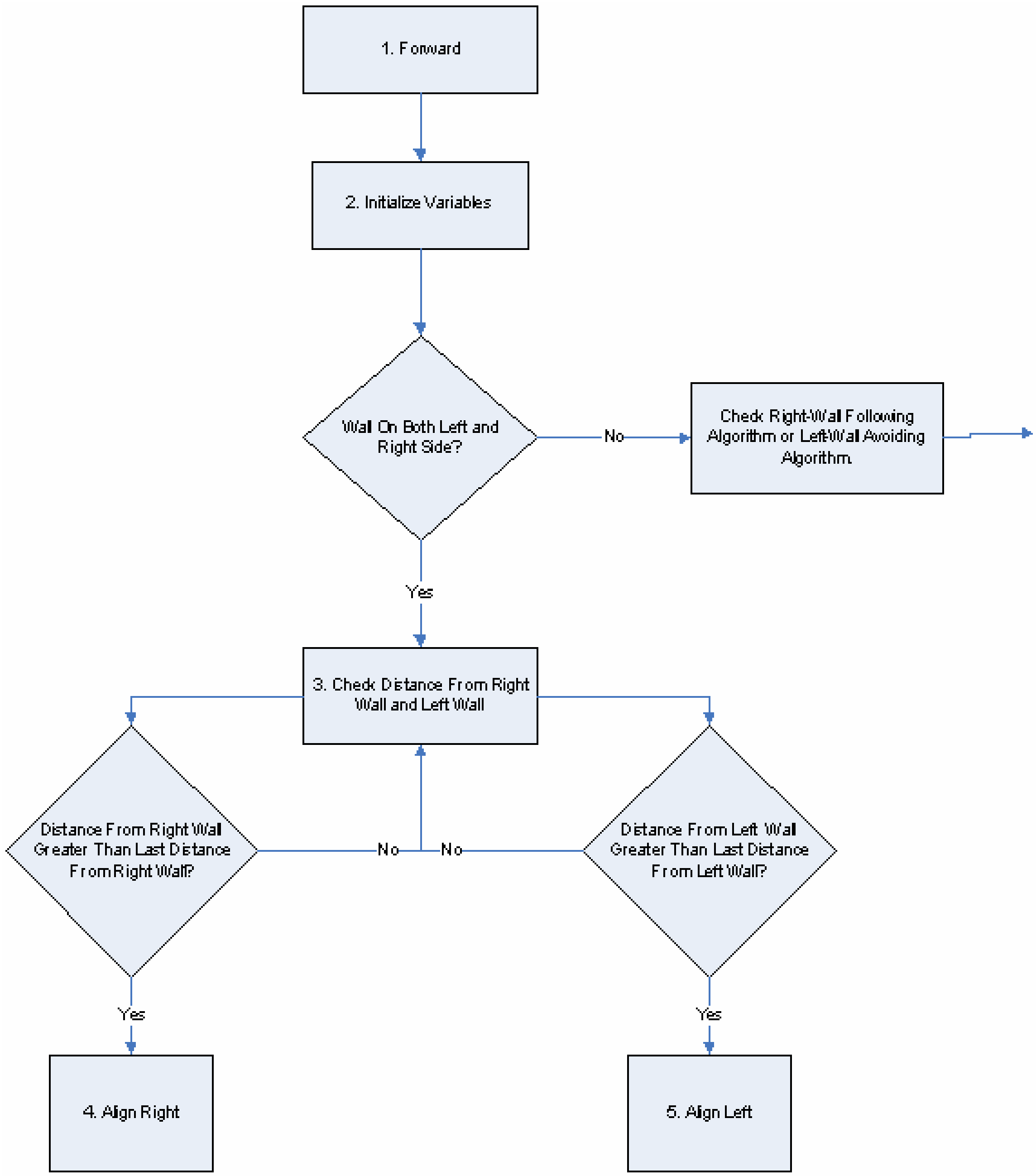


Figure: Two-Wall Following Flowchart.

Problems

1. At first, we used proportional-derivative control for both one-wall and two-wall following but that did not give us the amount of control we desired.
2. The sonar sensors timed out sometimes when we were testing the control algorithm. This caused the robot not respond to the code properly.

3. The sonar sensors also gave us a huge value when the robot was too close to the walls, because they violated the constraints for operation (must read 3cm – 3m) causing the robot not to respond to the code properly.

Solutions

1. Instead of using a complicated proportional-derivative control we used simple proportional control for the two-wall following and small left and right corrections for the one-wall following algorithm.
2. The reason the sonar sensors timed out sometimes was because the wiring was not reliable. We fixed that problem by making better wiring.
3. We also moved the sonar sensors a little inward on the robot, and it gave us reasonable values

Conclusions

We consider our project to be a success. We successfully demonstrated it publicly twice. The first time was in Tulsa, where our robot ran the course without violating any rules at least 5 times (once in each room). When demonstrating it during the public demonstration for our class, our robot exhibited that it could extinguish the candle at least 2 out of 3 times, which meets all of our specifications.

In addition to meeting our specifications, we also demonstrated the robot both times with a "smoke alarm starter" which would have given our score a bonus if we had been able to attend the competition. Every time our robot was demonstrated publicly, it was started using this starter, and never failed to start when the alarm was sounded.

We had fun participating in the firefighting robot project, and hope you enjoyed hearing about how we accomplished our goals.

References

[1] "Fire Fighting Robot Rules and Regulations", [PDF Online Document], Available HTTP: http://www.trincoll.edu/events/robot/Rules/Rules2004.pdf

[2] Acroname, "Sensors." [online] 2003, [2004 January 23], Available HTTP: http://www.acroname.com/robotics/parts/c_Sensors.html

[3] Lynxmotion, Inc, "Single Line Detector," [online] 2003, [2004 February 4], Available HTTP: http://www.lynxmotion.com/Product.aspx?productID=58&CategoryID=8

[4] "BOT40-II BotBoard Page" , [Online Document], Available HTTP: http://www.oricomtech.com/bot40-2.htm

Appendix A

Official Rules and Regulations

2004 RULES AND REGULATIONS

Saturday April 17 & Sunday April 18, 2004

(Thanks to all those who gave advice and suggestions. Even if we didn't use them, we appreciated getting the feedback. Special thanks to David Shilling for his editing help.)

1. CONTEST OBJECTIVE

The specific challenge of this contest is to build a computer controlled Robot that can move through a model floor plan structure of a house, find a lit candle and then extinguish it in the shortest time subject to a few operating factors (see Scoring Procedure, section #23). This is meant to simulate the real-world operation of a Robot performing a fire protection function in an actual home. The candle represents a fire which has started in the home which the Robot must first find and then extinguish. The goal of the contest is to advance Robotics technology and knowledge while using Robotics as an educational tool to enhance students' learning.

2. DIMENSIONS AND SPECIFICATIONS

This is a contest that simulates real world activities and capabilities. As such, there are many areas of uncertainty that a Robot must be able to handle in order to succeed. The goal of the contest is not to make a Robot that can perform only in the laboratory, but carry out its function in the real world with all the variations and problems that exist there. Therefore, all the dimensions and specifications listed in these rules are given as a general aid to the participants. None of them are meant to be exact and they will vary somewhat at the actual contest. Any Robot that is meant to function in the real world needs to be able to handle these uncertainties. This contest, like the real world, is imprecise and your Robot needs to be able to handle that.

3. AWARD DIVISIONS

JUNIOR - This division is for students in grade 8 and below. All the rules and Operating Mode (see section #18) options apply to this division, except that the candle will only be placed in one of the two big rooms (the lower two rooms on the arena map - see Attachment A). The two upper small rooms will not be used and need not be searched.

HIGH SCHOOL - This division is for students in high school grades 9 through 13. All the rules and Operating Mode (see section #18) options apply to this division and the candle can be in any of the 4 rooms.

SENIOR - This division is for anyone out of high school, that is, college students and adult contestants. All the rules and Operating Mode (see section #18) options apply to this division and the candle can be in any of the 4 rooms.

WALKING - This division is for any Robot that uses only legs to move around. The number of legs does not matter. The size restriction on the Robot is slightly changed to allow Robots that are a maximum of 46 cm long, 31 cm wide and 31 cm high. All the

rules and Operating Mode (see section #18) options apply to this division, except that
the candle will only be placed in one of the two big rooms (the lower two rooms on the
arena map - see Attachment A). The two upper small rooms will not be used and need
not be searched.

EXPERT - This is the most advanced division and as such it has a number of modifications to the rules and changes to the arena. This division is open to anyone willing to take the challenge. For specific information, see section #25**.**

4. THE HOUSE FLOOR PLAN STRUCTURE AND FEATURES

The size and shape of the arena is the same as in previous years. The dimensions are
given in metric units in keeping with the international scope of the contest and with scientific and technical usage.

The official floor plan structure contest area for the Junior, High School, Senior and Walking divisions is shown in Attachment A. The Expert division will have a different arena and differe nt rules also. See the Expert Division (Section #25) for more information. For the Junior, High School, Senior and Walking divisions the design of the
structure will be known beforehand (see Attachment A) and the goal is to find and extinguish the candle as quickly and reliably as possible.

The walls of the structure will be made of wood and will be 33 cm. high. The walls will be
painted with flat white latex paint. The floor of the arena will be a smooth wood surface
painted with flat black latex paint. Any seams in the floor will be taped over and painted
with the same flat black latex paint. The seams in the floor section may not be perfectly
flat however. Make sure that your Robot can handle a discontinuity of up to 5 mm.

As noted in the Attachment A floor plan, all hallways and doorways to room will be 46
cm wide. There will not be a door in the doorways, just a 46 cm opening. There will be a
white 2.5 cm wide line made with white tape across each doorway to indicate the entrance to each room.

The floor of the arena will be painted black, but some Robots may use foam, powder or
other substances to attack the candle flame. Our best efforts will be made to clean up
after each Robot, but there is no guarantee that the floor will stay uniformly black throughout the entire contest. The floor may also have small (3 mm diameter) red or blue dots on it to indicate the potential locations of candles and furniture. Unless you are
operating in the Uneven Floor mode (see Operating Modes, section #18), the floor will
be level with no ramps or stairs.

The Robot will start at the Home Circle location marked by the H in a circle on the arena
floor plan (see Attachment A). The actual Home Circle will be a solid white circle (without the H) on the floor. The 30 cm diameter white Home Circle will be in the center

of the 46 cm hallway. Notice that on the arena floor plan in Attachment A there is no gap in the outer wall of the arena behind the Home Circle. This part of the wall can be removed to allow contestants easier access to their Robots in setting them up, but the wall can be replaced in this section if it helps the Robot's operation. Robots may also use any placement fixtures if they help the Robot initially align itself in the Home Circle. The Robot must start within the Home Circle, but once started, it can go in any direction desired.

5. AMBIENT LIGHTING

Contestants will be given time on the contest days to take ambient light level readings to calibrate their Robot. Once set on Saturday, the lighting in the contest area will not be changed to suit individual competitors. Part of the challenge of the contest is to make a Robot that can operate in real world situations and that includes inconsistent lighting, shadows, glare, etc.

6. ROBOT OPERATION

Once turned on, the Robot must be autonomous --self-controlled, without any human intervention. That is, they are to be computer controlled and not manually controlled devices.

The Robot can bump into or touch the walls of the arena as it travels, but it cannot mark or damage the walls in doing so. There will be a penalty for touching a wall. (See Penalties, section #20) The Robot cannot leave anything behind as it travels through the arena. It cannot make any marks on the floor of the arena that aid in navigation as it travels. Any Robot that deliberately, in the judges' opinion, damages the contest arena (including the walls) will be disqualified. This does not include any accidental marks or scratches made in moving around.

The Robot must, in the opinion of the judges, have found the candle before it attempts to put it out. For example, the Robot cannot just flood the arena structure with CO2 thereby putting the candle out by accident.

7. PUTTING OUT THE CANDLE

The Robot must not use any destructive or dangerous methods to put out the candle. It can use such items as water, air, CO2, Halon, etc., but any method or material that is dangerous or will damage the arena is prohibited.

It will be permissible to put out the candle by blowing air on it. Although this is not a very practical method of extinguishing a fire in the real world, it will be allowed in this contest since the goal of the contest is the advancement of Robotics and not necessarily that of firefighting techniques.

The candle cannot be touched by the Robot itself while it is still lit. (See Penalties, section #20)
The Robot must come within 30 cm of the candle before it attempts to extinguish the flame. There will be a white 30 cm radius solid circle (or circle segment, if a wall is in the way) on the floor around the candle and the Robot must have some part of its body over the circle before it puts out the candle. The candle circle will be made of thin (0.5 mm) poster board and the candle will be placed in the center of the circle.

8. ROBOT SIZE

Robot must be able to fit in a box 31 cm long by 31 cm wide by 31 cm high. If the Robot has feelers to sense an object or wall, the feelers will be counted as part of the Robot's total dimensions. The Robot cannot separate into multiple parts and must never extend itself beyond the 31 cm allowed.
If contestants want to add a flag, hat or other purely decorative, non-functional items to the Robot, they may do so as long as the item has absolutely no effect on the operation of the Robot.
As noted previously, the Robots in the Walking Division can be up to 46 cm long.

9. ROBOT WEIGHT

There are no restrictions on the weight of the Robot.

10.ROBOT CONSTRUCTION MATERIALS

There are no restrictions on the types of materials used in the construction of the Robot.

11.THE CANDLE

The lit candle is supposed to represent a small house fire that the Robot is attempting to find and put out. The candle flame will be between approximately 15 to 20 cm from the floor. This height includes the height of the wooden candle support base. The candle used will be a standard approximately 2.5 cm thick white candle. The exact height and size of the flame is unknown and variable and will be determined by the specific conditions of candle and its surroundings. The Robot is required to find the candle no matter what the size of the flame is at that particular moment.
The candle will be placed at random in one of the rooms in the arena. The candle has an equal chance of being in any of the 4 rooms in each of the 3 trials that a Robot has. Hopefully the candle will be placed in different rooms in each trial to best test the Robot's operation, but it might be possible for the candle to be in the same room twice.
If it happens that the candle is placed in the same room for both the 1st and 2nd trials, then we will make sure that it is not in that room for the 3rd and last trial. Thus every Robot will have the candle in at least 2 rooms and possibly 3, during its 3 trials. (The Junior and Walking Divisions are an exception to this rule since they only use 2 rooms.

In the Junior and Walking Divisions the candle will be in one room twice and the other
room once during the 3 trials.)
The candle will not be placed in a hallway, but it might be placed just inside a doorway
of a room. The candle circle will not touch the doorway line and this means that the front of the Robot will be able to move at least 33 cm into the room before it encounters
the candle.
The contestants cannot measure or touch the candle before it is used. Violation will result in immediate disqualification of the team and the robot.
The candle will be mounted on a small wooden base painted semi-gloss yellow. This base is used to help keep the candle from tipping over easily, but it will be possible to
knock the candle over by bumping into it (which you don't want to do - see Penalties, section #20).

12.SENSORS

There is no restriction on the type of sensors that can be used as long as they do not violate any of the other rules or regulations.
Contestants are not allowed to place any markers, beacons or reflectors on the walls or
floors to aid in the Robot's navigation.
Robot builders should be aware that many cameras transmit infrared light as part of their automatic focusing systems. Ambient lighting in the contest room may also be a
source of IR, visible and UV light. During the course of the contest, sunlight may come
into the contest room through open outside doors. The sunlight will not shine directly on
the arenas, but may be detectable by very sensitive sensors. During the course of the
contest, judges at other arenas may be lighting candles or lighters. These incidental flames will be above the arena and further away than the candle, but still may be detectable by an undiscriminating sensor. In setting up the arena, contest officials may
put their arms into the arena and some very sensitive sensors may mistake that IR emission as the flame. If a Robot uses light sensors to find the candle or detect walls or
furniture, it is the Robot builder's responsibility to design their Robot to prevent these
and other unintended UV, visible and IR sources from interfering with its operation. Part
of the challenge of this contest is to design a Robot that can find the candle flame and
ignore everything else.

13.ELECTRICITY

The maximum electrical requirements for any system needing electricity at the arena will be 10 amps at 120 VAC.

14.CABLES

If the Robot is connected to an external computer system for instructions and/or power
make sure that the cable is long enough for the Robot to get to all areas of the arena. If

a contestant wants to hold the cable above the walls while the Robot runs, they c an, but
if during the trial, in the opinion of the judges, they use the cable to assist the Robot, then that trial will be ended with no score.

15. THE ORDER OF RUNNING

The Robots will be assigned numbers to determine the order in which they will compete
in the contest. Each Robot will make a trial run in the arena in the order in which it is assigned. The Robots will compete consecutively and when everyone is done with their
first attempt the whole process will repeat for the second and third attempts.
Contestants will have time between their trials to make any adjustments, modifications
or repairs to their Robot, but once the Robot before them has completed its trial, then
they will have 1 minute to get their Robot in the arena and started on its trial. There will
be a special clock at each arena that the judges will start when they call for the next contestants to get ready. The Robot must begin its trial before that clock reaches 1 minute. Any Robot that is not ready to run after 1 minute will forfeit its chance at that
trial. It may still compete in any other trials. Once assigned, the order of running will not
be changed. If you are not ready, then you've missed your turn. The time between turns
is undetermined and is controlled by how long the other comp etitors take to complete
their trials.
Once the Robot is ready, the location of the candle and any furniture shall be determined by the arena master.
The contestants will show a judge how to actuate the Robot. The judge will press whatever buttons necessary to start the Robot.

16.TIME LIMITS

In order to achieve the contest objective of building a Robot that can find and extinguish
a fire in a house, finding the fire within a reasonable period of time is very important.
The maximum time limit for a Robot to find the candle will be 5 minutes. After 5 minutes
the trial will be stopped. The maximum time for the Robot to return to the Home circle in
the Return Trip mode will be 2 minutes. If in any trail, a Robot gets stuck in a loop and
performs the same movement 5 times in a row, that trial will be stopped. Any time the
Robot does not move at all for 30 seconds, the trial will be stopped. Stopping a trial run
for any of the above reasons will have no impact on any of the other two trial runs that
the Robot has.

17.SCORING

The Robot with the lowest Final Score (FS) is the winner. The Final Score is calculated
from a number of different factors, which are explained below. The scoring process is

really not as complicated as it might seem at first. It is intended to make the contest as
realistic and as fair as possible. We are sorry if it reminds you of the federal tax code.

18.OPERATING MODES

For any trial, the lower the Operating Score (OS), the better. The simplest method of running a Robot is in the Standard Operatio n mode. There are 5 different Operating Modes (OM) which contestants can choose to apply to their Robot either individually or
in combination to change the Operating Score for that trial. These Operating Modes are
the Tethered, Sound Activation, Return Trip, Furniture and Uneven Floor.

STANDARD OPERATION - In this mode there are no wires connecting the Robot to anything. The deciding factor in determining this mode is whether or not there is a tether connected to the Robot. The Mode Factor for running in the Standard Mode is 1.0
(MF= 1.0)

TETHERED - In this mode, the Robot has a wire connecting it to either an external computer or power supply. This Mode Factor is actually a penalty and increases the Operating Score. If the Robot has its own on-board power supply and is controlled by either an on-board computer or via a wireless link to another computer then it will not
be in this mode and will not have an increased score. The Mode Factor for running in the
Tethered mode is 1.2 (MF = 1.2).

SOUND ACTIVATION - Instead of being manually activated by pressing buttons on the
Robot or on the keyboard, the Robot activates itself when it detects a sound signal between 3.0 kHz and 4.0 kHz. This is the frequency commonly used in smoke detectors
and is created by piezo-electric devices available at Radio Shack and many other sources. Once turned on, the Robot cannot start to move until the sound signal is activated. If the Robot starts to move before the sound signal is activated, for example
because it mistakenly detected ambient room noise (even the sound of another Robot
being activated in a different arena), then the trial can still count, but the Robot will not
get credit for operating in the Sound Mode. If the Robot does not start to move in response to the sound signal it will not be given a second chance (i.e. another press of
the sound button) to run in the sound mode for that trial. The sound signal device can
be held at any distance from the Robot that the contestants want and can continue for
up to 5 seconds. The time for the trial will begin when the sound signal is created and
not when the Robot actually starts to move in response to that signal. There will be an
official sound signal device at the contest, but contestants can bring and use their own
sound devices operating within the proper frequency range if they want. There will be a

5% reduction in score for a Robot operating in this mode. The Operating Mode factor for running in the Sound Activation mode is 0.95 (OM = 0.95).

RETURN TRIP - After extinguishing the candle, the Robot returns to the Home Circle. It does not have to retrace its path in returning to the Home Circle or even take the most efficient route, it just must get back once it has put out the candle. It must leave that room and return to the Home Circle without entering any other rooms.

The Robot will be considered to have returned to the Home Circle if the Robot stops with any part of the Robot within the 30 cm white Home Circle. The Robot does not have to be in the same position that it was when it started the contest.

If a Robot is entered to run in the Return Trip mode and finds and extinguishes the candle, but doesn't return to the Home Circle, the Robot would not be disqualified. Instead the Robot would drop back into the Standard Operation Mode and it would just receive the Operating Score with no Return Trip mode factor reduction.

The Actual Time (AT) score will include just the time the Robot takes to find and extinguish the candle. It will not include the time for the Robot's return trip to the Home Circle. Operating in this mode will result in a 20% reduction in the score. The Operating Mode factor for running in the Return Trip mode is 0.8 (OM = 0.8).

FURNITURE - In this mode there will be one or more pieces of furniture in each room. The Robot may touch the furniture, but it cannot push it out of the way. The furniture will be made of cylinders approximately 11 cm in diameter, painted semi-gloss yellow. The cylinders are 30 cm high and weigh more than 2 kg.

The furniture will NOT block a doorway and a Robot will be able to come into a room at least halfway before it encounters furniture. The furniture will always be placed so that there is at least one path to the candle that is at least 31 cm wide. The possibility that the furniture may be blocking the Robot's view of the candle or that the Robot may have to go around the furniture to get to the candle is what makes the Furniture Mode such an interesting and real-world realistic challenge. The Robot may have to look around the room from different locations to see if the candle is there. If the candle is indeed behind furniture, the Robot may have to determine what is the best way to go around the furniture to get to the candle. Successfully operating in this mode will result in a 50% reduction in the score. The Operating Mode factor for running in the Furniture mode is 0.5 (OM = 0.5).

UNEVEN FLOOR - Many Robots use a form of dead-reckoning to travel through the arena. That is, once correctly oriented at the start of the arena, they count the distance
moved and angle turned and add them to their old position to obtain their new location
and orientation. While this is a perfectly good and legitimate method of traveling through the arena in this contest, it is not as practical or useful in the real world where
the floors are often uneven and surfaces irregular. So to encourage Robots to use more
sophisticated methods of determining their position and orientation within the arena, we
are giving a score reduction bonus to Robots that do not use a dead-reckoning method.

The key to using dead-reckoning is knowing the distance before-hand to the various rooms in the arena. Under normal conditions to aid in dead-reckoning, the floor surface
of the arena is made as smooth and uniform as possible. However, if you decide to operate in the Uneven Floor mode, the uniformity of the floor will be taken away by adding Floor Items that will change how the Robot wheels travel over them. For example, there might be a Floor Item placed on the floor that is slightly elevated, thus
giving one wheel a greater distance to travel. If you decide to run your Robot in this Uneven Floor mode, we will place one or more Floor Items in the hallways of the arena,
which will have the effect of changing the path to the rooms. Since the robot will not know exactly where this changed path will be placed or which wheel it might effect and
by how much, the Robot will have to use other methods besides dead reckoning to determine its location and orientation within the arena.

There may be more than one Floor Item used during a trial. The Floor Items will only be
used in hallways and not in rooms. The Floor Items will not be placed in the hallway directly outside of a doorway, although one could be placed next to a doorway. The number and location of the Floor Items will be changed from trial to trail. The Floor Items will remain in place during the Return Trip portion of the trial. The Floor Items are
NOT meant to be a barrier, but to solely disrupt dead-reckoning by changing the condition of the floor surface. The maximum height of the Floor Items will be 5 cm. If necessary, the Floor Items will be tapered and there will be as smooth an intersection
with the flat floor as practical. There will NOT be any steps or sharp drops greater than 5
mm. The average maximum slope of the incline will be 15 degrees. The Floor Items will
be painted the same flat black as the floor. The exact placement of the Floor Items will
be unknown to the Robot before the start of the contest. Successfully operating in this
mode will result in a 30% reduction in the score. The Operating Mode factor for running
in the Uneven Floor mode is 0.7 (OM = 0.7).

19. STARTING THE ROBOT MANUALLY

If the Robot is not being run in the Sound Mode then it must be started manually, that is, by having a contest official press the indicated buttons.

EXTERNAL COMPUTER - If the Robot is using a tether connecting it to an external computer then the only key that can be pressed to start the Robot is the "Enter" or "Return" key on the computer keyboard. A contest official will press the key. Any program that needs to be run must be loaded and ready to go before the Robot is put in the arena. Once the Robot is in place and the candle put into position, only the "Enter" or "Return" key can be pressed to start the Robot. If for any reason the Robot does not start, then that trial is over.

ON-BOARD COMPUTER - If the Robot is using an internal computer, then there can be one and only one button that can be pressed to start the Robot. This button must be positioned some place easy to see and get to on the Robot and must be labeled as such, i.e., "START". "RUN", "GO", etc. Any program necessary must be downloaded to the Robot before it is put into the arena. Once that is done then the specific "start button" and only that "start button" can be pressed to actually start the Robot. If for any reason the Robot does not start that trial is over.

20.PENALTIES

The goal of this contest is to be as realistic as possible. Touching a wall or touching the candle are not illegal but they are not good operating procedures for the real world. Penalty Points (PP) will be added to the Actual Time (AT) of any Robot that touches a wall or touches the candle. Don't let these penalties scare you too much. These penalties are generally a small price to pay for a Robot that manages to accomplish the task.

Touching a wall - Any Robot that touches a wall with any part of its body or feeler, either deliberately or accidentally will have 5 points added to its Actual Time score for each occurrence. Any Robot that slides along a wall will have an additional 1 point added to its time score for each 2 cm of wall it touched as it was sliding along. A Robot can still touch a wall to orient itself, but it will be penalized for doing so. (PP = 5 per hit and PP = 1 per 2 cm of sliding)

There are no penalties counted for hitting the wall on the Return Trip back to the Home Circle after extinguishing the candle.

Touching the candle - Any Robot that touches the candle or its base with any part of its body or feeler, either deliberately or accidentally while the candle is lit, will have 50 points added to its Actual Time score. If the touch occurs as part of the actual extinguishing process (i.e. smothering the flame with a wet sponge) or after the candle is extinguished, there is no penalty. This touching refers only to a part of the Robot's

body and does not include any water, air or other material that the Robot might use to
extinguish the candle. (PP = 50)

21.ROOM FACTOR

In order to make the contest realistic and to encourage the creation of smart Robots, we
have deliberately added uncertainty into the contest. The Robot does not know in which
of the 4 rooms the candle has been placed. Sometimes a Robot gets lucky and the candle is in the first room it searches and sometimes a Robot is unlucky and the candle
is in the 4th room searched. The unfairness of this is that finding the candle in the 4th
room you look in is a lot harder and takes longer than finding it in the 1st room you search. To reduce the impact of "luck" and give some credit to the more sophisticated
Robots that can search multiple rooms successfully, there will be a Room Factor involved
in the scoring that will be multiplied by the Time Score to get the Operating Score. The
more rooms a Robot has to search before it finds the candle, the lower the Room Factor
and thus the better the Operating Score.

If the candle is in the 1st room searched, the Room Factor will be 1.0
If the candle is in the 2nd room searched, the Room Factor will be 0.85
If the candle is in the 3rd room searched, the Room Factor will be 0.50
If the candle is in the 4th room searched, the Room Factor will be 0.35

It does not matter in which order the Robot searches the rooms. The only thing that matters is how many rooms the Robot has searched before it finds the candle.
After searching a room with a lit candle in it, there is no further reduction of room factor. This is true whether or not the robot extinguishes the candle. No matter how many rooms the robot continue to search, it will have no effect on room factor.
Some Robots have extremely sensitive sensors and can tell if the candle is in the room
by merely looking in the doorway as it passes by. The Robot does not have to enter a
room to be considered to have searched it. Any Robot going past a doorway that it has
not gone past before will be considered to have searched that room. If the Robot has already searched a room and then goes past the doorway again on its way to a different
room, that room will not be counted twice.

22.SCORING PROCEDURE

A. Multiply the Operating Modes together to get the Mode Factor (MF)
(Tethered=1.2, Sound = 0.95, Return=0.8, Furniture=0.5, Uneven Floor=0.7
(If none of the Operating Modes are used and the Robot is running in the Standard Operation then MF=1.0
B. Record the Actual Time (AT) in seconds needed to put out the candle
C. Add all the Penalty Points (PP) together
hitting a wall = 5 points per hit
sliding along wall = 1 point per 2 cm
touching the candle or base while the candle is lit = 50 points
D. Record the Room Factor (RF)

1st room = 1.0, 2nd room = 0.85, 3rd room = 0.50, 4th room = 0.35
E. Add the Actual Time to the Penalty Points to get the Time Score (TS)
TS = AT + PP
F. Multiply the Time Score, Room Factor and Mode Factor together to get the Operating
Score (OS) for that trial.
OS = TS x RF x MF
G. The method for determining the winner in the Expert Division is given below in Section 25. The top three robots in each of the other divisions will be determined as follows:

- Robots with three successful runs (runs when the candle is extinguished) will form the highest group. The top three robots in the group will be ranked according to the sums of their three OS scores as determined in A-F above.
- If there are fewer than three robots in any division with three successful runs, the remainder of the top three prizes will be determined by ranking, according to finishing times, the robots that complete two successful runs.
- Prizes will not be awarded to robots that have fewer than two successful runs.

23.SCORING EXAMPLES

1st Trial: If a Robot runs its first trial in the Standard, Sound and Return modes, takes 1 minute and 23 seconds to extinguish the candle in the 2nd room while hitting the wall 3 times, its Operating Score for that trial would be:
A. Multiply the Operating Modes together to get the Mode Factor (MF) (Standard=1.0, Sound = 0.95 and Return=0.8)
MF = Std x Snd x Rtn = 1.0 x 0.95 x 0.8 = 0.76
B. Record the Actual Time (AT) in seconds needed to put out the candle
AT = 83
C. Add all the Penalty Points (PP) together (hitting a wall = 5 points/hit)
PP = 15
D. Record the Room Factor (RF) (2nd room = 0.85)
RF = 0.85
E. Add the Actual Time to the Penalty Points to get the Time Score (TS)
TS = AT + PP = 83 + 15 = 98
F. Multiply the Time Score, Room Factor and Mode Factor together to get the Operating
Score (OS)
OS = TS x RF x MF = 98 x 0.85 x 0.76 = 63.31
2nd Trial: If the Robot runs its second trail in the Standard, Sound, Return and Uneven Floor modes, takes 1 minute and 41 seconds to extinguish the candle in the fourth room searched while accidentally bumping into the candle, it Operating Score for that trial would be:
A. Multiply the Operating Modes together to get the Mode Factor (MF)
(Standard=1.0, Sound = 0.95, Return=0.8 and Uneven Floor=0.7)
MF = Std x Snd x Rtn x UnF = 1.0 x 0.95 x 0.8 x 0.7 = 0.532
B. Record the Actual Time (AT) in seconds needed to put out the candle
AT = 101
C. Add all the Penalty Points (PP) together (hitting candle = 50 points)
PP = 50
D. Record the Room Factor (RF) (4th room = 0.35)
RF = 0.35

E. E. Add the Actual Time to the Penalty Points to get the Time Score (TS)
TS = AT + PP = 101 + 50 = 151
F. Multiply the Time Score, Room Factor and Mode Factor together to get the Operating
Score (OS)
OS = TS x RF x MF = 151 x 0.35 x 0.532 = 28.12
3rd Trial: In the third trial the Robot tried to run in the Sound, Return and Furniture modes. It extinguished the candle in the first room in 1 minute and 10 seconds, but it
did not make it back to the Home Circle.
A. Multiply the Operating Modes together to get the Mode Factor (MF) The Robot did not
make it back to the Home Circle so it loses the Return mode reduction.
(Standard=1.0, Sound = 0.95 and Furniture = 0.5)
MF = Std x Snd x Frn = 1.0 x 0.95 x 0.5 = 0.475
B. Record the Actual Time (AT) in seconds needed to put out the candle
AT = 70
C. Add all the Penalty Points (PP) together
PP = 0
D. Record the Room Factor (RF) (1st room = 1.0)
RF = 1.0
E. Add the Actual Time to the Penalty Points to get the Time Score (TS)
TS = AT + PP = 70 + 0 = 70
F. Multiply the Time Score, Room Factor and Mode Factor together to get the Operating
Score (OS)
OS = TS x RF x MF = 70 x 1.0 x 0.475 = 33.25
Final Calculations: Now the Robot is done with its 3 trials.
G. The three OS scores are added together to get the total OS score (TOS) for the robot: TOS = 63.61 + 28.12 + 33.25 = 124.98.
Note: The Robot can choose different modes during each of its three trials. The candle and any furniture, if necessary, will be moved to different locations for each trial.

24.EXPERT DIVISION

The Expert Division was established in 2001 to challenge the most experienced firefighting Robot designers and to clearly identify the best Robots in the contest. Each
year the Expert Division has presented new and more challenging tasks. In 2001 and 2002 Expert Division Robots were required to operate within the standard maze using all
of the deductions (audio start, arbitrary start, furniture, etc.) In 2003 the contest added
a new challenge, a larger maze whose geometry changed from run to run.
For 2004 we have increased the challenge by including both firefighting and search-andrescue
tasks. In 2004 Expert Division Robots will carry out the tasks of a fire department scout Robot that searches for a baby in a two-story house, marks the baby's location, and puts out fires.
The first floor of the house consists of the same 3m x 3m arena used in the 2003 contest. The new second floor measures 2m x 2m. In 2004 Robots will reach the second
floor via a ramp (see ramp specifications below).
There will be two bedrooms on the second floor. A simulated baby located in one of

these upstairs bedrooms must be found and marked (so that a fire department rescue
robot can save the baby). In addition, the scout Robot must extinguish two candles.
Candles can be located in any room on either floor.

THE EXPERT DIVISION TASKS

The FF Robot scout's goal is to complete the four tasks below. The tasks can be completed in any order.

1) Put out candle 1;
2) Put out candle 2;
3) Find and mark the baby by placing an audible beeper (see specifications below) within 20 cm of the baby;
4) Go up and down the ramp at least once. This must be accomplished in a controlled
fashion.

Note: Robots that elect not to go to the second floor will run using essentially the 2003
Expert Division rules. However, the 2004 scoring method will be different. See section
on scoring below.

RULES FOR THE 2004 EXPERT DIVISION

The penalties for the Expert division will be the same as for the other divisions. Differences between the Expert division and the other divisions are listed below.

A. The First Floor

1) The first floor will measure 3 meters by 3 meters square. (Click here for possible Expert Division arena diagrams)
2) The outer walls will be stationary, but the inner walls that define the hallways and rooms will be moveable and will, in fact, be moved between each trail.
3) Wall height will be approximately equal to that of the standard maze.
4) There will be 2 to 5 rooms in any trial and their position, size and doorway location will change from one trial to another. Note: A room will have at least a 2 by 2 grid area, where 1 grid length is approximately 50 cm. A room does not have to be rectangular and it may have alcoves and bends. The door to a room will not be smaller then 1 grid, but it could be wider. Everything else is a hallway.
5) A room will only have one doorway and that doorway will be connected to the hallway and not to another room.
6) Hallways may lead to dead ends.
7) The hallways and doorways will be approximately 48 cm wide.

B. Second floor

1) The second floor will measure approximately 2 meters by 2 meters square. (Click here for possible Expert Division arena diagrams.)
2) Wall height will be approximately equal to that of the standard maze.
3) The outer walls will be stationary, but the inner walls that define the hallways and rooms may be moveable and may be moved from run to run
4) There will be 2 bedrooms on the second floor with connecting hallway (s).
5) A room will only have one doorway and that doorway will be connected to the hallway and not to another room.
6) Hallways may lead to dead ends.
7) The hallways and doorways will be approximately 48 cm wide.

C. Staircase (ramp)

1) A straight ramp will connect the first and second floors.
2) The ramp will start and end on the edges of the floors.
3) The ramp will meet the lower and upper floors at 90 degree angles.

4) The ramp will not necessarily be centered on grid lines.
5) The entrance to the ramp is not marked in any way.
6) The ramp angle will not exceed 15 degrees.
7) The width of the ramp will be approximately 48 cm.
8) The ramp will have walls that are similar, in height and style, to the walls of the maze.
9) The length of the ramp is not specified exactly, but it will be between 150 and 300 cm.

These arena diagrams show only a few of the many possibilities. The Expert Division is trying to encourage development of fantastic state-of-the-art Robots that can operate in a truly real-world environment, where nothing is precisely known.

D. Baby

1) The baby is a toy doll made of soft fabric material. The baby is approximately 28 cm. in length.
2) Through a belt on the baby, the baby emits infrared signals that simulate body heat. The nominal IR wavelength will be 880 nm.
3) The IR emitted by the belt is modulated so it can be detected by standard remote control receiver modules. The carrier frequency is 36.7 kHz ± 5%. This carrier is modulated by a 300 Hz ± 5% rectangular wave with a duty cycle of 20%.
4) The belt contains multiple IR emitter diodes and a diffuser to achieve a wide radiation pattern. However, we do not specify or guarantee any radiation pattern.
5) Robots may employ a non-destructive probe to verify the baby’s position. Robots will be disqualified if the baby is injured.
6) The baby will be in a wooden bed. The height of the bed will be in proportion to the room size and the size of the baby.

E. Beeper

1) The beeper dropped by the robot must emit a 1 kHz tone pulsed twice per second.
2) The beeper must operate for at least one minute and be loud enough to be heard by the judges at a distance of at least 3 meters.
3) Participants must provide their own beepers.

F. The robots in this division must be untethered and using either on-board computers
or an external desktop computer with an RF link. There cannot be any wires from the external computer to the Robot.

G. Expert Division robots must operate in the Sound, Uneven Floor and Furniture modes.

H. The Sound mode is mandatory and a failure to start properly in Sound mode will nullify the run (zero tasks, 6 minutes adjusted time—see Scoring below).

I. There will NOT be a white line in the doorway to a room.

J. Even though some part of the Robot must still come within 30 cm of a candle before
it attempts to extinguish the candle, there will NOT be a candle circle to indicate that the Robot is within the correct distance. Thus the Robot will somehow have to make sure that it is close enough to the candle before it starts the extinguishing process.

K. The floor in the rooms may not be uniformly black or even smoothly flat. Some rooms may contain more real-world type floors made of such materials as linoleum, tile or even thin rugs (less than 5 mm).

L. The walls in the Expert division may not be uniformly white or even smoothly flat. There could be pictures or other materials hung on the walls which change the color, texture or reflectivity. In any case, nothing will extend more than 5 mm from the

wall surface.
M. Eligibility for cash prize: In order to win a cash prize in the Expert mode, a Robot must complete at least three tasks during three runs.
N. Qualification: To qualify for the final competition, a robot must complete at least two
tasks in one run within six minutes.
O. Home Circle. There will be a 30 cm (diameter) white home circle for the robot to start in. The home circle may or may not be centered on the grid. If the robot completes at least one task and returns to the home circle within the ma ximum six minutes, it will receive a 10% time reduction for that run. The home circle can be anywhere in the first floor of the arena, even in a room. The robot must not enter any rooms on its way back to the home circle (must stay in hallways), but it does not have to take the most direct route back. This Return Mode will be optional and a failure to return will NOT cancel the trial.
P. All maze intersections will be at right angles. There will NOT be any diagonal hallways or walls.
Q. All rooms will be at least 2x2 grids in size (a grid is approximately 48 cm on a side).
Rooms do not have to be square or even rectangular.
The candle will NOT be in a hallway.
There may be more than one Furniture item in a room.
The Room Factor discount (Section # 21) will NOT be applied to the Expert division.
The orientation of the Robots on the Home circle will be determined by the judges at the
beginning of a trial.
EXPERT DIVISION SCORING
Note: There is no special bonus for reliability in the Expert Division since all three runs
are used in determining scores.
The scoring method (see example on our web pages) counts completed tasks and it uses
time as a differentiator among robots with the same number of completed tasks.
A. Each robot is allowed three runs.
B. The maximum run time for each run is 6 minutes. This time will be recorded for runs
that are not fully completed.
C. For each run, judges count the number of tasks completed and measure the raw time and penalties.
D. An adjusted time is computed by adding penalty time to raw time.
E. Robot score is then taken as follows:
Add up total number of completed tasks for the three runs;
Add up the total adjusted run times for the three runs;
Finishing rank is computed by the total number of tasks completed using time as the tiebreaker.
EXPERT DIVISION SCORING EXAMPLE
Robot A has the following runs:
Run 1: Finds a candle on the first floor and puts it out. Goes up the ramp. Finds and marks the baby. Fails to go down ramp (does not return to first floor in controlled fashion). During the run, the robot hits the wall twice.
Two tasks completed.
Raw time = 6 minutes (did not complete four tasks)
Adjusted time = 6:10 (penalty times added for the two wall hits)
Run 2: Hits wall and stops. Zero tasks, 6:05.

Run 3: Completes all four tasks, 2:56.
Score: 6 tasks, total time = 15:11.
Robot B has following runs:
Run 1: Extinguishes one candle, 6 minutes.
Run 2: Extinguishes two candles (both on first floor), 6 minutes.
Run 3: Completes all four tasks in 5:24 and returns home, receiving a 30 second deduction. Adjusted time = 4:56.
Score: 7 tasks, total time = 16:56.
Robot B's final ranking is higher. It completed more tasks even though its total run time
is higher.

25.DIVISION DECISIONS

Each division will have its own set of winners and prizes (see section #30 - Prizes).
Anyone who meets the criteria for a particular division may, at their option, decide to run in a higher division. Contestants will not be able to run in a lower division than that
which they should be in. This means that an 8th grader could decide to run in the High
School (or even Expert) division if they want to try to win more money, fame and glory.
When registering for the contest, contestants will be asked to select a division to run in
and no division changes will be allowed after registration.
No single Robot can be entered in two different divisions. If a contestant wants to operate in two divisions, then they must enter two different Robots. (See the rule on Multiple Entries - section #31)

26.CHALLENGES OF JUDGES' RULINGS

The Contest Master is the FINAL AND ABSOLUTE authority on the interpretation of all rules and decisions. Any contestant who wishes to challenge any ruling or scoring of the
arena judges to the Contest Master must do so BEFORE they leave the arena area. If a
contestant has a problem or question about any decision the Arena Judges have made,
they simply have to say that they wish to appeal this to the Contest Master. The Contest
Master will then be called in to arbitrate the matter. Once the contestants have left the
arena they cannot appeal any decision or scoring of the Arena Judges.

27.ADULT HELP

The division structure was created to make the event more fun for students, but at the
same time we realize that we are opening another entire area of possible conflict and problems. The problem occurs with a Robot submitted by a group consisting of people
both in and out of school.
An easy case might be one in which a microprocessor controlled, stepper motor driven
Robot using modulated IR sensing with the programming written in C++ is submitted by
a 2nd grader whose father just happens to work for NASA. This Robot would probably
end up in the Senior division.

In general, a Robot created by a group of 6th and 7th grade students with an adult teacher advisor, would probably be entered into the Junior division since it is our experience that the students really do build and program the Robots themselves. We don't care who helps a team of college students since they will be in the Senior division which is open to anyone. However, the Robot entries in the Junior and High School divisions are supposed to be actually created by the students themselves. This does not mean that the students have to do everything, i.e., mechanics, hardware, electronics, software completely on their own. But on the other hand, we would not like to see a teacher spending hours upon hours writing and debugging a student's software. Adults helping are OK; adults taking over are not.

As far as the students are concerned, the goal of the contest should be education and not necessarily winning. We know that the students desperately want to win, but the adults should let them compete (win or lose) on their own. This contest is pretty much on the honor system, but we expect that the student contestants are primarily responsible for the creation of their Robots. If we find any case to the contrary, they will be assigned to a more appropriate division. We will try to be very fair, and as in everything else, the decision of the Contest Judging Committee is final.

28.PRIZES

There will be cash prizes for the top Robots in each division that compete at least one successful run. The exact value of the cash prizes will be listed on the contest website.

There will also be additional prizes donated by our contest sponsors and other interested supporters. All Robot entries, which participate in the contest, will receive a Certificate of Achievement and an official contest T-shirt.

29.MULTIPLE ENTRIES

The guiding principle of the Trinity College Fire-Fighting Home Robot Contest and its regional contests is that every Robot entered is to be an original and unique design. Thus an individual, team or school cannot enter multiple identical Robots. A team may enter more than one Robot, but they must be signific antly different from each other in at least some aspects of electronics, software and mechanics. The challenge of this contest is for every contestant or team to complete a unique Robot of their own design.

30. QUALIFICATION TRIALS

In order to run in the final competition on Sunday each robot must demonstrate that it can function in the arena as intended (see the exception in Section D below). The Saturday qualification period begins at 10 a.m. and ends promptly at 9 p.m. Robots may qualify at any time during that period. During the qualification period each robot will have a maximum of three chances to find and extinguish the candle, subject to the following rules:

A. The 3 qualification trials do not have to be run consecutively. A robot can come back
after adjustments to try again.
B. Once a robot has successfully qualified by finding and extinguishing the candle, it does not have to complete any further trials. A robot only has to find the candle once to be qualified for the contest on Sunday.
C. If the robot cannot find and extinguish the candle once during its 3 qualification trials, then it has not qualified for the contest on Sunday.
D. First, Second and Third place winners of Official Regional Contests do not have to qualify, but the head of those Regional Contests must notify the Event Coordinator (juliet.manalan@trincoll.edu) of their names by April 1.
E. The candle will be placed in a room chosen by the contestant. The qualification judge will place the candle in a randomly-chosen position in that room.
F. There is a five-minute limit on each qualification run. Any run that exceed five minutes WILL BE RECORDED AS UNSUCCESSFUL AND will be counted as one of the three allowed runs.
G. The rules concerning not moving for 30 seconds or repeating the same movement 5
times will apply.
H. When you are ready to make a qualification trial, you will notify the qualification judges and they will give you a trial position. (For example: "There are 3 robots ahead of you in line and when they are done then you go.")
I. When it is your time to make your qualification trial you will have 1 minute to get set
up and begin your run. If you can't begin within the 1 minute setup time, this particular qualification trial is over and it is counted as one of the three runs.
J. The qualification period will end at 9 pm on Saturday sharp. Any robots that have not qualified by that time FOR ANY REASON will not be qualified for the contest on Sunday. It is your responsibility to qualify before the Qualification period ends.
K. The qualification trials will only take place on Saturday. There will be a short practice
session on Sunday, but there will NOT be any qualification trials on Sunday.
L. Robots do not have to qualify in the same operating modes that they will run in on Sunday except that robots competing in the Expert Division will have to qualify in the special Expert Division arena and will be subject to the Expert Division rules.
M. Final decisions regarding qualification issues will be made by the Chief Judge (christopher.wynschenk@trincoll.edu).

31.PRACTICE TIME

The Robot should be built, programmed and ready to run on arrival at the contest site.
Practice time in the arenas will be limited due to the number of participants and because
some of the arenas will be used all day Saturday for Qualific ation Trials. Practice time is
intended to be used for calibrating sensors to the conditions in the gym and trouble shooting last minute problems. Don't expect to be able to do extensive code development and testing.

32.ASSIGNED ARENAS

There will be arenas set aside for the Qualification Trials on Saturday. The other arenas
will be available for competitors to practice in. However on Sunday morning before the

contest, the qualifying competitors will be told which arenas they will actually compete
in for which trial. They will have some limited time before the actual start of the contest
on Sunday in which to make any final adjustments to their Robots in this arena. It is very likely that Robots in the High School and Senior divisions will run each trial in a different arena. We will strive to make the lighting and other factors the same for each
arena, but there will be some variations. Your Robot should be able to handle them. The
Robots should be prepared to run in any arena for any trial. The Robots in the Expert, Walking and Junior divisions will only run in their single assigned arena and will not switch arenas.

33.SAFETY

The contest judges may stop any Robot at any time if they feel that it is performing, or
is about to perform, any action that is dangerous or hazardous to people or equipment.
No Robot is allowed to use any flammable or combustible processes.

34."SPIRIT OF AN INVENTOR" PRIZE

In 1999 a walking Robot was entered in the contest. It was an incredible device that could actually walk on 2 legs and find and extinguish the candle. Even though it had absolutely no chance of winning the contest because it was so slow, the inventor entered
it anyway because it was such a good idea. We were so impressed with this attitude that
there will be a special prize for the most unique Robot that does not win the contest, but
shows the greatest creativity, ingenuity and a true "Spirit of an Inventor." A Robot does
not have to conform to all the rules in order to be eligible for this prize.

35."COST-EFFECTIVE" PRIZE

Robotics does not have to be expensive. Spending money does not guarantee success,
in fact, some of the very best Robots have been some of the cheapest. To award financial efficiency, there will be a special prize for the best performing Robot built with
the smallest amount of money in material cost. If you put in $50,000 in labor and destroyed $5,000 in parts finally getting it to work, but your final Robot has less than $200 in actual parts in it, then it is a good contender for this prize. It does not matter
what you paid for the parts, but only what they are worth. A motor which originally cost
$50, but is now for sale in a surplus catalog for $5 is now a $5 motor. However, if you
got a $50 motor for free from a friend, then it's still a $50 motor regardless of the fact
that you got it for free. If, on the other hand you destroyed three $50 motors in building
the Robot, you only have to account for the one motor that is actually on the Robot.
Evaluation Method:
A. As part of the on-line registration process teams will indicate in a check box on the

registration form whether they wish to be considered for the Cost-Effective Prize (CEP).
B. Participating teams will prepare an inventory for their robot that lists all parts and their prices. Use guidelines above.
C. To qualify for the CEP, robots must qualify for the competition on Saturday.
D. Following the qualification run, two judges will inspect the robot and verify the inventory.
E. Each robot will be put into a cost category (CC): (1) CC1: under $100 U.S.; (2) CC2: $100-$150 U.S
F. Robots will be ranked as follows:

- Best two runs will be used to compute a total operating score (TOS).
- CC1 robots will be identified and winner determined according to TOS.
- If there are no successful CC1 robots, judges will determine winner from CC2 group using TOS for two runs as above.

36.PRESENTATION PRIZE

A section of the contest floor will be reserved for the display of posters, presentations and exhibits, dealing with topics of interest and there will be some great prizes for the winners. The poster, display, presentation and/or exhibit can be any shape or size and deal with any sort of Robotics related topic. This could include anything such as: school programs, software algorithms, historical information or trivia, basic descriptions of research, educational curriculum or strategies, mechanical construction techniques, descriptions of technology used or proposed, write-up and descriptions of Robots running in the contest, explanations and descriptions of any other Robots that might be in progress, or any topic or subject that might be of any interest or value to anyone at the contest This prize is open to anyone of any age or affiliation, whether they are competing with a Robot or not. There is no registration or fee to enter the poster session. Simply show up and set up your presentation. Judging will take place on Sunday after 12 noon on the basis of interest, presentation and informative value. Winner will be announced at the final awards ceremony on Sunday. All materials will be returned to their creators at the end of the contest.

37.INTERPRETING THE RULES

In all matters of interpreting these rules before and during the contest and in any issues not covered by these rules, the decisions of the Contest Judging Committee will be final.

38.WHO CAN ENTER

There are no restrictions as to who can enter a Robot. Although most Robot entries will be submitted by individuals, there is no limit on the number of people, who as a group, can submit a Robot entry. Only one prize will be given to each winning Robot entry.

39.ENTERING A ROBOT

A non-refundable registration fee is required for each Robot entered into the contest. Any individual or group can enter more than one Robot, but a registration fee must accompany each entry. The same physical Robot cannot be entered twice even if two entry fees are paid. If you want to enter two Robots, then you must build two Robots.

Please make the checks payable to Trinity College and include the check with the entry
form.

40.ONLINE REGISTRATION PROCESS

A. Go to the secure registration web site and fill in ALL of the information. If you don't
have all the required information then wait until you do have all the information before you fill out the form. A pre-registration sheet will be available for download on the website to help you prepare.

B. Fill in the required fields on the website.

C. Confirmation of your successful registration will be emailed within three days to the
contact person provided on the form.

41.REGISTRATION DEADLINE

The sole purpose of requiring advanced registration is to help us plan the event.
If you do not register by April 1, 2004 then your Robot will not be in the contest. There
are NO EXCEPTIONS.
You have spent hundreds of hours and dollars on your Robot. PLEASE REGISTER EARLY!

42.LOCATION, DATE & SCHEDULE

The contest will be held at Trinity College in Hartford, Connecticut on Saturday & Sunday, April 17 & 18, 2004. A final schedule for the contest weekend will be posted on
the website.

43.REGIONAL CONTEST EVENTS

In order to enable people from all over to participate in this scientific, educational and
fun event, we are working with local groups around the world to establish regional contests that will occur before the main Trinity contest. The rules and regulations in the
regional contests will be approximately the same as those used in the main Trinity contest. (However they may not be exactly the same so check with the organizer of the
regional contest to find out any differences.) It is NOT mandatory for a Robot entered in
the main Trinity contest to have first competed in an regional event, but if you want to
compete in both, you certainly can. Any Robot that has come in 1st, 2nd or 3rd in a regional event does NOT have to qualify for the main Trinity contest, but THEY STILL DO
HAVE TO REGISTER and pay the appropriate registration fee. When they arrive at the
main Trinity contest they should also be sure that the Qualification Master is aware that
they had previously won a regional event and thus do not have to complete preliminary
qualification. Check the contest website for a list, schedule and contact information for
all of the regional contests. If your organization is interested in sponsoring a regional contest in your area next year, contact the Contest Coordinator for more information.

44.UPDATED INFORMATION

As updated information is developed it will be posted on the website so check it often for
the latest information.

45.CONSTRUCTION SCHEDULE

Contestants are supposed to have built their Robots at home and then merely bring them to the contest to run. This is NOT a construction contest where the devices are built at the event. Trinity will try to help out by providing some time and space for last
minute changes, adjustments and improvements, but the Robots are supposed to be done (or at least nearly so), by the time they get here. Contestants should also try to
bring any and all materials and equipment that they might need.

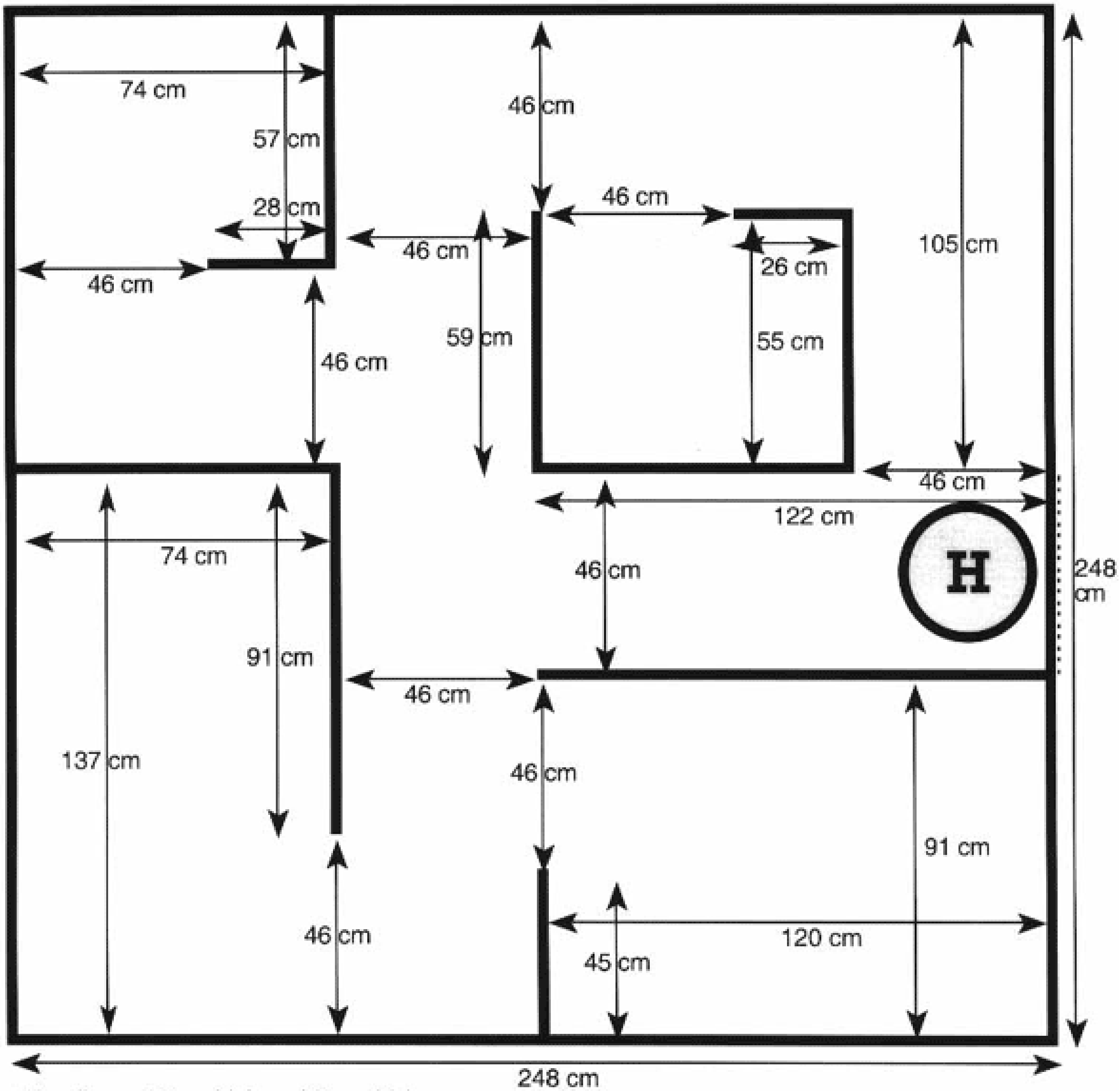
46.PERSONAL CONTACT

The best way to contact the Contest Coordinator, Dave Ahlgren, is by e-mail: dahlgren@mail.trincoll.edu

Trinity College FIRE FIGHTING Robot Contest

2004 Arena Floor Plan
Contest Rules, Attachment A

Senior & High School Divisions use all 4 rooms
Walker & Junior Divisions use only the bottom 2 rooms



All walls are 33 cm high and 2 cm thick

These dimensions are approximations and are NOT 100% accurate. Welcome to the real world!

Special thanks to Steve Bell for helping to rationalize the dimensions.

Appendix A References

[1] "Fire Fighting Robot Rules and Regulations", [PDF Online Document], Available HTTP: http://www.trincoll.edu/events/robot/Rules/Rules2004.pdf

Appendix B

AutoCAD layout of the chassis

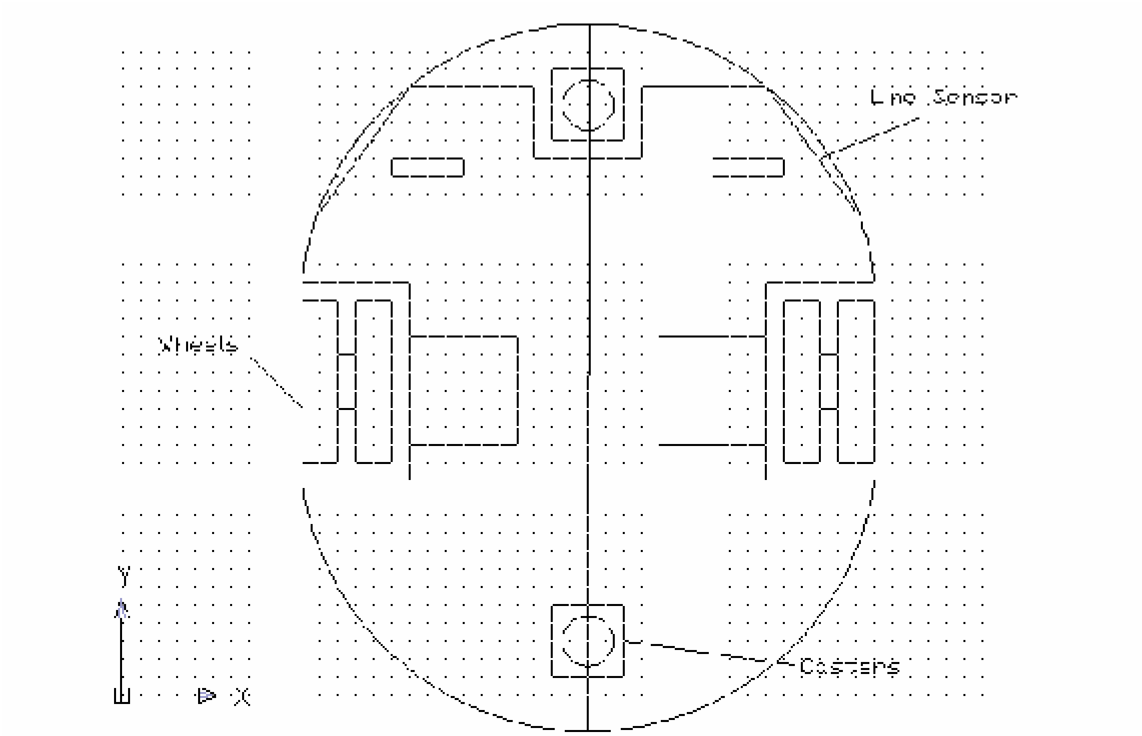


Figure: Bottom View of Robot

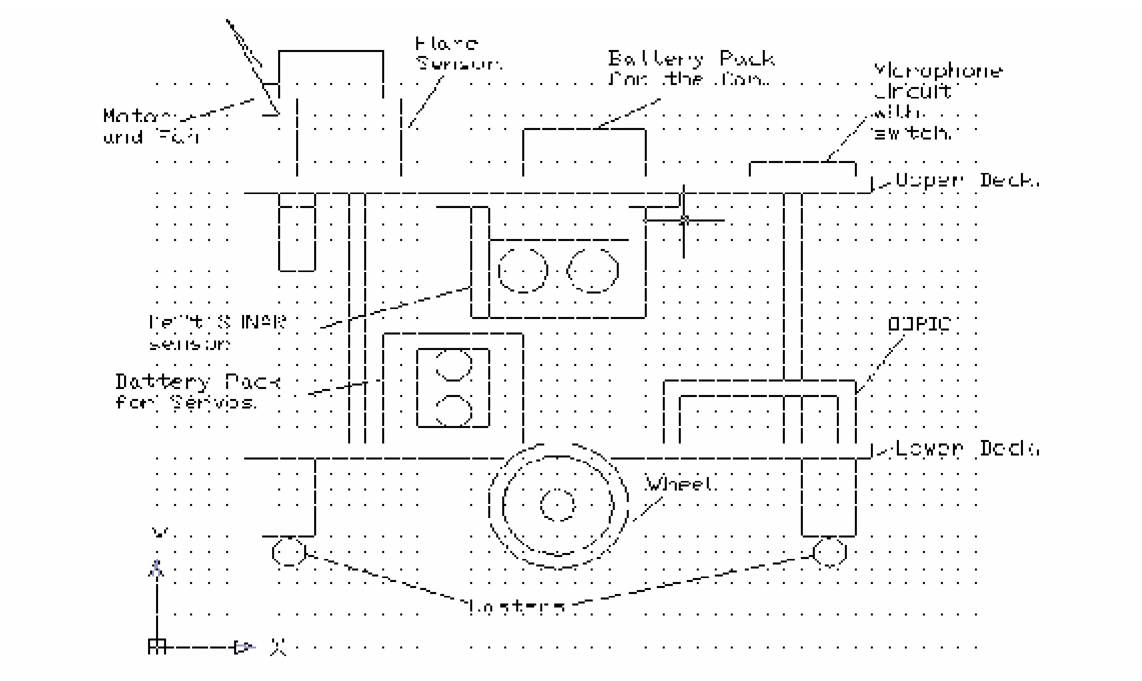


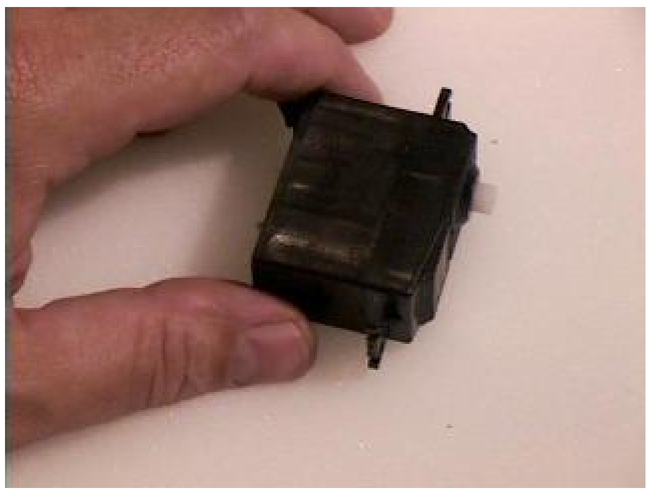
Figure: Left View of Robot

Appendix C

Servos

What is a Servo?

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio controlled cars, puppets, and of course, robots.



(Click on picture for larger view)

A Futaba S-148 Servo

Servos are extremely useful in robotics. The motors are small, as you can see by the picture above, have built in control circuitry, and are extremely powerful for thier size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is pretty strong for its size. It also draws power proportional to the mechanical load. A lightly loaded servo, therefore, doesn't consume much energy. The guts of a servo motor are shown in the picture below. You can see the control circuitry, the motor, a set of gears, and the case. You can also see the 3 wires that connect to the outside world. One is for power (+5volts), ground, and the white wire is the control wire.

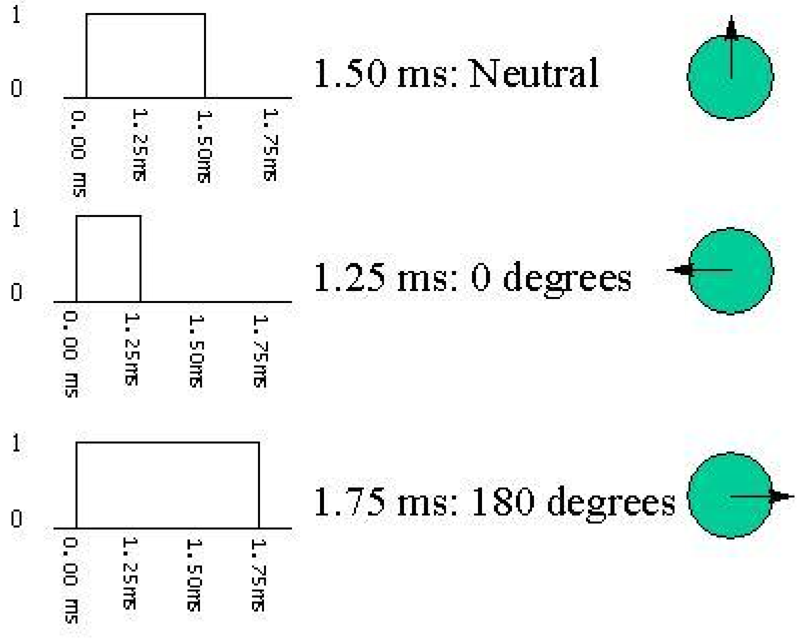
(Click on picture for larger view)

A servo disassembled.

So, how does a servo work? The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) that is connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. The output shaft of the servo is capable of travelling somewhere around 180 degrees. Usually, its somewhere in the 210 degree range, but it varies by manufacturer. A normal servo is used to control an angular motion of between 0 and 180 degrees. A normal servo is mechanically not capable of turning any farther due to a mechanical stop built on to the main output gear.

The amount of power applied to the motor is proportional to the distance it needs to travel. So, if the shaft needs to turn a large distance, the motor will run at full speed. If it needs to turn only a small amount, the motor will run at a slower speed. This is called proportional control.

How do you communicate the angle at which the servo should turn? The control wire is used to communicate the angle. The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse Coded Modulation. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90 degree position (often called the neutral position). If the pulse is shorter than 1.5 ms, then the motor will turn the shaft to closer to 0 degress. If the pulse is longer than 1.5ms, the shaft turns closer to 180 degress.



As you can see in the picture, the duration of the pulse dictates the angle of the output shaft (shown as the green circle with the arrow). Note that the times here are illustrative, and the actual timings depend on the motor manufacturer. The principle, however, is the same.

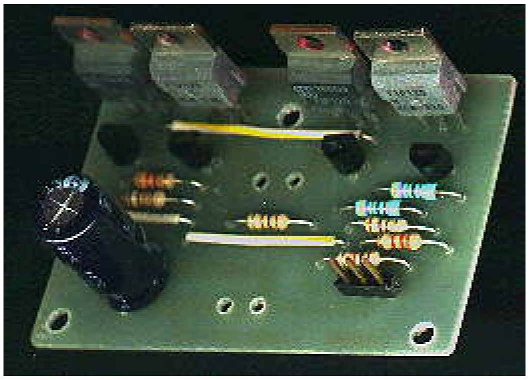
Reference:

[1] Seattle Robotic Society, "What is a Servo?", [Online Document], Available HTTP: http://www.seattlerobotics.org/guide/servos.html

Appendix D

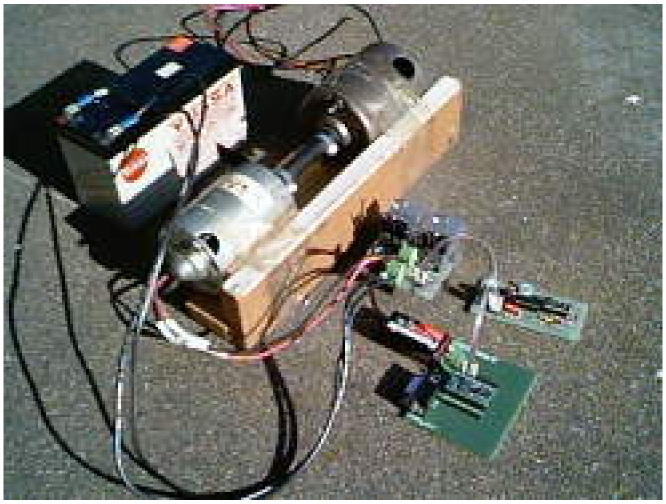
H Bridge

H-Bridge



This circuit drives small DC motors up to about 100 watts or 5 amps or 40 volts, whichever comes first. Using bigger parts could make it more powerful. Using a real H-bridge IC makes sense for this size of motor, but hobbyists love to do it themselves, and I thought it was about time to show a tested H-bridge motor driver that didn't use exotic parts.

Operation is simple. Motor power is required, 6 to 40 volts DC. There are two logic level compatible inputs, A and B, and two outputs, A and B. If input A is brought high, output A goes high and output B goes low. The motor goes in one direction. If input B is driven, the opposite happens and the motor runs in the opposite direction. If both inputs are low, the motor is not driven and can freely "coast", and the circuit consumes no power. If both inputs are brought high, the motor is shorted and braking occurs. This is a special feature not common to most discrete H-bridge designs, drive both inputs in most H-bridges and they self-destruct. About 0.05 amp is consumed in this state.



Stress-testing the H-bridge with Basic Stamp 2, PWM circuit, motor-dynamo, and 12 volt battery.

To do PWM(pulse width modulation) speed control, you need to provide PWM pulses. PWM is applied to one input or the other based on direction desired, and the other input is held either high("locked rotor") or low("float"). Depending on the frequency of PWM and the desired reaction of the motor, one or the other may work better for you. Holding the non-PWM'ed input low generally works best for low frequency PWM, and holding the non-PWM'ed input high generally works best at high frequencies, but is not efficient and produces a lot of heat, especially with these Darlingtons, so locked rotor is not recommended for this circuit.

Truth table:

```
input | output
A | B | A | B
---------------
0   0 | float
1   0 | 1   0
0   1 | 0   1
1   1 | 1   1
```

Performance:
Please reference the accompanying schematic diagram. The circuit uses Darlington power transistors to reduce cost. Forward losses are typically 1 to 2 volts, and since the current must pass through two transistors, expect losses to total up to 4 volts at maximum current. The 4 Darlington transistors need to be heatsunk based on your expected current and duty cycle.

PWM operation over 3 khz will likely lead to high losses and more heat dissipation, due to the simplicity of the circuit and the construction of Darlington transistors. You might get away with higher frequencies if you put a 1K resistor emitter-base on each TIP12x transistor. I prefer to go with very low frequencies, 50 to 300Hz.

Not shown in the schematic are the internal pinch-off resistors(5K and 150 ohms) and the damper diode that are built into all TIP12x series transistors. If you build your own variation of the circuit with other parts, include these neccessary parts. To the right is a picture of the internals of the TIP12x transistors.

Operation with logic signals greater than the motor supply voltage is allowed and absorbed by R7 and R8. The circuit is really intended to be operated with CMOS logic levels, logic high being about 4 volts.

If you live in the U.S., expect the TIP120 and TIP125 transistors to cost about $0.50 and the very common and generic "quad-2" PN2222A to cost about $0.10. An inexpensive source for hobbyist-grade parts like these is Jameco Electronics. At low duty cycles, currents up to the 8 amp rated peak of the transistors is allowed, but there is no current limiting in this circuit, so it would be unwise to use this circuit to drive a motor that consumes more than 5 amps when stalled.

Notes on circuit assembly:
Transistors Q1,2,3 and 4 must be heatsunk. Insulators should be used, or two separate heatsinks isolated from each other and the rest of the world. Note that Q1 and Q3 are grouped together and share common collectors and can share a heatsink. The same is true for Q2 and Q4.

Operation over 3khz will lead to higher losses. If it is required to run at higher frequency, additional pinch-off resistors can be added to Q1,2,3 and 4, supplementing the internal resistors. A good value would be 1k, and the resistors should be soldered from base to emitter.

To reduce RF emissions, keep the wires between the circuit and the motor short. No freewheel diodes are required, they are internal to the TIP series Darlington transistors.

Drive the circuit from 5-volt logic. Drive levels higher than 5 volts will tend to heat up R1 and 2. This is OK for short periods of time.

Power supply voltage is 5 to 40 volts. Output current up to 5 amps is allowed if the power supply voltage is 18 volts or less. Peak current must be kept below 8 amps at all times.

There is no current limiting in this circuit. Reversing a motor at full speed can cause it to draw huge currents, understand your load to avoid damage. There are higher powered devices in the TIP series of Darlington transistors, these can be substituted if needed. Look at the TIP140 and TIP145, please note they are in a bigger package and don't fit the PC board layout.

Reference:

[1] Bobblick, "H-Bridge", [Online Document], Available HTTP: http://www.bobblick.com/techref/projects/hbridge/hbridge.html

Appendix E

Datasheet for Servo S3010

Figure: High torque servo

Detailed Specifications
Control System: +Pulse Width Control 1520usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage: 4.8-6.0 Volts
Operating Temperature Range: -20 to +60 Degree C
Operating Speed (4.8V): **0.20sec/60 degrees at no load**
Operating Speed (6.0V): **0.16sec/60 degrees at no load**
Stall Torque (4.8V): **72 oz/in. (5.2kg.cm)**
Stall Torque (6.0V): **90 oz/in. (6.5kg.cm)**
Operating Angle: 45 Deg. one side pulse traveling 400usec
360 Modifiable: Yes
Direction: Counter Clockwise/Pulse Traveling 1520-1900usec
Motor Type: 3 Pole Ferrite
Potentiometer Drive: Indirect Drive
Bearing Type: Top Ball Bearing
Gear Type: All Nylon Gears
Connector Wire Length: 12"
Dimensions: 1.6" x 0.8"x 1.5" (41 x 20 x 38mm)
Weight: 1.4oz. (41g)

Reference:
Servocity, “S3010 Std. HT Ball Bearing”, [Online Document], Available HTTP: http://www.servocity.com/html/s3010_std__ht_ball_bearing.html

Appendix F

Specification for wheel and tires

Figure: Wheel with tire

part #: R167-RED-BLACK-WHEEL

weight: 0.05 lbs.

Description:

Here is an easy way to mount wheels to your servos. This molded plastic wheel comes in red with a black traction band. The wheel is 2-3/4" across and 5/16" thick. This wheel will fit Futaba as well as our Standard Servo and Continuous Modified Servo. These do not fit the Hitec servo line.

Reference:

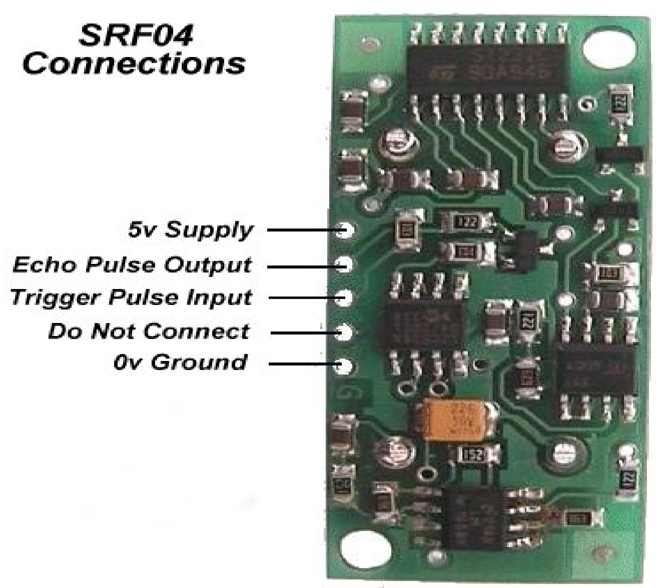
Acroname, “Red wheel with black band”, [Online Document], Available HTTP: http://www.acroname.com/robotics/parts/R167-RED-BLACK-WHEEL.html

Appendix G

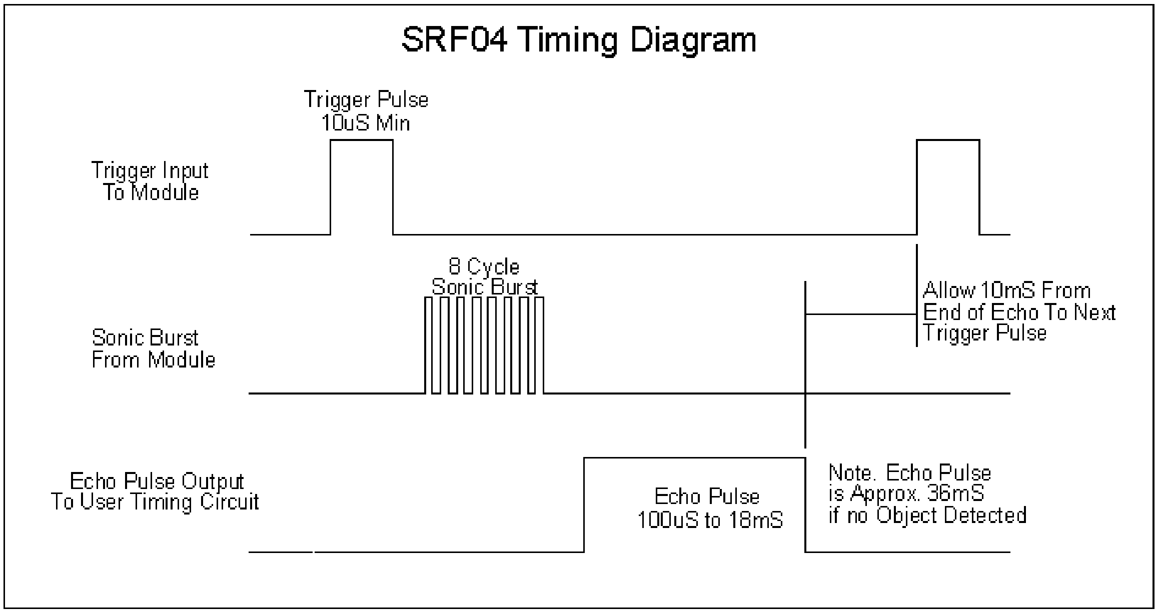
Sensors

Devantech SRF04 Ultra-Sonic Ranger

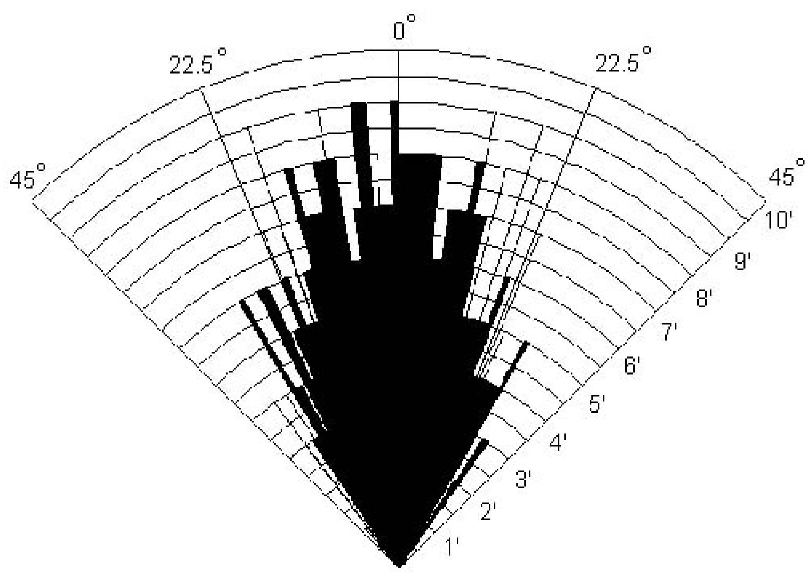
Devantech SRF04 Ultra Sonic Ranger [1]



Devantech Ranger (Back View) [2]

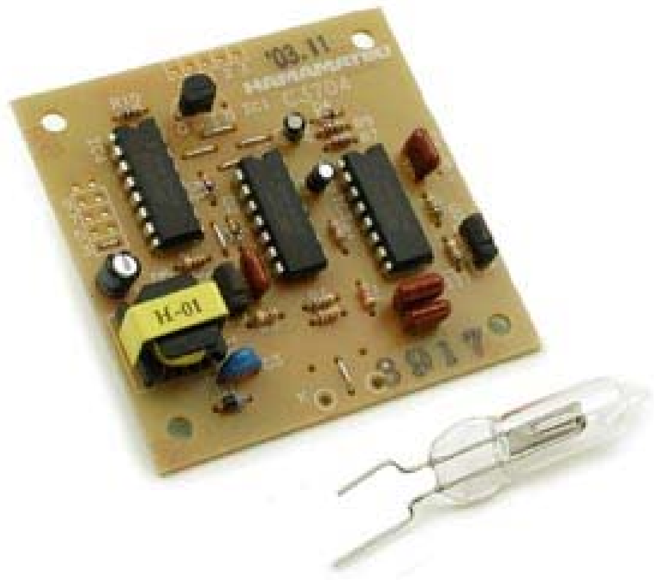


Devantech Ranger Timing Diagram [2]



Devantech Ultra-Sonic Rangefinder Beam Pattern [1]

UVTron Flame Detector



UVTron Flame Detector [1]

The following four pages are the data sheets for the UVTron Flame Sensor and UVTron Drive Circuit [1].

HAMAMATSU

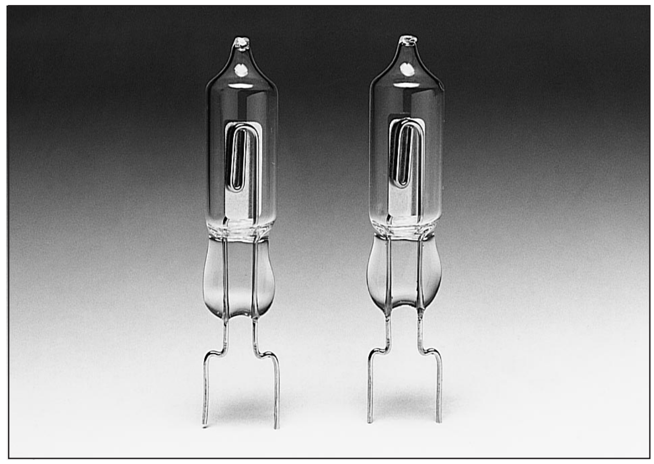
FLAME SENSOR
UV TRON® R2868

Quick Detection of Flame from Distance, Compact UV Sensor with High Sensitivity and Wide Directivity, Suitable for Flame Detectors and Fire Alarms.

Hamamatsu R2868 is a UV TRON ultraviolet detector that makes use of the photoelectric effect of metal and the gas multiplication effect. It has a narrow spectral sensitivity of 185 to 260 nm, being completely insensitive to visible light. Unlike semiconductor detectors, it does not require optical visible-cut filters, thus making it easy to use.

In spite of its small size, the R2868 has wide angular sensitivity (directivity) and can reliably and quickly detect weak ultraviolet radiations emitted from flame due to use of the metal plate cathode (eg. it can detect the flame of a cigarette lighter at a distance of more than 5 m.).

The R2868 is well suited for use in flame detectors and fire alarms, and also in detection of invisible discharge phenomena such as corona discharge of high-voltage transmission lines.



APPLICATIONS

- **Flame detectors for gas/oil lighters and matches**
- **Fire alarms**
- **Combustion monitors for burners**
- **Inspection of ultraviolet leakage**
- **Detection of discharge**
- **Ultraviolet switching**

GENERAL

Parameters	Rating	Units
Spectral Response	185 to 260	nm
Window Material	UV glass	—
Weight	Approx. 1.5	g
Dimensional Outline	See Fig. 3	—

MAXIMUM RATINGS

Parameters	Rating	Units
Supply Voltage	400	Vdc
Peak Current 1)	30	mA
Average Discharge Current 2)	1	mA
Operating Temperature	-20 to +60	°C

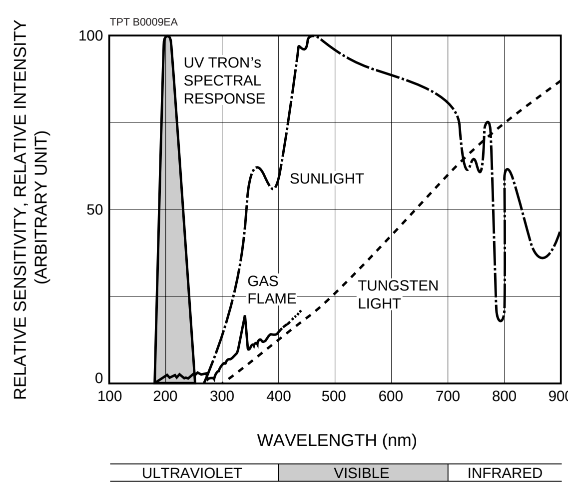
CHARACTERISTICS (at 25°C)

Parameters	Rating	Units
Discharge Starting Voltage (with UV radiation)	280	Vdc Max.
Recommended Operating Voltage	325±25	Vdc
Recommended Average Discharge Current	100	µA
Background 3)	10	cpm Max
Sensitivity 4)	5000	cpm Typ.

NOTES:

1) This is the maximum momentary current that can be handled if its full width at half maximum is less than 10 µs.
2) If the tube is operated near this or higher, the service life is noticeably reduced. Use the tube within the recommended current values.
3) Measured under room illuminations (approximately 500 lux) and recommended operating conditions. Note that these values may increase if the following environmental factors are present.
   1. Mercury lamps, sterilization lamps, or halogen lamps are located nearby.
   2. Direct or reflected sunlight is incident on the tube.
   3. Electrical sparks such as welding sparks are present.
   4. Radiation sources are present.
   5. High electric field (including static field) generates across the tube.
4) These are representative values for a wavelength of 200 nm and a light input of 10 pW/cm^2. In actual use, the sensitivity will vary with the wavelength of the ultraviolet radiation and the drive circuitry employed.

Figure 1: UV TRON's Spectral Response and Various Light Sources



FLAME SENSOR UV TRON® R2868

Figure 2: Angular Sensitivity (Directivity)

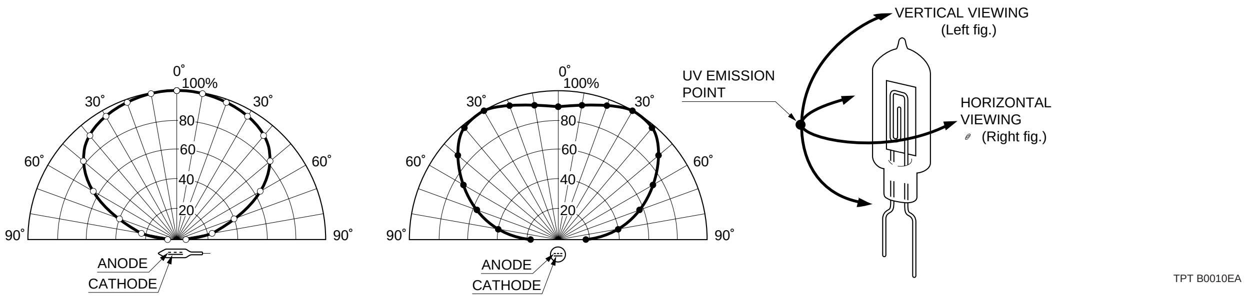


Figure 3: Dimensional Outline (Unit: mm)

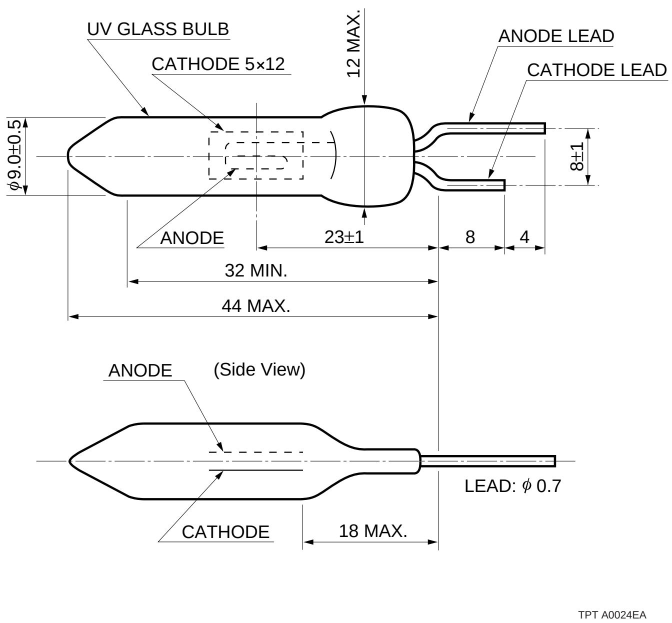
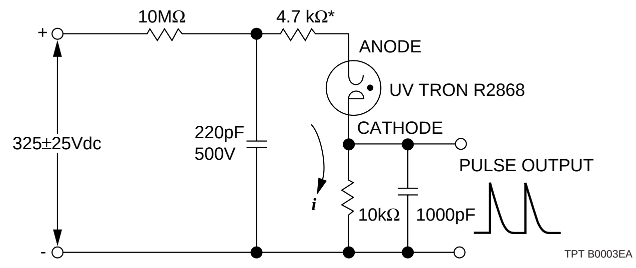
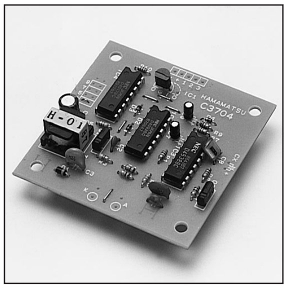


Figure 4: Recommended Operating Circuit



* Be sure to connect the 4.7 kΩ resistor within 2.5 cm from the anode lead end of UV TRON.

- UV TRON Driving Circuit C3704 series (Option)



Hamamatsu also provide the driving circuit C3704 series for R2868 operation. C3704 series include a high voltage power supply and a signal processing circuit in printed circuit board, which allows to operate R2868 easily as a flame sensor with the low input voltage (DC 6 to 30 V) only.
For the details, please refer to the datasheet of C3704 series.

PRECAUTIONS FOR USE

• Ultraviolet Radiation

The UV TRON itself emits ultraviolet radiation in operation. When using two or more UV TRONs at the same time in close position, care should be taken so that they do not optically interfere with each other.

• Vibration and Shock

The UV TRON is designed in accordance with the standards of MIL-STD-202F (Method 204D/0.06 inch or 10g, 10- 500Hz, 15 minutes, 1 cycle) and MIL-STD-202F (Method 213B/100g, 11ms, Half-sine, 3 times). However, should a strong shock be sustained by the UV TRON (e.g. if dropped), the glass bulb may crack or the internal electrode may be deformed, resulting in deterioration of electrical characteristics. So extreme care should be taken in handling the tube.

• Polarity

Connect the UV TRON with correct polarity. Should it be connected with reverse polarity, operating errors may occur.

WARRANTY.

The UV TRON is covered by a warranty for a period of one year after delivery. The warranty is limited to replacement of any defective tube due to defects traceable to the manufacturer.

HAMAMATSU

HAMAMATSU PHOTONICS K.K., Electron Tube Center
314-5, Shimokanzo, Toyooka-village, Iwata-gun, Shizuoka-ken, 438-0193, Japan, Telephone: (81)539/62-5248, Fax: (81)539/62-2205,

U.S.A.: Hamamatsu Corporation: 360 Foothill Road, Bridgewater, N.J. 08807-0910, U.S.A., Telephone: (1)908-231-0960, Fax: (1)908-231-1218
Germany: Hamamatsu Photonics Deutschland GmbH: Arzbergerstr. 10, D-82211 Herrsching am Ammersee, Germany, Telephone: (49)8152-375-0, Fax: (49)8152-2658
France: Hamamatsu Photonics France S.A.R.L.: 8, Rue du Saule Trapu, Parc du Moulin de Massy, 91882 Massy Cedex, France, Telephone: (33)1 69 53 71 00, Fax: (33)1 69 53 71 10
United Kingdom: Hamamatsu Photonics UK Limited: Lough Point, 2 Gladbeck Way, Windmill Hill, Enfield, Middlesex EN2 7JA, United Kingdom, Telephone: (44)181-367-3560, Fax: (44)181-367-6384
North Europe: Hamamatsu Photonics Norden AB: Färögatan 7, S-164-40 Kista, Sweden, Telephone: (46)8-703-29-50, Fax: (46)8-750-58-95
Italy: Hamamatsu Photonics Italia S.R.L.: Via Della Moia, 1/E, 20020 Arese, (Milano), Italy, Telephone: (39)2-935 81 733, Fax: (39)2-935 81 741

TPT 1008E01
MAR.1998 CR
Created in Japan

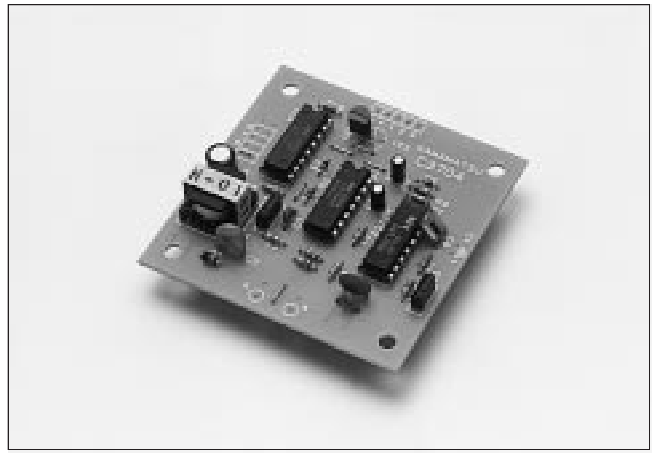
UV TRON® DRIVING CIRCUIT C3704 SERIES

Compact, Lightweight, Low Current Consumption, Low Cost
Operates as High Sensitivity UV Sensor with UV TRON
Suitable for Flame Detectors and Fire Alarms

Hamamatsu C3704 series UV TRON driving circuits are low current consuming, signal processing circuits for the UV TRON, well known as a high sensitivity ultraviolet detecting tube. The C3704 series can be operated as a UV sensor by connecting the UV TRON and applying DC low voltage, as they have both a high-voltage power supply and a signal processing circuit on the same printed circuit board.

Since background discharges of the UV TRON caused by natural excitation lights (such as a cosmic ray, scattered sunlight, etc.) can be cancelled in the signal processing circuit, the output signals from the C3704 series can be used without errors.

When the high sensitivity sensor "UV TRON R2868" (sold separately) is used, the flame from a cigarette lighter (flame length: 25mm) can be detected even from a distance of more than 5m.



APPLICATIONS

- **Flame detectors for gas and oil lighters**
- **Fire alarms**
- **Combustion monitors for burners**
- **Electric spark detector**
- **UV photoelectric counter**

SPECIFICATIONS

Dimensional outline .... Figure 1
Weight .... Approx. 20g
Output signal .... Open collector Output (50 V, 100 mA Max.)
10 ms width pulse output (Note : 1)
UV TRON supply voltage .... DC 350 V (Note : 2)
Quenching time .... Approx. 50 ms
Operating temperature .... -10 to +50°C
(with no condensation)
Suitable UV TRON .... Low voltage operation UV TRON
(such as R2868)

	C3704	C3704-02	C3704-03
Input Voltage	10 to 30 Vdc	5Vdc ± 5%	6 to 9 Vdc
Current consumption	3 mA Max.	300µA Max.	300µA Max.

Note 1: The output pulse width can be extended up to about 100s by adding a capacitor to the circuit board.

Note 2: Since the output impedance of this power supply is extremely high, an ordinary voltmeter cannot be used. Use a voltmeter that has an input impedance of more than 10 GΩ.

Figure 1: Dimensional Outline (Unit : mm)

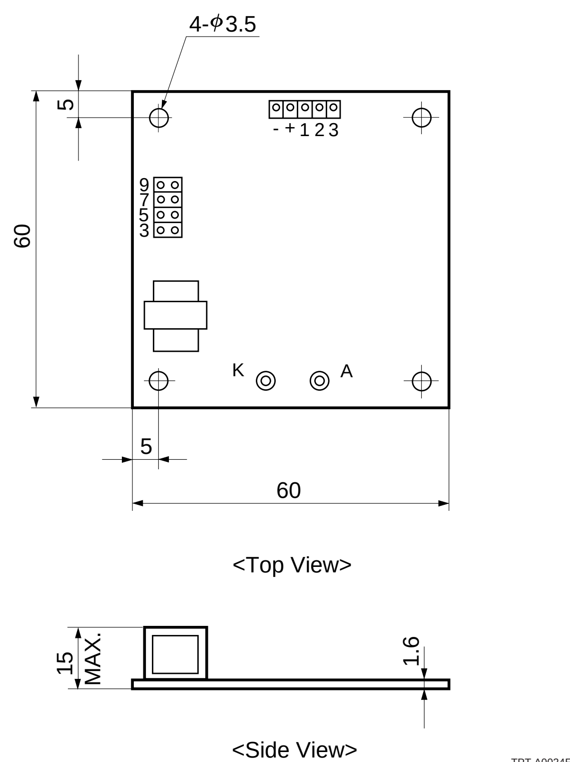


Figure 2: Schematic Diagram

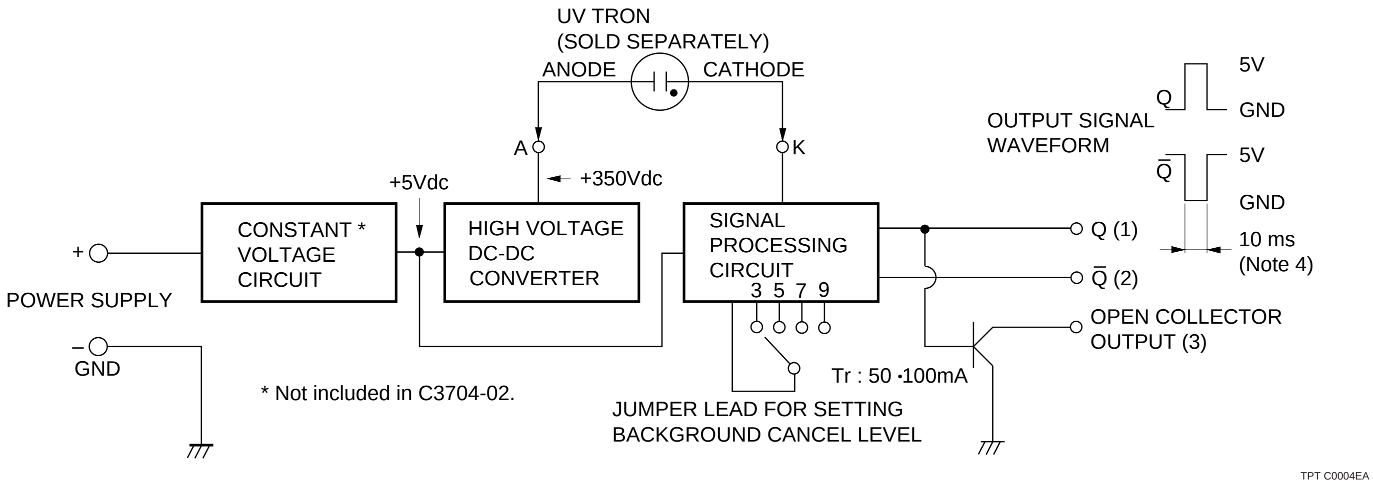
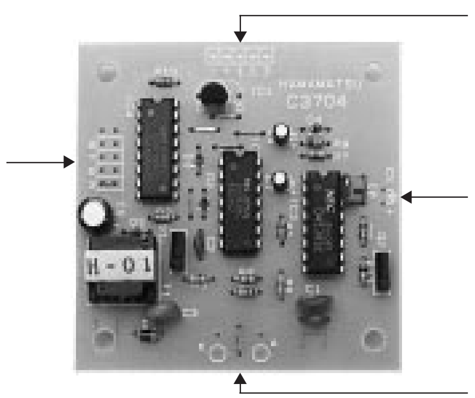


Figure 3: Method of Connection



This background cancel circuit outputs signal pulses of 10ms width only when 3 to 9 pulses enter the circuit with a time interval of 2 seconds or less from the UV TRON. Cancel level can be set in 4 steps (number of pulses: 3, 5, 7, 9) by this jumper lead. Connect to “3” for general use. When using where much natural excitation light is present, set the cancel level at a higher step (5, 7, 9).

Note 3: No load can be driven by an output from points “1” and “2” because these signals are output from the only C-MOS IC directly. When a load such as a buzzer and a relay is connected to this circuit, it should be connected to the point open collector output. The transistor ratings of the open collector is 50V, 100mA. Be careful not to exceed the ratings.

Note 4: The output pulse width is set to 10ms at shipping. If the pulse width needs to be extended, add a capacitor to this point. (When using an electrolytic condenser, make sure the polarity is correct.)
e.g. CX = 1 μF: Pulse Width ≃ 1s, CX=10 μF : Pulse Width ≃ 10s

PRECAUTIONS FOR USE

- Since the operation impedance is extremely high, the UV TRON should be connected as close as possible to the circuit board within 5 cm.
- Take care to avoid external noise since a C-MOS IC is used in the circuit. It is recommended that the whole PC board be put in the shield box when it is used.
- To reduce current consumption, oscillating frequency is very low (approx. 20 Hz) in this DC-DC converter. Thus, the output impedance of the high voltage power supply is extremely high. If the surrounding humidity is high, electrical leakage on the PC board surface may lead to a drop in the supply voltage to the UV TRON. This voltage drop may result in lowered detection performance, so a moistureproof material (silicone compound, etc.) should be applied at the connecting point of the UV TRON, etc., if using the unit in a humid environment.

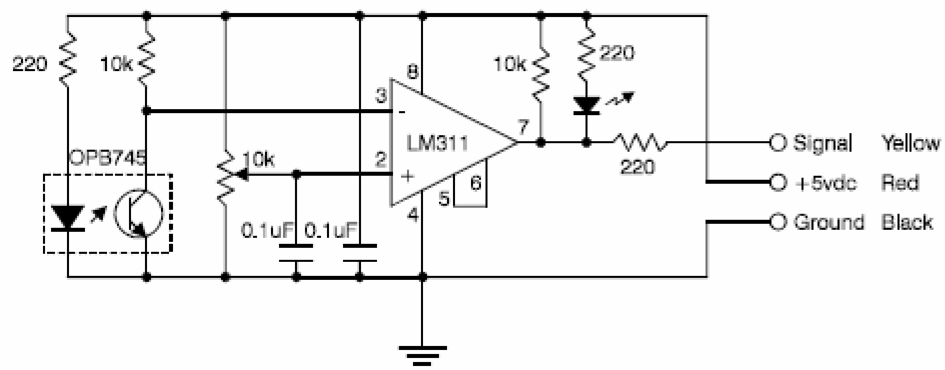
- A model equipped with a flame sensor (R2868) is also available.

HAMAMATSU

HAMAMATSU PHOTONICS K.K., Electron Tube Center
314-5, Shimokanzo, Toyooka-village, Iwata-gun, Shizuoka-ken, 438-0193, Japan, Telephone: (81)539/62-5248, Fax: (81)539/62-2205, Telex: 4225-186HAMAHQ
U.S.A.: Hamamatsu Corporation: 360 Foothill Road, Bridgewater, N.J. 08807-0910, U.S.A., Telephone: (1)908-231-0960, Fax: (1)908-231-1218
Germany: Hamamatsu Photonics Deutschland GmbH: Arzbergerstr. 10, D-82211 Herrsching am Ammersee, Germany, Telephone: (49)8152-375-0, Fax: (49)8152-2658
France: Hamamatsu Photonics France S.A.R.L.: 8, Rue du Saule Trapu, Parc du Moulin de Massy, 91882 Massy Cedex, France, Telephone: (33)1 69 53 71 00, Fax: (33)1 69 53 71 10
United Kingdom: Hamamatsu Photonics UK Limited: Lough Point, 2 Gladbeck Way, Windmill Hill, Enfield, Middlesex EN2 7JA, United Kingdom, Telephone: (44)181-367-3560, Fax: (44)181-367-6384
North Europe: Hamamatsu Photonics Norden AB: Färögatan 7, S-164-40 Kista, Sweden, Telephone: (46)8-703-29-50, Fax: (46)8-750-58-95
Italy: Hamamatsu Photonics Italia S.R.L.: Via Della Moia, 1/E 20020 Arese, (Milano), Italy, Telephone: (39)2-935 81 733, Fax: (39)2-935 81 741

TPT1007E01
JUL.1997 CR
Created in Japan

White Line Sensor

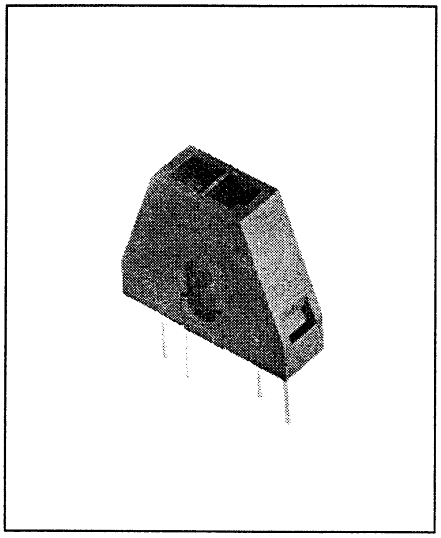


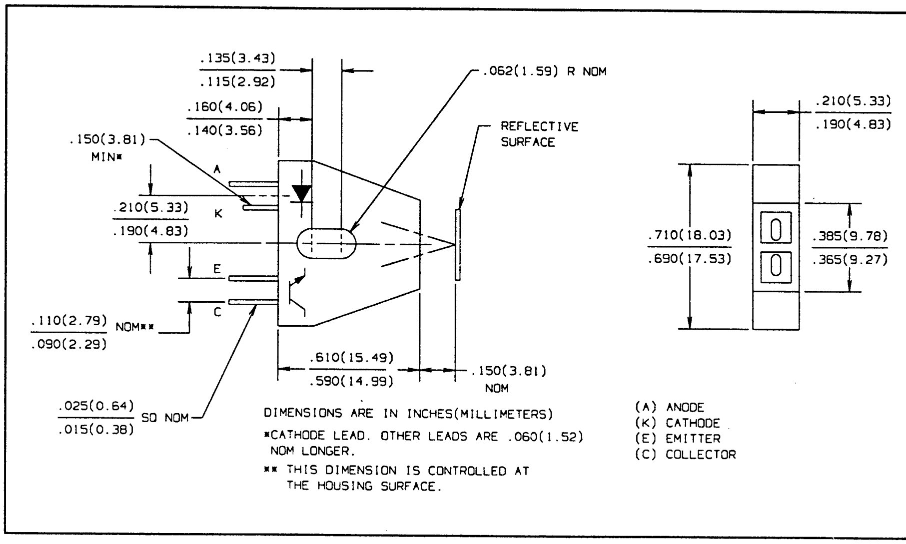
White Line Sensor Circuit Diagram [3]

The following two pages are the data sheet for the Optek OPB745 [4]

Product Bulletin OPB745
June 1996

Reflective Object Sensor
Type OPB745





Features

- Focused for maximum sensitivity
- Photodarlington output
- Crosstalk does not exceed specified I_{CEO}
- PC board mounting

Description

The OPB745 reflective object sensor consists of an infrared emitting diode and an NPN silicon photodarlington mounted side by side on converging optical axes in a black plastic housing.

The photodarlington responds to radiation from the emitter only when a reflective object passes within its field of view.

Absolute Maximum Ratings (T_A = 25° C unless otherwise noted)

Storage and Operating Temperature Range . . . . -40° C to +85° C
Lead Soldering Temperature [1/16 inch (1.6 mm) from case for 5 sec. with soldering iron] . . . . 240° C(1)

Input Diode

Continuous Forward Current . . . . 40 mA
Reverse Voltage . . . . 2.0 V
Power Dissipation . . . . 100 mW(2)

Output Photodarlington

Collector-Emitter Voltage . . . . 15 V
Emitter-Collector Voltage . . . . 5.0 V
Power Dissipation . . . . 100 mW(2)

Notes:
(1) RMA flux is recommended. Duration can be extended to 10 sec. max. when flow soldering.
(2) Derate Linearly 1.67 mW/° C above 25° C.
(3) d is distance from the assembly face to the reflective surface.
(4) Reflective surface is Eastman Kodak neutral white test card with 90% diffuse reflectance as a reflecting surface. Reference: Eastman Kodak, Catalog #1257795.
(5) Lower curve is based on calculated worst case condition rather than the conventional -2σ limit.
(6) Crosstalk is the photocurrent measured with current to the input diode & no reflecting surface.
(7) All parameters tested using pulse technique.

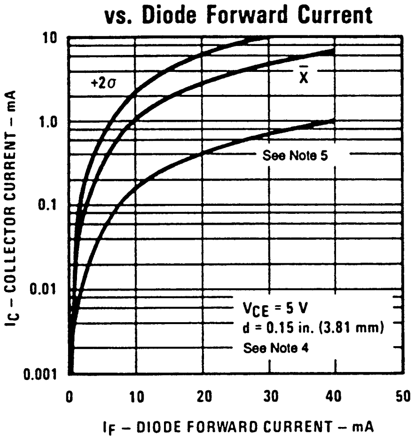
Type OPB745

Electrical Characteristics ($T_A = 25^\circ$ C unless otherwise noted)

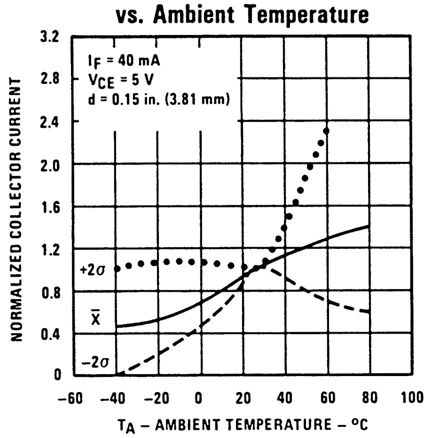
SYMBOL	PARAMETER	MIN	MAX	UNITS	TEST CONDITIONS
Input Diode					
V_F	Forward Voltage		1.70	V	$I_F = 40$ mA
I_R	Reverse Current		100	µA	$V_R = 2.0$ V
Output Photodarlington					
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage	15		V	$I_C = 100$ µA
$V_{(BR)ECO}$	Emitter-Collector Breakdown Voltage	5.0		V	$I_E = 100$ µA
I_{CEO}	Collector Dark Current		250	nA	$V_{CE} = 10$ V, $I_F = 0$, $E_e = 0$
Combined					
$I_{C(ON)}$(3)(4)	On-State Collector Current	1.0		mA	$V_{CE} = 5$ V, $I_F = 40$ mA, d = 0.15"
I_{CX}(2)	Crosstalk		250	nA	$V_{CC} = 5$ V, $I_F = 40$ mA

Typical Performance Curves

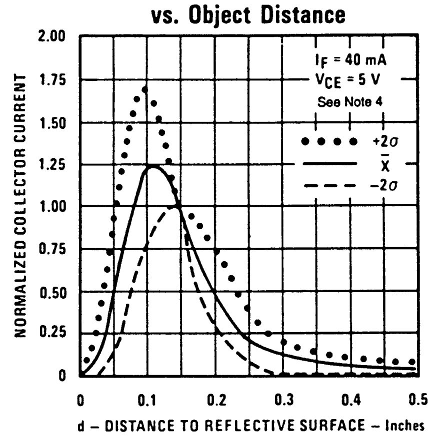
Collector Current vs. Diode Forward Current



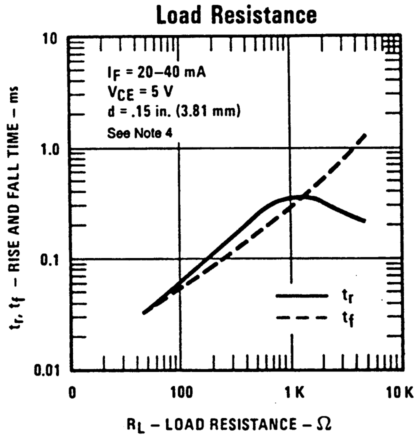
Normalized Collector Current vs. Ambient Temperature



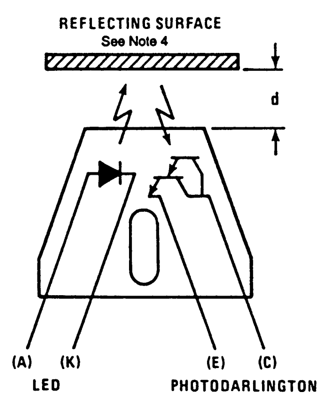
Normalized Collector Current vs. Object Distance



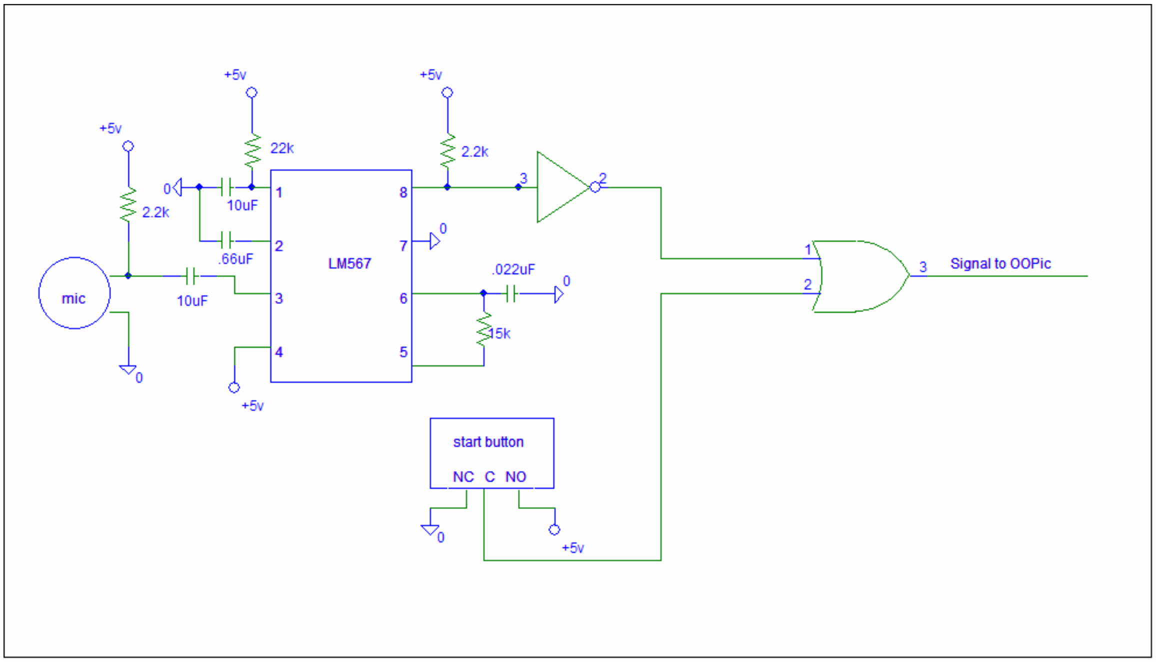
Rise and Fall Time vs. Load Resistance



Test Condition



Tone Decoder



Activation Circuit Schematic

The following pages are the data sheet for the LM567 [5].

National Semiconductor

May 1999

LM567/LM567C
Tone Decoder

General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

Features

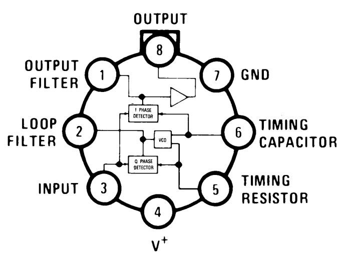
- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%
- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

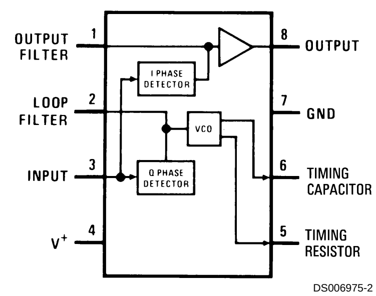
Connection Diagrams

Metal Can Package



Top View
Order Number LM567H or LM567CH
See NS Package Number H08C

Dual-In-Line and Small Outline Packages



Top View
Order Number LM567CM
See NS Package Number M08A
Order Number LM567CN
See NS Package Number N08E

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage Pin	9V
Power Dissipation (Note 2)	1100 mW
V_8	15V
V_3	−10V
V_3	V_4 + 0.5V
Storage Temperature Range	−65˚C to +150˚C

Operating Temperature Range	
LM567H	−55˚C to +125˚C
LM567CH, LM567CM, LM567CN	0˚C to +70˚C
Soldering Information	
Dual-In-Line Package	
Soldering (10 sec.)	260˚C
Small Outline Package	
Vapor Phase (60 sec.)	215˚C
Infrared (15 sec.)	220˚C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics

AC Test Circuit, T_A = 25˚C, V^+ = 5V

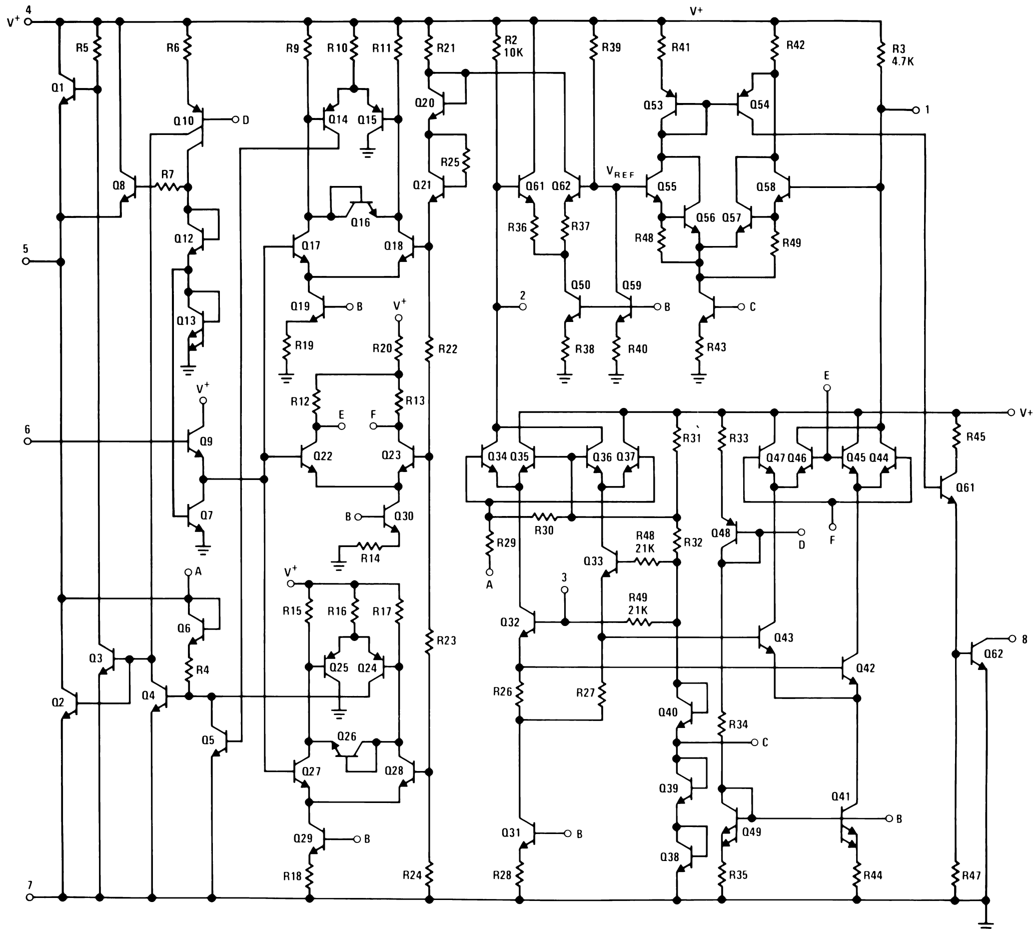
Parameters	Conditions	LM567			LM567C/LM567CM			Units
		Min	Typ	Max	Min	Typ	Max	
Power Supply Voltage Range		4.75	5.0	9.0	4.75	5.0	9.0	V
Power Supply Current Quiescent	R_L = 20k		6	8		7	10	mA
Power Supply Current Activated	R_L = 20k		11	13		12	15	mA
Input Resistance		18	20		15	20		kΩ
Smallest Detectable Input Voltage	I_L = 100 mA, $f_i = f_o$		20	25		20	25	mVrms
Largest No Output Input Voltage	I_C = 100 mA, $f_i = f_o$	10	15		10	15		mVrms
Largest Simultaneous Outband Signal to Inband Signal Ratio			6			6		dB
Minimum Input Signal to Wideband Noise Ratio	B_n = 140 kHz		−6			−6		dB
Largest Detection Bandwidth		12	14	16	10	14	18	% of f_o
Largest Detection Bandwidth Skew			1	2		2	3	% of f_o
Largest Detection Bandwidth Variation with Temperature			±0.1			±0.1		%/˚C
Largest Detection Bandwidth Variation with Supply Voltage	4.75–6.75V		±1	±2		±1	±5	%/V
Highest Center Frequency		100	500		100	500		kHz
Center Frequency Stability (4.75–5.75V)	$0 < T_A < 70$ $-55 < T_A < +125$		35 ± 60 35 ± 140			35 ± 60 35 ± 140		ppm/˚C ppm/˚C
Center Frequency Shift with Supply Voltage	4.75V–6.75V 4.75V–9V		0.5	1.0 2.0		0.4	2.0 2.0	%/V %/V
Fastest ON-OFF Cycling Rate			f_o/20			f_o/20		
Output Leakage Current	V_8 = 15V		0.01	25		0.01	25	µA
Output Saturation Voltage	e_i = 25 mV, I_8 = 30 mA e_i = 25 mV, I_8 = 100 mA		0.2 0.6	0.4 1.0		0.2 0.6	0.4 1.0	V
Output Fall Time			30			30		ns
Output Rise Time			150			150		ns

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 2: The maximum junction temperature of the LM567 and LM567C is 150˚C. For operating at elevated temperatures, devices in the TO-5 package must be derated based on a thermal resistance of 150˚C/W, junction to ambient or 45˚C/W, junction to case. For the DIP the device must be derated based on a thermal resistance of 110˚C/W, junction to ambient. For the Small Outline package, the device must be derated based on a thermal resistance of 160˚C/W, junction to ambient.

Note 3: Refer to RETS567X drawing for specifications of military LM567H version.

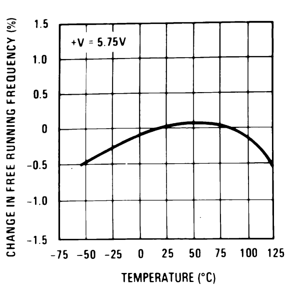
Schematic Diagram



DS006975-3

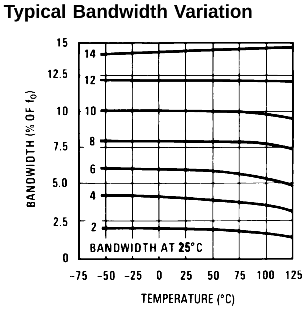
Typical Performance Characteristics

Typical Frequency Drift



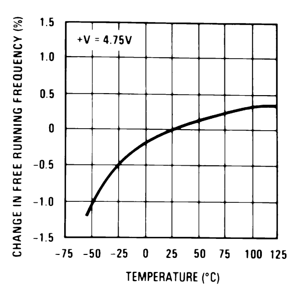
DS006975-10

Typical Bandwidth Variation



DS006975-11

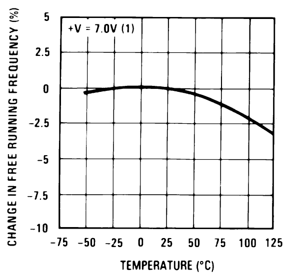
Typical Frequency Drift



DS006975-12

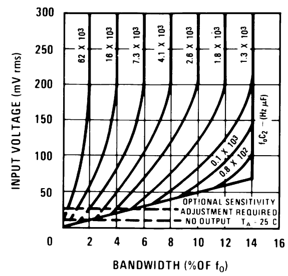
Typical Performance Characteristics (Continued)

Typical Frequency Drift



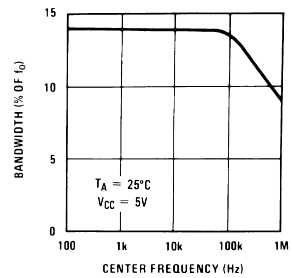
DS006975-13

Bandwidth vs Input Signal Amplitude



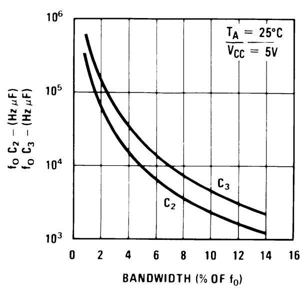
DS006975-14

Largest Detection Bandwidth



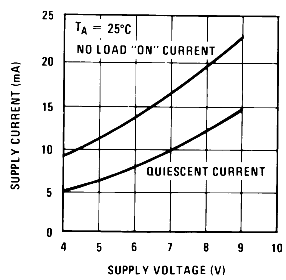
DS006975-15

Detection Bandwidth as a Function of C_2 and C_3



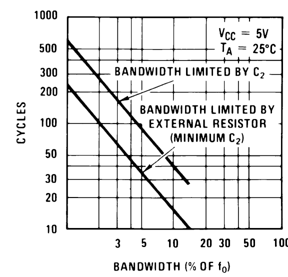
DS006975-16

Typical Supply Current vs Supply Voltage



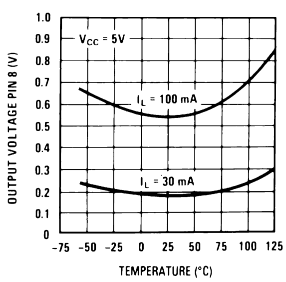
DS006975-17

Greatest Number of Cycles Before Output



DS006975-18

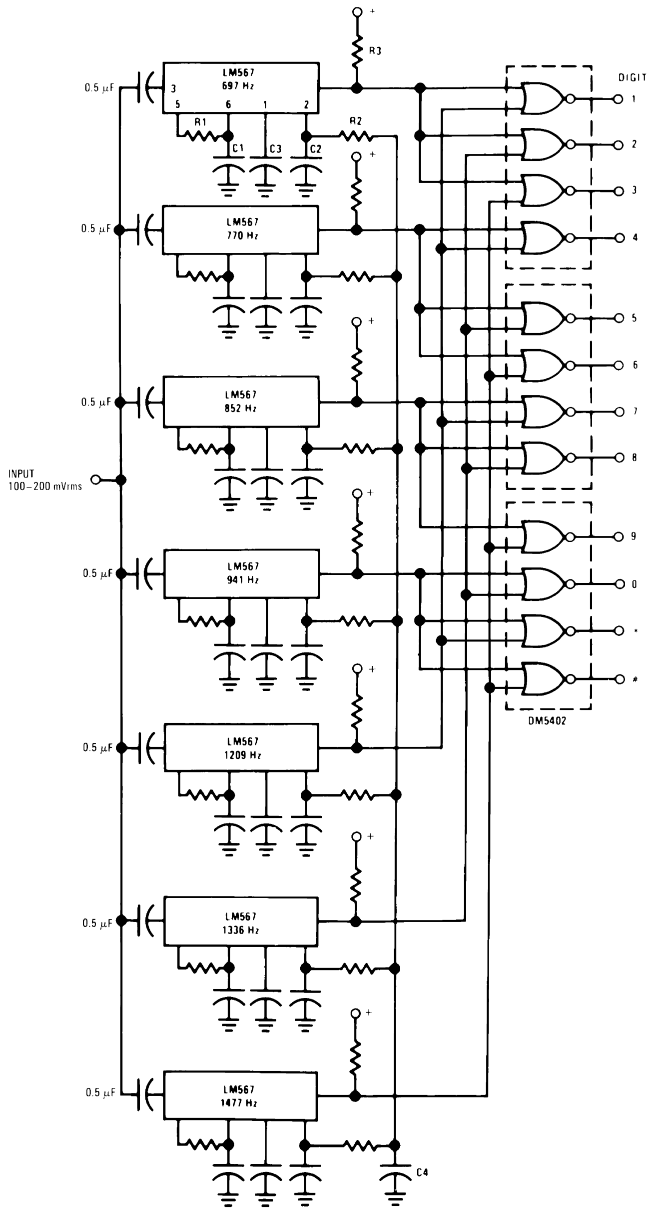
Typical Output Voltage vs Temperature



DS006975-19

Typical Applications

Touch-Tone Decoder



DS006975-5

Component values (typ)

R1 6.8 to 15k

R2 4.7k

R3 20k

C1 0.10 mfd

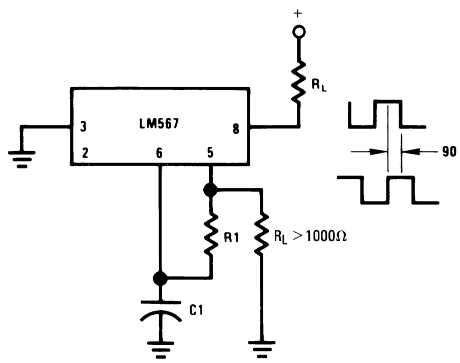
C2 1.0 mfd 6V

C3 2.2 mfd 6V

C4 250 mfd 6V

Typical Applications (Continued)

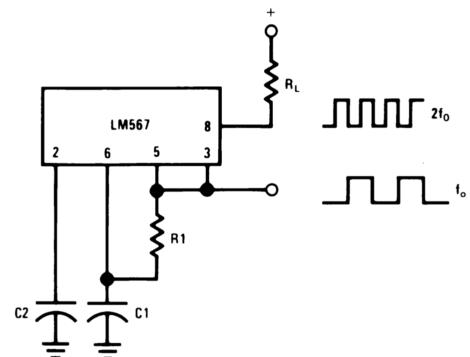
Oscillator with Quadrature Output



DS006975-6

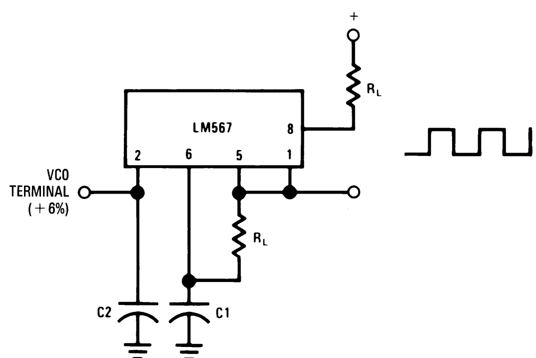
Connect Pin 3 to 2.8V to Invert Output

Oscillator with Double Frequency Output



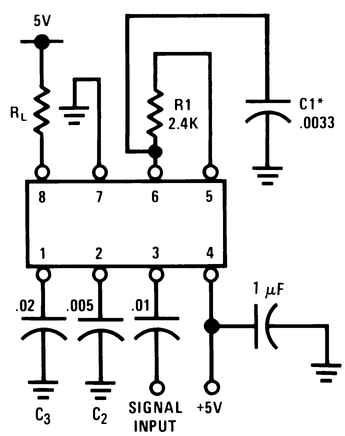
DS006975-7

Precision Oscillator Drive 100 mA Loads



DS006975-8

AC Test Circuit



DS006975-9

f_i = 100 kHz + 5V

***Note:** Adjust for f_o = 100 kHz.

Applications Information

The center frequency of the tone decoder is equal to the free running frequency of the VCO. This is given by

$$f_o \cong \frac{1}{1.1\ R_1 C_1}$$

The bandwidth of the filter may be found from the approximation

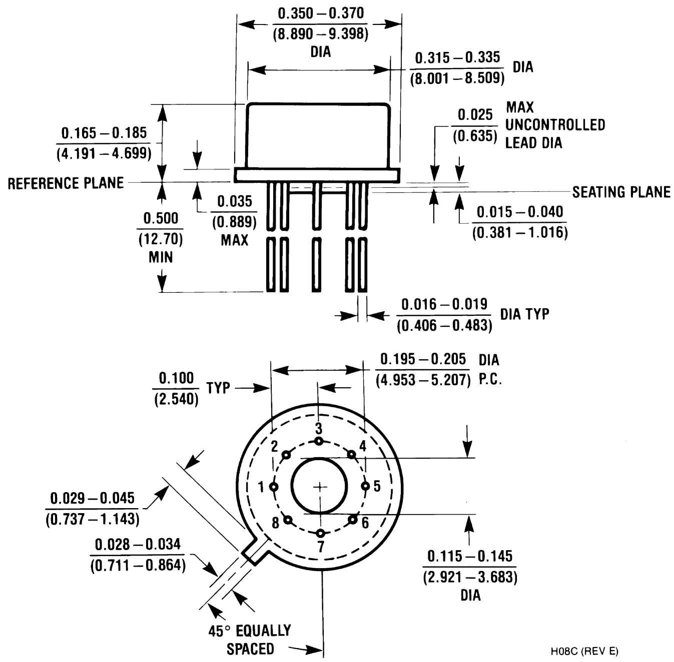
$$BW = 1070 \sqrt{\frac{V_i}{f_o C_2}} \text{ in \% of } f_o$$

Where:

V_i = Input voltage (volts rms), $V_i \leq 200$ mV

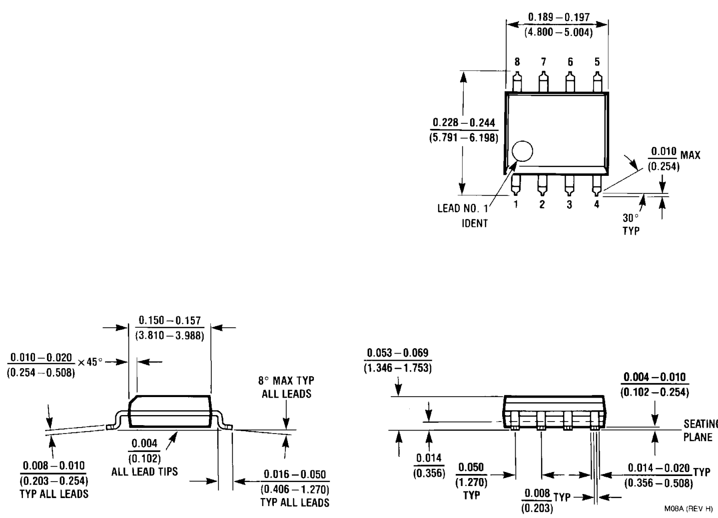
C_2 = Capacitance at Pin 2 (µF)

Physical Dimensions inches (millimeters) unless otherwise noted



H08C (REV E)

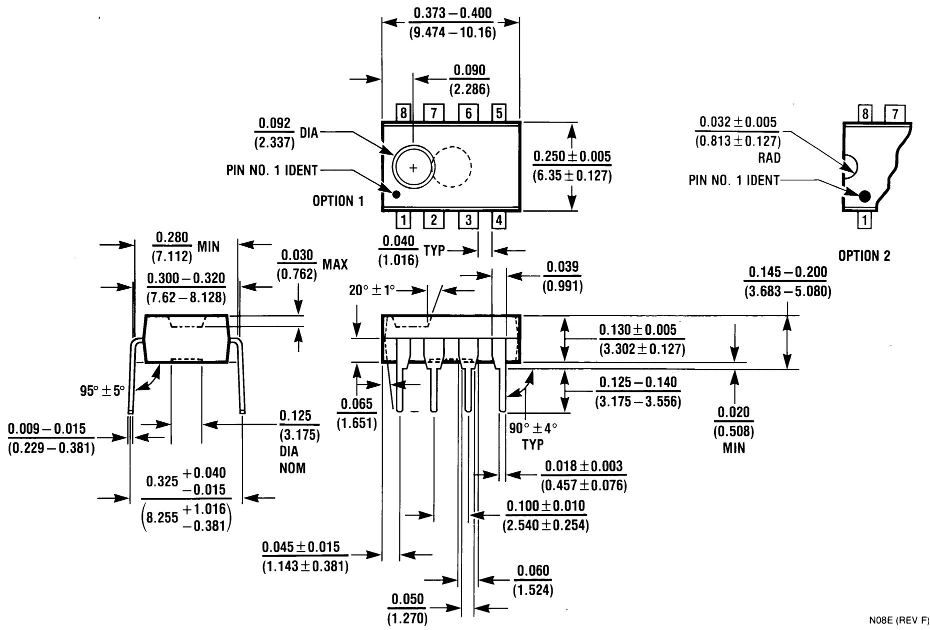
Metal Can Package (H)
Order Number LM567H or LM567CH
NS Package Number H08C



M08A (REV H)

Small Outline Package (M)
Order Number LM567CM
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Dual-In-Line Package (N)
Order Number LM567CN
NS Package Number N08E

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

www.national.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

Appendix G References

[1] Acroname, "Sensors." [online] 2003, [2004 January 23], Available at http://www.acroname.com/robotics/parts/c_Sensors.html

[2] Robot Electronics, "SRF04 - Ultra-Sonic Ranger Technical Specification," [Online] May 2003, [2004 January 23], Available at http://www.robot-electronics.co.uk/htm/srf04tech.html

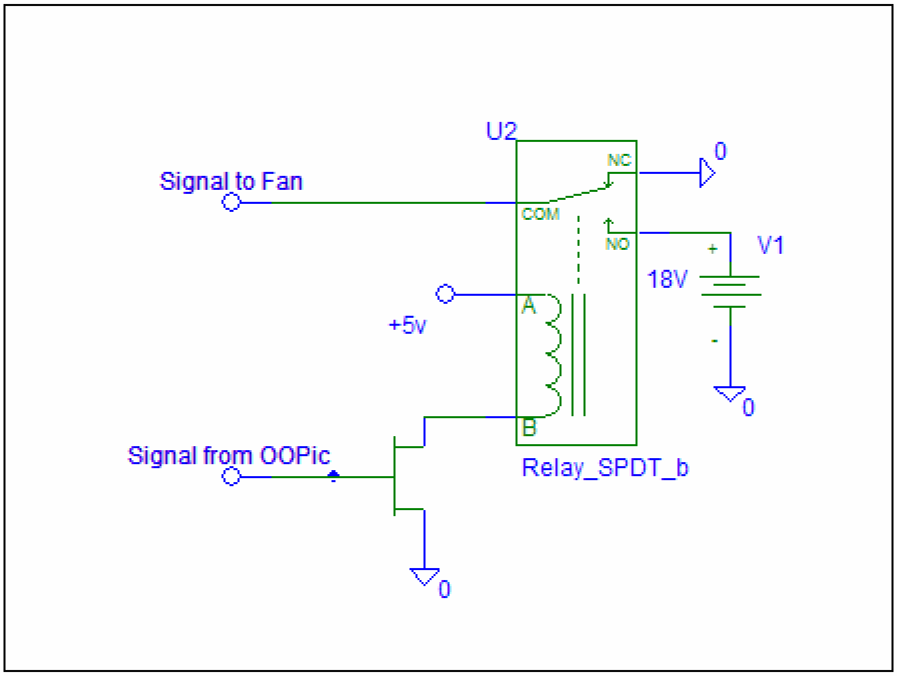
[3] Lynxmotin, Inc, "Single Line Detector," [online] 2003, [2004 February 4], Available at http://www.lynxmotion.com/Product.aspx?productID=58&CategoryID=8

[4] Optek, "Reflective Object Sensor, Type OPB745," [online] June 1996, [2004 February 4], Available at http://www.optekinc.com/pdf/OPB745.pdf.

[5] National Semiconductor, "LM567/LM567C Tone Decoder," [online] 1999, [2004 February 10], Available at http://cache.national.com/ds/LM/LM567.pdf.

Appendix H

Switching Circuit for Fan

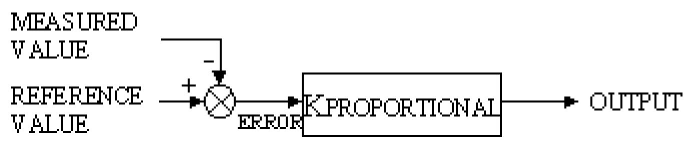


Switching Circuit for Powering Fan

Appendix I

Proportional Control

The first and most basic part is the P, for proportional control. All this means is that if you have a reference you are trying to control to, you provide a control output proportional to the error from your reference.



PROPORTIONAL CONTROL

(note: K is commonly used to represent such gain factors, and is usually given with a subscript indicating which gain is meant. E.g . $K_{steering}$)

As a standard robot example , if you are trying to follow a wall using a sensor that can measure the distance to the wall (e.g. sonar), you would turn your steering system to return you to the reference distance from the wall. If you are 2 inches to the right, you might turn your steering 4 degrees to the left. If 4 inches to the right, then turn the steering 8 degrees to the left. That is the proportional part.

A block diagram of this operation might look like:

PROPORTIONAL CONTROL FOR STEERING

The drawing indicates that the actual sonar distance is subtracted from the reference distance (the distance from the wall that you want the robot to stay at) . This gives you an error signal showing the distance from where you want to be in inches (or feet or whatever you want). The "O" (with an X in it) where the two signals come together is called a summation point. The "+" indicates that the reference signal is added; and the "-" means that the sonar sensor is subtracted to get the error signal. That is:

error = reference distance – sonar distance.

The steering gain block represents just a multiplier to get the desired amount of steering per inch of error. The whole thing can be written as:

Steering angle command = Ksteering * (reference distance – sonar distance)

So what this equation does is to steer the robot back toward the reference distance. Sounds like a good idea, but in reality, this may not be enough. What will happen if we try to steer this way?

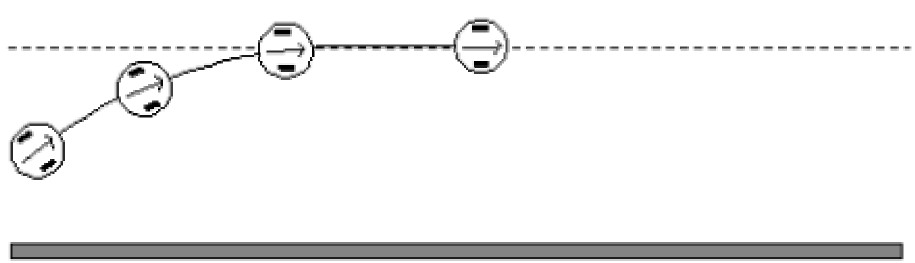
I'm going to use the two Example robot systems that I described on an earlier page. If, you didn't read that page, I'd recommend you do it now.

Differential steering control:

If you have a differentially steering robot which starts out rolling along parallel a wall (on its right side) at a distance of 24 inches from the wall, it will TEND to continue to roll along parallel to the wall (until errors build up). The robot has a sonar or Infrared range sensor which continuously gives it the distance to the wall. Let's say we pick the robot up and put it down again pointed in the same direction (parallel to the wall) but 12 inches closer to the wall.. The proportional equation above will subtract the measured distance from the wall (12 inches) from the reference value (24 inches) getting an error signal of 12 inches. Multiplying this by the gain (Ksteering) gives a steering command of 24 degrees. Obviously, the correction should be to turn left to get back to the 24 inch reference value.

This 24 degree command must be translated into a command to the two drive motors. The 24 degrees is equal to 0.42 radians, and using the distance between the two side wheels of 10 inches, the right wheel should rotate forward 0.42 * 10 = 4.2 inches to turn 24 degrees. This turn can be achieved by adding half of the 4.2 inches to the right motor control distance command and subtracting half from the left motor. This turn will occur very quickly (as fast as the motors can drive the wheels) and I'll assume it is instantaneous.

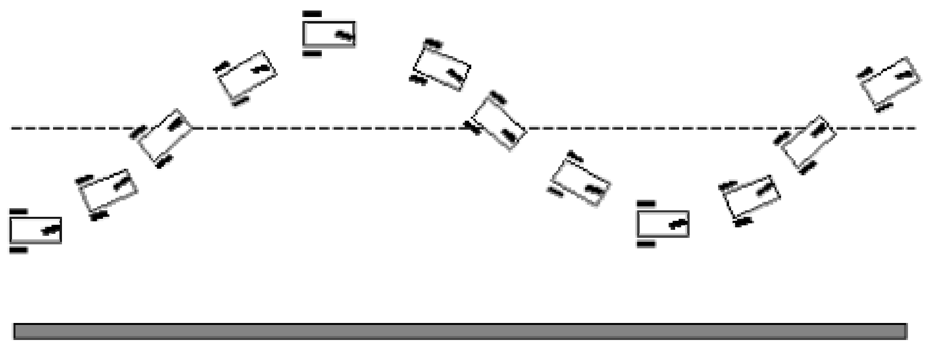
The robot is now pointed back toward its reference distance, and as time goes by, it will get closer and closer. As the error decreases, the steering command decreases proportionately. and as the robot reaches the line, the steering error will be zero and the robot will drive parallel to the wall at the correct distance.



Car type steering command:

Now, what happens if we do the same thing to a robot with car type steering? If the robot is placed 12 inches closer to the wall as was done above, the robot will create the same steering command of 24 degrees. The RC type servo will be instructed to move the front wheel to the 24 degrees to the left position.

As the robot moves forward, it will begin to turn to the left; and hence, it will begin to approach the reference line. As it approaches, the calculated steering command begins to decrease. So, the robot reduces the angle of the steering wheels, but they are still pointed toward the left and the robot continues to turn left...just more slowly. When the robot reaches the reference distance, the steering command goes to zero and the front steering wheels go to center; but the robot is still pointed toward the left and continues past the line. As it goes out to the left of the reference distance, it will begin to turn the steering wheels back to the right. Eventually, it will get back to the line again, but will overshoot it again and go out on the right side.



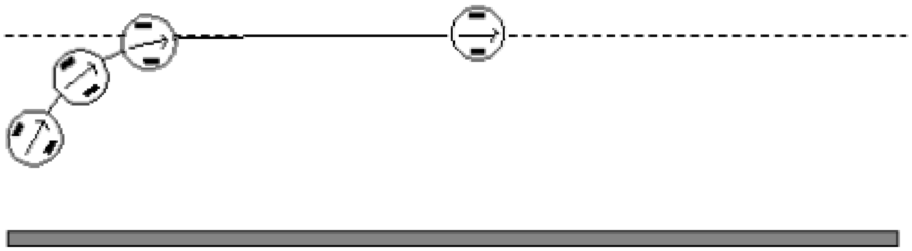
So, what's going on here? Why do the two robots behave differently? The difference is that with the differentially steering robot, your steering command directly controls the angle of the robot relative to the line. With the car steering robot, you are controlling the derivative of the angle of the robot. That is, for a given forward speed, the angle of the robot is the integral of the angle of the front steering wheel. And integrators, as we'll discuss later, tend to be unstable.

The lesson is that the P, I and D equations do not always do the same thing. It depends how you apply them to your robot's control Try to consider what you are controlling when you select the equations, and how the application may affect the results when things aren't working properly. We'll talk about this more later.

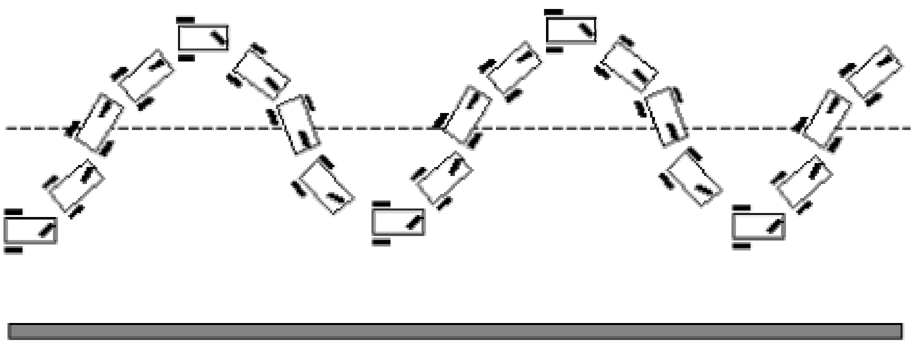
GAIN

The only variable in the proportional equation (after you've calculated the error) is the gain, Ksteering. What is the effect of changing that gain?

Basically, increasing the gain will make things happen more rapidly. Since an increased gain makes the steering of the differential robot point the robot more toward the reference line, it turns more abruptly and gets on the line sooner



For the car type steering, it oscillates back and forth across the line but does it at a higher frequency.



Appendix I References

[1] "Proportional Control", [Online Document], Available HTTP: http://abrobotics.tripod.com/ControlLaws/proportional.htm

Appendix J

Firmware

```
oSonarDV lso = new oSonarDV;          // Left Sonar
oSonarDV rso = new oSonarDV;          // Right Sonar
oSonarDV fso = new oSonarDV;          // Front Sonar
oServoSP1 lse = new oServoSP1;        // Left Servo Motor
oServoSP1 rse = new oServoSP1;        // Right Servo Motor
oUVTronHM fireMeas = New oUVTronHM; // Flame Sensor (Ultraviolet Light with filters)
oLCD LCD = New oLCD;                  // LCD
oDio1 StartBut = new oDio1;           // Start Button and Microphone Start
oDio1 Fans = new oDio1;               // Fans
oDio4 bLineStatus = new oDio4;        // Line Sensors

// Shoaib vars
word error;
word errorlast;
word prop;
word rate;

// Shoaib consts
const refval = 42;
const deltaT = 4;
const KP = 6;

//defining constants
const KA = 9;
const ALIGN = 71;
const MAXBRK = 0;
const TRNSPD = 150;
const MAXSPD = 245; // 35 * SCALE b/c servo MAX is 35.
                        // This also allows for greater control from
                        // proportional constant.  Usually val is 245
const HALL = 100;   // Hall width in tics
const SCALE = 7;        // Divisor for more control over servos
const ALIGNSPD = 10;// Speed when robot is aligning with white
const FLAMESPD = 1; // Speed when searching for flame
const FIRETHRESH = 200; // Trigger over this value of fire on UVTron
const ENTERDIST = 40;   // Distance in tics to move into room
const LCENTER = 10;     // Center of left wheel is off ten?



sub void main(void)
{
       // LCD Initialization
       //LCD.IOLineRS = 26; //RS line to the LCD module
       //LCD.IOLineE = 27;  //E line to the LCD module
       //LCD.IOGroup = 3;   //I/O lines 28-31 (pins 32-26)
       //LCD.Nibble = 1;    //Upper nibble
       //LCD.Operate = cvTrue;
       //OOPic.delay = 3;    //wait For LCD to come up
       //LCD.Init;           //Perform the icky initialization
       //LCD.Clear;


       // Hardware Initialization
       lso.IOLineP = 3;  // RA2
       lso.IOLineE = 4;  // RA3
       rso.IOLineP = 5; // RE0
       rso.IOLineE = 6; // RE1
       fso.IOLineP = 1; // RA0
       fso.IOLineE = 2; // RA1
       lso.Operate = 1;
       rso.Operate = 1;
       fso.Operate = 1;
       lse.IOLine = 14;  // RB6
       rse.IOLine = 15;  // RB7
       lse.Operate = cvTrue;
       rse.Operate = cvTrue;
       rse.InvertOut = cvTrue;
       startBut.IOLine = 24;
       startBut.Direction = cvInput;
       bLineStatus.IOGroup = 1;
```

```
bLineStatus.Nibble = 0;
fireMeas.IOLine = 25;  // RD1
fireMeas.Operate = 0;
Fans.IOLine = 16; // RC0
Fans.Direction = cvOutput;

//LCD.Clear;
//LCD.String = "Alarm or Push Button ->";
while(startBut == 0)
{
  // wait for start button to be pushed.
}
startBut.Direction = cvOutput;

//forward(65, 100, 0);

// Travel to Room A
forward(65, 100, 0);
trnL(36);
forward(165, 100, 0);
trnl(36);
OOPic.Delay = 20;

bLineStatus.Direction = cvInput;
fireMeas.Operate = 1;
// Room A
AlignFunc(0);
Seek;
if(fireMeas.NonZero == 1)
{
       destroy;
       return 0;
}
else
       AlignFunc(30);
bLineStatus.Direction = cvOutput;
fireMeas.Operate = 0;
forward(68, 0, 0);
trnl(34);


// Travel to Room B
forward(67, 0, 0);
trnl(36);
forward(325, 0, 100);
trnr(67);
forward(65, 0, 0);
trnr(65);


fireMeas.Operate = 1;
bLineStatus.Direction = cvInput;
// Room B
AlignFunc(0);
Seek;
if(fireMeas.NonZero == 1)
{
       destroy;
       return 0;
}
else
       AlignFunc(107);
bLineStatus.Direction = cvOutput;
fireMeas.Operate = 0;
//forward(65, 0, 0);
trnl(34);

// Travel to Room C
forward(69, 0, 0);
trnr(70);
forward(145, 60, 0);
```

```
        trnl(40);

        fireMeas.Operate = 1;
        bLineStatus.Direction = cvInput;
        // Room C
        AlignFunc(50);
        Seek;
        if(fireMeas.NonZero == 1)
        {
                destroy;
                return 0;
        }
        else
                AlignFunc(30);
        fireMeas.Operate = 0;
        bLineStatus.Direction = cvOutput;
        forward(65, 0, 0);
        trnl(40);

        // Travel to Room D
        forward(69, 0, 100);
        trnr(72);

        fireMeas.Operate = 1;
        bLineStatus.Direction = cvInput;
        // Room D
        AlignFunc(50);
        Seek;
        if(fireMeas.NonZero == 1)
        {
                destroy;
                return 0;
        }
        else
                AlignFunc(30);
        bLineStatus.Direction = cvOutput;
        fireMeas.Operate = 0;
        forward(65, 0, 0);
        trnl(38);
        forward(150, 0, 0);
        trnr(67);

}

sub void forward(word fstop, word lstop, word rstop)
{

        word lso2;
        word rso2;
        word lso3;
        word rso3;

        word mrso;
        word mlso;

        lso2 = ALIGN / 2 - 1;
        rso2 = ALIGN / 2 - 1;
        lso3 = ALIGN / 2 - 1;
        rso3 = ALIGN / 2 - 1;

        // Initial Ping
        fso.Operate = 0;
        fso.Operate = 1;
        lso.Operate = 0;
        lso.Operate = 1;
        rso.Operate = 0;
        rso.Operate = 1;

        while((fso.received == 0) & (fso.TimeOut == 0))
        {
                // Wait
```

```
	}

	while	((fso.value > fstop) | (lso < lstop) | (rso < rstop))
	{
		// PING

		lso.Operate = 0;
		lso.Operate = 1;
		rso.Operate = 0;
		rso.Operate = 1;
		fso.Operate = 0;
		fso.Operate = 1;
		OOPic.delay = 10;

		// One Wall Calcs
		error = abs(refval - rso.value);
		prop = KP * error;

		// Display PING
		//LCD.Locate(0,0);
		//LCD.String = "O:" + Str$(lso);
		//LCD.Locate(0,9);
		//LCD.String = "F:" + Str$(fso);
		//LCD.Locate(0,17);
		//LCD.String = "O:" + Str$(rso);



		if(
		    ((rso + lso) <= ALIGN ) | ((lso >= HALL) & (rso >= HALL)) // No walls
to follow.
		  )
		{
			rse = MAXSPD / SCALE;
			lse = MAXSPD / SCALE;
		}
		// sometimes stalls if too far away from wall.
		else if( (lso >= HALL) & (rso <= HALL) ) // right wall follow
		{
			//rse = MAXSPD / SCALE;
			//lse = MAXSPD / SCALE;

			if (rso.value <= refval)
			{
				rse = MAXSPD / SCALE;
				lse = 24;
			}
			else if(rso.value > refval)
			{
				rse = 24;
				lse = MAXSPD / SCALE;
			}

		}
		else if( (rso >= HALL) & (lso <= HALL) ) // left wall follow
		{
			//rse = MAXSPD / SCALE;
			//lse = MAXSPD / SCALE;

			//if (lso.value < refval)
			//{
				rse = 25;
				lse = MAXSPD / SCALE;
			//}
			//else if(lso.value > refval)
			//{
			//	rse = MAXSPD / SCALE;
			//	lse = 25;
			//}
		}
```

```
		else  // then we're between two walls.
		{
			// median filter for three
			//if( (lso <= lso2) & (lso2 <= lso3) )
			//	mlso = lso2;
			//else if( (lso2 <= lso3) & (lso3 <= lso) )
			//	mlso = lso3;
			//else
			//	mlso = lso;

			//if( (rso <= rso2) & (rso2 <= rso3) )
			//	mrso = rso2;
			//else if( (rso2 <= rso3) & (rso3 <= rso) )
			//	mrso = rso3;
			//else
			//	mrso = rso;


			if(rso > rso2)
			{
				lse = MAXSPD / SCALE;
				//rse = (MAXSPD - (abs((lso + rso) - ALIGN))*KA) / SCALE;
// median
				rse = (MAXSPD - (abs((lso + rso) - ALIGN))*KA) / SCALE; //
no median

			}
			else if(lso > lso2)
			{
				//lse = (MAXSPD - (abs((lso + rso) - ALIGN))*KA) / SCALE;
// median
				lse = (MAXSPD - (abs((lso + rso) - ALIGN))*KA) / SCALE; //
no median
				rse = MAXSPD / SCALE;
			}
			else
			{
				rse = MAXSPD / SCALE;
				lse = MAXSPD / SCALE;
			}
		}


		//rso5 = rso4;
		//rso4 = rso3;
		rso3 = rso2;
		rso2 = rso;
		//lso5 = lso4;
		//lso4 = lso3;
		lso3 = lso2;
		lso2 = lso;

		// Display servos
		//LCD.Locate(1,0);
		//LCD.String = "L:" + Str$(lse);
		//LCD.Locate(1,17);
		//LCD.String = "R:" + Str$(rse);
	}

	lse = 0;
	rse = 0;
}


sub void trnL(word stopl)
{
	lse = -1 * TRNSPD / scale;
	rse = TRNSPD / scale;
	OOPic.Delay = stopl;
	lse = 0;
	rse = 0;
```

```
        OOPic.Delay = 50;
}


//sub void trnL(word stopl)
//{
//      OOPic.Delay = 10;
//      fso.Operate = 0;
//      fso.Operate = 1;
//      OOPic.Delay = 10;
//
//      lse = -1*(5) - 3;
//      rse = 5;
//
//      // PING right sonar
//      OOPic.Delay = 5;
//      rso.Operate = 0;
//      rso.Operate = 1;
//      OOPic.Delay = 5;
//
//      while((fso - rso) >= 4)
//      {
//              OOPic.Delay = 5;
//              rso.Operate = 0;
//              rso.Operate = 1;
//              OOPic.Delay = 5;
//      }
//}


sub void trnr(word stopr)
{
        lse = TRNSPD / scale;
        rse = -1 * TRNSPD / scale;
        OOPic.Delay = stopr;
        lse = 0;
        rse = 0;
        OOPic.Delay = 50;
}

////////////////////////
// DESTROY ALGORITHMS //
////////////////////////

// IF one of the line sensors is high, then turn off the
// corresponding servo, and continue until one on each side
// is high.  In this way, the robot will align with the tape.

// Method: Align
// Preconditions:  Robot is pointing toward entrance of room and has
// not yet passed the entrance line with more than one sensor.
// It also has a relatively straight path into the room.
// Postconditions:  Robot is fully aligned with the room entrance line.
sub void AlignFunc ( byte extraF )
{
        byte lFlag;
        byte rFlag;
        lFlag = 0;
        rFlag = 0;
        lse = ALIGNSPD + LCENTER - 3;
        rse = ALIGNSPD;

        while((lFlag == 0) | (rFlag == 0))
        {
                if(bLineStatus & 0011)
                {
                        lse = 0;
                        lFlag = 1;
                }
                if(bLineStatus & 1100)
                {
```

```
                        rse = 0;
                        rFlag = 1;
                }
        }
        OOPic.delay = 5;
        if(extraF > 0)
        {
                lse = (MAXSPD - 100) / scale;
                rse = (MAXSPD - 150) / scale;
        }
        OOPic.delay = extraF;
        rse = 0;
        lse = 0;
}


// Method: seek
// Preconditions:  Already aligned with doorway line
// Postconditions:  Flame sensed in room, or done searching and perpendicular to doorway
again.
sub void seek( void )
{
        byte lastDir;
        word fso2;

        // PING front Sonar
        OOPic.delay = 5;
        fso.Operate = 0;
        fso.Operate = 1;
        OOPic.delay = 5;
        lso.Operate = 0;
        lso.Operate = 1;
        OOPic.delay = 5;
        rso.Operate = 0;
        rso.Operate = 1;
        OOPic.Delay = 5;

        fso2 = fso;

        LCD.Locate(1,0);
        LCD.String = Str$(fso2);
        rse = ALIGNSPD;
        lse = ALIGNSPD + LCENTER;

        while (fso > (fso2 - ENTERDIST)) // This should be a constant number -
        // Move forward ENTERDIST tics.
        {
                // PING front Sonar
                fso.Operate = 0;
                fso.Operate = 1;
                OOPic.delay = 4;
        }
        OOPic.delay = 5;
        lso.Operate = 0;
        lso.Operate = 1;
        OOPic.delay = 5;
        rso.Operate = 0;
        rso.Operate = 1;
        OOPic.delay = 5;

        lse = 0;
        rse = 0;

        if ( lso > rso )
        {
                FlameL( 5 , 74 );
                lastDir = 10;
        }
        else
        {
                FlameR( 14, 81 );
```

```
                lastDir = 01;
        }

        lse = 0;
        rse = 0;
}

// Method: destroy
// Precondition:  Robot has found flame (fireMeas.NonZero == 1)
// Postcondition:  Flame has been extinguished
sub void destroy( void )
{
        Byte flameSense;
        Byte fireMeas2;
        Byte fireMeas3;
        byte vThresh;

        lastDir = 10;

        flameSense = 0;
        fireMeas2 = 0;
        fireMeas3 = 0;

        // Reset the flags
        lFlag = 0;
        rFlag = 0;
        vThresh = 0;
/*
        // Find a new maximum

        if( lastDir == 10 )
        {
                LCD.Clear;
                LCD.String = "DESTROY TURN R";
                lastDir = 01;
                rse = -1*FLAMESPD - 12;
                lse = FLAMESPD + 7;
        }
        else if (lastDir == 01)
        {
                LCD.Clear;
                LCD.String = "DESTROY TURN L";
                lastDir = 10;
                rse = FLAMESPD;
                lse = -1*FLAMESPD;
        }


        while( flameSense >= vThresh )
        {
                vThresh = flameSense;
                // Median filter
                if((fireMeas <= fireMeas2) & (fireMeas2 <= fireMeas3))
                        flameSense = fireMeas2;
                else if((fireMeas2 <= fireMeas3) & (fireMeas3 <= fireMeas))
                        flameSense = fireMeas3;
                else // last if statement unnecessary.
                flameSense = fireMeas;

                fireMeas3 = fireMeas2;  // Assignment control
                fireMeas2 = fireMeas.value;
        }
        rse = 0;
        lse = 0;
        LCD.Clear;
        LCD.Locate(1,0);
        LCD.String = "Maximum: " + Str$(vThresh);
*/
        byte i;
        // Hone in on Maximum
        //for( i = 0; i < 1; i++)
```

```
while( (lFlag == 0) | (rFlag == 0) ) // while not aligned yet.
{
        LCD.Clear;
        LCD.String = "DESTROY WHILE 1";

        // Median Filter sets these to 0 because
        // they will not trigger anything.
        fireMeas2 = 0;
        fireMeas3 = 0;

        if( (lastDir == 10) )
        {

                        LCD.Clear;
                        LCD.String = "DESTROY TURN R";
                        LCD.Locate(1,0);
                        LCD.String = "L:" + Str$(lso);
                        LCD.Locate(1,17);
                        LCD.String = "R:" + Str$(rso);
                        lastDir = 01;

                        rse = -1*FLAMESPD - 12;
                        lse = FLAMESPD + 7;

        }
        else if ( (lastDir == 01) )
        {

                        LCD.Clear;
                        LCD.String = "DESTROY TURN L";
                        LCD.Locate(1,0);
                        LCD.String = "L:" + Str$(lso);
                        LCD.Locate(1,17);
                        LCD.String = "R:" + Str$(rso);
                        lastDir = 10;
                        rse = FLAMESPD - 2;
                        lse = -1*FLAMESPD;

        }
        while( (flameSense <= FIRETHRESH) )
        {
                // Median filter
                if((fireMeas <= fireMeas2) & (fireMeas2 <= fireMeas3))
                        flameSense = fireMeas2;
                else if((fireMeas2 <= fireMeas3) & (fireMeas3 <= fireMeas))
                        flameSense = fireMeas3;
                else // last if statement unnecessary.
                        flameSense = fireMeas;

                fireMeas3 = fireMeas2;  // Assignment control
                fireMeas2 = fireMeas.value;
        }

        rse = 0;
        lse = 0;
        FlameAlign(5);
}

//AlignFunc;


// By this point, the candle has been sensed, because the fso
// sensor has reached its limit, or we've crossed the line.

// Turn on fans
Fans = cvOn;

LCD.Clear;
LCD.String = "EXTINGUISHING LOOP";
while(fireMeas.NonZero)
{
```

```
            LCD.Clear;
            LCD.String = "Extinguishing: " + Str$(fireMeas.value);

            // Wiggle
            // Servos left
            rse = FLAMESPD - 2;
            lse = -1*FLAMESPD;
            OOPic.delay = 40;
            // Servos right
            rse = -1*FLAMESPD - 12;
            lse = FLAMESPD + 7;
            OOPic.delay = 80;
            // Servos left
            rse = FLAMESPD - 2;
            lse = -1*FLAMESPD;
            OOPic.delay = 80;
            // Servos right
            rse = -1*FLAMESPD - 12;
            lse = FLAMESPD + 7;
            OOPic.delay = 80;
            // Servos left
            rse = FLAMESPD - 2;
            lse = -1*FLAMESPD;
            OOPic.delay = 80;
            // Servos right
            rse = -1*FLAMESPD - 12;
            lse = FLAMESPD + 7;
            OOPic.delay = 40;

            if(fireMeas.NonZero)
            {
                OOPic.delay = 2;
                lse = ALIGNSPD + LCENTER - 3;
                rse = ALIGNSPD;
                OOPic.delay = 25;
            }
        }
        LCD.Clear;
        LCD.String = "Extinguishing: " + Str$(fireMeas.value);
        LCD.Locate(1,0);
        LCD.String = "COMPLETE";
        lse = 0;
        rse = 0;
        Fans = cvOff;
}


sub void FlameR(byte spdr, word stopfr)
{
        lse = spdr;
        rse = -1*(spdr) - 3;

        // LCD
        LCD.Clear;
        LCD.String = Str$(lse);
        LCD.Locate(1,0);
        LCD.String = Str$(rse);

        i=0;
        while ((i < stopfr) & (fireMeas.Nonzero == 0))
        {
            OOPic.Delay = 1;
            i++;
        }
        lse = 0;
        rse = 0;
}

sub void FlameL(byte spdl, word stopfl)
{
```

```
        lse = -1*(spdl) - 3;
        rse = spdl;
        LCD.Clear;
        LCD.String = Str$(lse);
        LCD.Locate(1,0);
        LCD.String = Str$(rse);
        i=0;
        while ((i < stopfl) & (fireMeas.Nonzero == 0))
        {
                OOPic.Delay = 1;
                i++;
        }
        lse = 0;
        rse = 0;
}

sub void FlameAlign( byte timeOut)
{
        LCD.Clear;
        LCD.String = "ALIGN ROUTINE";
        // Reset the timer
        byte time;
        time = 0;

        lse = ALIGNSPD + LCENTER;
        rse = ALIGNSPD + 2;

        lFlag = 0;
        rFlag = 0;
        fso = 0;
        fso = 1;
        OOPic.delay = 5;

        while ( (time < timeOut) & (flameSense >= fireMeas) & ((lFlag == 0) | (rFlag ==
0)))
        {
                // set flags if over the circle
                if(bLineStatus & 0011)
                {
                        lse = 0;
                        lFlag = 1;
                }
                if(bLineStatus & 1100)
                {
                        rse = 0;
                        rFlag = 1;
                }
                time++;

                // Averaging Filter
                flameSense = (fireMeas + fireMeas2 + fireMeas3)/3;

                fireMeas3 = fireMeas2;
                fireMeas2 = fireMeas;

                // Ping

                fso.Operate = 0;
                fso.Operate = 1;
                OOPic.Delay = 5;
                lso.Operate = 0;
                lso.Operate = 1;
                OOPic.Delay = 5;
                rso.Operate = 0;
                rso.Operate = 1;
                OOpic.Delay = 5;

                if( (lflag == 0 ) & (rflag ==0) )
                {

                //turn right if too close to front wall
```

```
if (fso < 51)
{
        lse = 0;
        rse = 0;

        LCD.Clear;
        LCD.String = "ALIGN front trigger";

        //turn until facing away from the wall
        rse = -1*FLAMESPD - 12;
        lse = FLAMESPD + 7;

        OOPic.delay = 350;

        lse = 0;
        rse = 0;

        //move forward
        lse = ALIGNSPD + LCENTER;
        rse = ALIGNSPD + 2;

        OOPic.delay = 20;

        lFlag = 0;
        rFlag = 0;

        // set flags if over the circle
        if(bLineStatus & 0011)
        {
                lse = 0;
                lFlag = 1;
        }
        if(bLineStatus & 1100)
        {
                rse = 0;
                rFlag = 1;
        }



}
//turn right if too close to left wall
if (lso < 43)
{

        lse = 0;
        rse = 0;

        LCD.Clear;
        LCD.String = "ALIGN left trigger";

        //turn right
        rse = -1*FLAMESPD - 12;
        lse = FLAMESPD + 7;

        OOPic.delay = 200;

        lse = 0;
        rse = 0;

        lFlag = 0;
        rFlag = 0;

        //move forward
        lse = ALIGNSPD + LCENTER;
        rse = ALIGNSPD  + 2;

        OOPic.delay = 20;

        // set flags if over the circle
```

```
            if(bLineStatus & 0011)
            {
                    lse = 0;
                    lFlag = 1;
            }
            if(bLineStatus & 1100)
            {
                    rse = 0;
                    rFlag = 1;
            }


            //move forward
            //lse = ALIGNSPD + LCENTER;
            //rse = ALIGNSPD;

            //OOPic.delay = 15;



    }
    //turn left if too close to right wall
    if (rso < 43)
    {
            lse = 0;
            rse = 0;

            LCD.Clear;
            LCD.String = "ALIGN Right trigger";

            //turn left
            rse = FLAMESPD + 1;
            lse = -1*FLAMESPD;

            OOPic.delay = 200;

            lse = 0;
            rse = 0;

            //move forward
            lse = ALIGNSPD + LCENTER;
            rse = ALIGNSPD + 2;

            OOPic.delay = 20;

            lFlag = 0;
            rFlag = 0;

            // set flags if over the circle
            if(bLineStatus & 0011)
            {
                    lse = 0;
                    lFlag = 1;
            }
            if(bLineStatus & 1100)
            {
                    rse = 0;
                    rFlag = 1;
            }


            //move forward
            //lse = ALIGNSPD + LCENTER;
            //rse = ALIGNSPD + 2;

            //OOPic.delay = 20;


    }
    }
}
```

```
        rse = 0;
        lse = 0;
}
```

Appendix K

Responsibility Division

Task	**Afandi**	**Austin**	**Behera**	**Porterfield**
Wheel and Chassis Design			X	
Motor Research/Design			X	
Sensor Research		X		
Microprocessor Research	X		X	X
Microprocessor/Motor Integration	X			X
Motor/Chassis Integration			X	
Motor Open Loop Control	X			
Motor Closed Loop Design	X			X
Motor Closed Loop Implementation	X			X
Sensor Integration Firmware				X
Locomotion Firmware	X			X
Reckoning Firmware	X			X
Extinguishing Firmware		X		X
Floor Sensor Design/Build		X		
Choose/Order Distance & Flame Sensors		X		
Extinguisher Design			X	
Extinguisher Implementation		X	X	
Microphone Starter Design		X		
Microphone Implementation		X		
Power Supply		X	X	
Overall Control Changes	X	X		X
Debug	X	X	X	X

Appendix L

Budget

Number	Part Description	Price	Quantity	Amount
1	Futaba S3010 servos	$27	2	$54.00
2	Wheels and Tires	$3.75	4	$15.00
3	Casters	$5	2	$10.00
4	OOPIC	$65	1	$65.00
5	Replacement Gear	$4	1	$4.00
6	Devantech SRF04 sonar sensors	$35	3	$105.00
7	Hamamatsu UVTron	$65	1	$65.00
8	Line sensors	$4	2	$8.00
9	Perf board	$3	3	$9.00
10	9V Batteries	$3.30	2	$6.60
11	Push Buttons	$3	2	$6.00
12	Buzzer	$3.50	1	$3.50
13	Battery Cases	$2	4	$8.00
14	Miscellaneous			$10.00
15	Standoffs	$.60	10	$6.00
16	AA Batteries	$3	22	On Loan
17	Velcro	$4.00	1	$4.00
TOTAL				$379.10

The batteries are on loan from Jaime.

Appendix M

Progress Reports

Group 9

Firefighting Robot

Labor Division

Current Progress

- Firmware
  - Research and Basic Testing Phase (Testbenches)
- Controls
  - Overall Logic Design (How many wheels, system definition)
- Sensors
  - Research (i.e.) Sonar vs. Infrared Distance, Pre-built Flame detector vs. Self-Designed
- Mechanical
  - Motor Research (Servo vs. Stepper)
  - Chassis Design (Shape and

Current Goals

- Testbench completion for testing of hardware (Firmware) and basic algorithm design
- Completion of Controller board and motor interfacing design (Mechanical)
- Calibration and placement of sensors (Sensors)
- Specific Mechanical Algorithm Design (Controls)

Gantt Chart

- We have created two Gantt Charts
  - Dallas Gantt (Fast track with Regional Competition)
  - Hartford Gantt (Contingency and National Competition)
- In the interest of time, only one will be presented.

Questions

- Feel free to ask questions as time permits.

Firefighting Robot

Group 9

John Porterfield, Asim Behera, Shoaib Afandi, Jaime Austin

Labor Division

Current Progress

- Firmware
  - Testing of motors, flame and wall sensors with microcontroller
- Controls
  - Testing of directional control of servos
  - Design of closed loop control
- Sensors
  - Tested operation of flame and wall sensors
  - Design of sound and floor sensors
- Mechanical
  - Chassis is built
  - Servos are “hacked” and mounted

Current Goals

- Firmware
  - PID control of robot
  - Obstacle detection and avoidance
- Controls
  - Sensor integration and calibration
  - Navigation control of robot
- Sensors
  - Build and test line and sound sensors
- Mechanical
  - Power system and mounting sensors

Gantt Chart

- We are currently on schedule to compete in the national competition
- We have decided not to compete in the regional competition

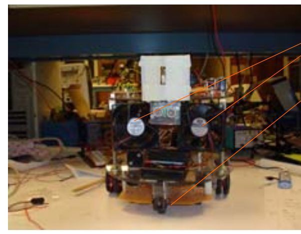
Final Status Report

Team 9

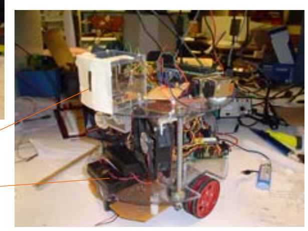
Shoaib Afandi
Jamie Austin
Asim Behera
John Porterfield

Our ROBOT

Front View

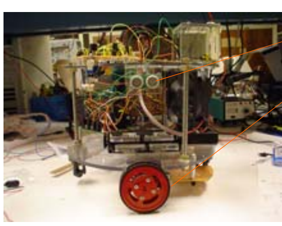


Profile

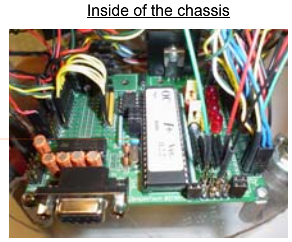


Our ROBOT

Side View



Inside of the chassis



Firmware and Integration (John Porterfield)

- TASK ACCOMPLISHED
  - → Extinguishing algorithm is complete and is now in debug stage.
  - → Sweeping of the maze algorithm is complete.
- TASK LEFT
  - → Currently working with Shoaib on the wall avoidance algorithm.

Controls
(Shoaib Afandi)

- TASK ACCOMPLISHED

  → Control algorithm is complete, needs little fine tuning.

- TASK LEFT

  → Currently working on the wall avoidance algorithm.

Sensors
(Jamie Austin)

- TASK ACCOMPLISHED

  → All of the sensors namely: Sonar, Line and the Flame detector are integrated.

- TASK LEFT

  → Debugging the fire extinguishing algorithm.

Mechanical
(Asim Behera)

- TASK ACCOMPLISHED

  → Robot is fully built, all the sensors and power supplies are mounted.

- TASK LEFT

  → Researching on a better blower circuit and then on its installation.

Gantt Chart

Appendix N

Minutes of Meetings

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT Purpose of Meeting: WEEKLY UPDATE Date/Time:03/21/04 Chair: Dr. Guoliang Fan			
Topic	Discussion	Action	Person Responsible
1.	Showed progress on current robot for navigating the maze.	N/A.	John Shoaib
2.	Decided to meet next week to discuss progress for competition.	N/A	Jamie Shoaib John Asim

Minutes of Last Week's Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT
Purpose of Meeting: WEEKLY UPDATE
Date/Time:02/10/04
Chair: Dr. Guoliang Fan

Topic	Discussion	Action	Person Responsible
1.	Our report should include in detail what we have done and where we lagged behind. Secondly the various parts should state the name of the persons responsible for them. A revised Gantt chart should be also included.	The directions were incorporated immediately.	The whole team on their individual parts.
2.	Chassis to be completed by 02/17/02. Test benches for different sensors to be written by the above date.	Completed the tasks.	Asim John
3.	Prototyping sensors	In progress	Shoaib Jamie

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT
Purpose of Meeting: WEEKLY UPDATE
Date/Time:02/24/04
Chair: Dr. Guoliang Fan

Topic	Discussion	Action	Person Responsible
1.	Chassis complete, servos mounted with wheels.	Accomplished	Asim
2.	Configured the LCD to work with the OOPIC	Accomplished	John.
3.	Microphone circuit built	Accomplished	Jaime

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT
Purpose of Meeting: WEEKLY UPDATE
Date/Time:03/2/04
Chair: Dr. Guoliang Fan

Topic	Discussion	Action	Person Responsible
1.	Most importantly we decided not to go for the Dallas competition…due to lack of time.	N/A.	Unanimous decision.
2.	All sensors are fully working. The sonar sensor outputs the value to the LCD. Photo sensor yet to be built.	Completed the task.	Jamie Shoaib John
3.	Codes written and implemented. Sonar sensors to be mounted on the robot.	In progress	John Shoaib Asim

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT Purpose of Meeting: WEEKLY UPDATE Date/Time:02/3/04 Chair: Dr. Guoliang Fan			
Topic	Discussion	Action	Person Responsible
1.	Our proposal is graded. We have ordered parts and in the meantime writing test benches.	N/A	The whole team.
2.	Each one of us should write our progress in a word document, no longer than 2 pages and turn it in to chair. This should include our weekly tasks and what we expect to complete next week.	Directions were followed immediately.	The whole team.
3.	Procure an old robot from Dr. Teague.	Accomplished	John

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT
Purpose of Meeting: WEEKLY UPDATE
Date/Time:02/17/04
Chair: Dr. Guoliang Fan

Topic	Discussion	Action	Person Responsible
1.	Chassis complete, waiting on the arrival of wheels.	Accomplished	Asim
2.	OOPIC board in the final stage with all of its components soldered.	Accomplished	The whole team.
3.	Chassis should be running by next meeting, flame sensor should be calibrated, line sensor should be made.	Accomplished	Asim Jaime

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT
Purpose of Meeting: WEEKLY UPDATE
Date/Time:03/9/04
Chair: Dr. Guoliang Fan

Topic	Discussion	Action	Person Responsible
1.	Chassis is complete, with everything mounted on it.	Accomplished	Asim
2.	Line detector circuit built. Sonar sensors integrated with the servos. Power supply for the OOPIC and servos upgraded.	Accomplished	The whole team.
3.	Decided not to work over spring break.	N/A	The whole team.

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT Purpose of Meeting: WEEKLY UPDATE Date/Time:03/30/04 Chair: Dr. Guoliang Fan			
Topic	Discussion	Action	Person Responsible
1.	Team further subdivided into two teams.	N/A	N/A
2.	Fans to be mounted on the robot. Chair suggested we use median filter for better control.	Accomplished	Asim Shoaib.
3.	Flame sensor had noise problems.	Fixed	John Jaime

Minutes of the Meeting

Name of Organization: Team 9-FIRE PROTECTION ROBOT Purpose of Meeting: WEEKLY UPDATE Date/Time:04/9/04 Chair: Dr. Guoliang Fan			
Topic	Discussion	Action	Person Responsible
1.	Robot is complete in every sense.	N/A	Asim
2.	Extinguishing algorithm works. Maps the floor plan with little control issues.	Still working on it.	John Shoaib
3.	Floor plan to be repainted and fixed so that it resembles the actual arena.	Accomplished	Asim Jaime