Fire-Fighting Robot

By Merima Jahic and Christopher Elpers
Table of Contents

I. Abstract..................................................................................................................III
II. Acknowledgment.....................................................................................................III
1. Introduction.............................................................................................................1
2. Project Requirements............................................................................................1
3. Software and Hardware Design..............................................................................2
   3.1 Audible Sensor................................................................................................2
   3.2 Light Sensors and Detectors............................................................................4
   3.3 Distance Sensor..............................................................................................6
   3.4 Encoder.........................................................................................................7
   3.5 The Chassis of the robot................................................................................8
   3.6 Software.......................................................................................................9
4. Results...................................................................................................................12
5. Conclusion...........................................................................................................13
6. Future work.........................................................................................................14
7. References...........................................................................................................14
8. Cost....................................................................................................................15

Appendices

A Official Rules for the IEEE SOUTHEASTCON 2003 Hardware Competition
B Source Code
C Lego board Version 6 Schematic
I. Abstract

In this paper we will try to describe the process of designing a fire-fighting robot, by the standards of the IEEE SouthEastCon 2003 Student Hardware competition requirements. The robot has to be autonomous. It will ultimately wait for the 1500 Hz signal to start the movement, navigate through the house-like maze, detect the LED “candle”, cover it with a plastic cup and go back to the starting position. The design will include the circuits for the audible sensors, wall detector, encoder and light sensors and detectors. Phillips microcontroller incorporated in the Lego board version 6 designed by the University of Evansville faculty is used to provide the main control for the robot and separate hardware previously mentioned.

II. Acknowledgment

We would like to thank Dr. Dick Blandford, Dr. Anthony Richardson, and Mr. David Mitchell for their support and help with the design. Also, our thanks go to The University of Evansville College of Engineering and Computer Science for providing funding for this project.
1. Introduction

In this paper we will describe the process of designing a fire-fighting robot, by the standards of the IEEE SOUTHEASTCON 2003 Student Hardware competition requirements. The robot must be autonomous. It will wait for an audible signal with a frequency of 1500 Hz to begin movement, navigate through a house-like maze, detect an LED “candle”, cover it with a plastic cup and return back to the starting position. The design will include the circuits for the audible sensor, distance sensor, and light sensors. An 89C51RD2 Phillips microcontroller incorporated in the Lego board version 6 designed by the University of Evansville faculty was used to provide the main control for the robot and separate hardware previously mentioned.¹

2. Project Requirements

The requirements for this project are based on the IEEE SOUTHEASTCON 2003 hardware competition.² The restrictions are set as follows:

- The Robot must be autonomous.
- It will start its movement after it detects the audio signal of 1500 Hz.
- The walls are not to be touched by any part of the robot during the motion.
- The robots dimensions must be no greater than 21 cm x 21 cm x 20 cm (W x L).
- The fire is considered extinguished when the LED “candle” is completely covered by the cup resting flush on the playing surface.
- The lighting conditions of the arena will not be specific or fixed.

To achieve a working and optimal design these conditions were considered, making the design procedure more difficult. (See Figures 2.1, and 2.2 for the course specifications and the LED “candle” respectively.)

¹ For more information on Lego board version 6 schematics go to: csserver.evansville.edu/~lego101
² For more information on IEEE SOUTHEASTCON 2003 hardware competition rules go to: http://www.ewh.ieee.org/r3/jamaica/southeastcon/robot.html
3. Software and Hardware design

The design procedure can be broken down into several steps:

- Audible sensor design to detect the start signal
- Light sensor design to detect the LED “candle” in the room
- Distance sensor to determine the distance from the front of the robot to the “candle”
- Encoder design to determine the distance the robot traveled
- Software with functions to control hardware developed
- The chassis design of the robot

The performance of each piece of hardware can be tested separately which speeds up the process of the design, hence the software has been broken down into separate functions to allow the designers to control the separate pieces of hardware.

3.1 Audible sensor

The audible sensor has been designed using an analog filter and an amplifier. In order to trigger the robot’s movement at the desired frequency, other possible noises at different frequencies had to be filtered out. To filter a wide range of frequencies that are not desirable, a simple amplifier with the gain of 100 has been used. This op-amp acts like a low-pass filter with a cut-off frequency of 300
Hz because of the real time performance of the LM741 operational amplifier (used in the implementation of the circuit), since $f_{3db}=\frac{1.5MHz}{A_v}$, where $A_v$ is the gain of the amplifier. In this case the gain will be one hundred. Output from this op-amp is taken through a simple RC high-pass filter, with the desired frequency just greater than 1500 Hz. Using the rule of thumb which says that a capacitor must be 10 times greater than the capacitor which gives the exact desired frequency if used in the high pass filter: $C > \frac{10}{(2\pi f_c)}$ where $f_c=1500\text{Hz}$, we chose values for $R$ and $C$ that satisfy the given equation. A gain of a 100 is used to amplify the small signal coming from the receiving microphone (as mentioned before). To achieve this gain a non-inverting amplifier is used as suggested previously, which also plays the role of the low pass filter. The values of the resistors for the amplifier are determined using the following equation: $A_v = \frac{(R_1 + R_2)}{R_2}$. Figure 3.1.1 illustrates the design and figure 3.1.2 illustrates the simulated performance of the filter.

---

3 See LM741 data sheet at cecsfp/resources/schematics/
3.2 Light Sensors and Detectors

Phototransistors were used to find the light of the candle and follow the white line that runs throughout the course. These sensors are connected to the analog-to-digital converter on the Lego board, returning values ranging from 0 to 255 (corresponding to 0 to 5 Volts).

The sensors that are used to follow the line are mounted with an LED parallel to them facing the same direction. This gives the sensor a better range of values by increasing the ambient light that the white tape reflects. Three of these sensors are oriented in a straight line (see figure 3.2.1).
This set-up allows the robot’s orientation with regard to the white line to be known at all times. If the middle sensor is the only one reading the white line, then the line is in the middle of the robot, and if the right sensor reads the line, then the line is to the right, etc. Using just two sensors would be inadequate; the position of the white line would be undetectable when it’s located in between the two sensors.

The use of more sensors would have been beneficial; however, the analog-to-digital converter on the Lego v6 board only has eight channels, limiting the total number of sensors that can be used.

The phototransistors located on the front of the robot are used to guide the robot to the candle once it is in the proper room. These sensors are built using the same phototransistors previously discussed. These sensors do not require any additional light as the candle emits light. Three sensors were used, one in the center of the robot, and one on each side of the robot facing forward. These sensors can be seen in the following picture, figure 3.2.2.
With such a configuration, the robot is able to easily determine the position of the candle and move into position to deploy the candle. The position of each sensor is important as well, the middle sensor is at a height to detect the light from the lower LEDs on the candle and the adjacent two are at a height to detect the higher LEDs on the candle. We found the sensors are able to ‘see’ the light of the candle at a distance of up to 3 feet. With the sensors organized in this manner, the robot can easily make adjustments to its trajectory and guide itself directly at the candle while moving towards it.

Again, the use of more sensors on the front of the robot would have been beneficial, but do to the limitations of the Lego board used, was unfeasible. Three sensors were used because they read the light much better when it shines directly into them. If, for instance, only 2 sensors had been used, it would be rather difficult to determine the exact bearing of the candle because of its circular form.

3.3 Distance sensor

The distance sensor was built using an infrared light emitting diode and phototransistor. The infrared signal emitted from the diode bounces off of any solid surface and the phototransistor detects the strength of the signal. Depending on the intensity of the signal we can determine how far the object is from the front of the robot. The analog-to-digital converter on the Lego board makes it easy to check the values since the intensity of the detected signal is nicely converted to numerical values ranging from 0 to 255 (corresponding to 0 to 5 Volts). The problem with using this approach to determine the distance is interference from the surrounding lighting. To avoid this interference, we use the fact that the

---

4 For more information on the infrared emitter and detector see www.radioshack.com for data sheet.
ambient light is a modulated light. Therefore, we created an oscillator that oscillates at a low frequency. This forces the diode to emit the infrared light at that frequency. On the receiving end, the signal from the phototransistor is sent through filters that only allow a signal at the frequency of the emitter pass through. For the design of the infrared emitting diode with the oscillator and phototransistor with filters see figure 3.3.1 and 3.3.2.

![Infrared emitting diode with the oscillator](image)

**Figure 3.3.1 Infrared emitting diode with the oscillator**

![Phototransistor with filters](image)

**Figure 3.3.2 Phototransistor with filters**

### 3.4 Encoder

The encoder consists of an infrared emitter and phototransistor. They are aimed at each other. The emitter is a simple diode that emits infrared light and the phototransistor detects the infrared light from the emitter unless there is interference from a solid object between the two. The circuit is illustrated in the figure 3.3.1. This principle is used to determine how far the robot has traveled. The gear attached to the motor is placed between the emitter and the phototransistor. When the robot is in motion, the teeth of the gear cross between the emitter and the transistor causing a hardware interrupt. It will be recognized as a PCA (Programmable Counter Array) interrupt since the encoder is connected to PCA interrupt pin. Counting the interrupts (each tooth of the gear) allows us to calculate the distance the robot has moved. The emitter used for the project is the
Omron EX-SX1088 LED and phototransistor packaged together.\(^5\) (See figure 3.4.1 and figure 3.4.2)

\[\text{Figure 3.4.1 (encoder circuit)}\]

3.5. Chassis

The chassis of the robot is constructed primarily of Legos. This made it easy to prototype design ideas and to make any necessary design alterations throughout the progress of the project. The specifications for the competition allow the robot’s dimensions to be at a maximum of 21 x 21 x 20 cm (width, length, and

\(^5\) For more information on the encoder see rocky.digikey.com/WebLib/Omron/Web Data/eesx1035_1081.pdf
A robot of this size would have a great deal of trouble navigating through the course as the hallways are only 23 cm wide, leaving only 1 cm on each side of the robot in the hallway. With this in mind, we wanted to keep the chassis of the robot as small as possible to increase maneuverability. We chose to use two wheels as the method of steering for its simplicity and also to conserve space. A roller ball was used as a third wheel of sorts; it balances the robot and gives the front end a ride height that is congruent with the height of the LEDs on the candle. We also made sure that we have enough space to put the Lego, audible and distance sensor circuit boards on top of the robot chassis.

3.6 Software

The controller board was programmed in C using Keil uVision 2. The software was designed with modularity in mind. Each task the robot needs to accomplish is done through a series of function calls. Many functions are used multiple times in one run to cover the candle. The flowchart illustrates the code used for this project. (See figure 3.6.1)
Figure 3.6.1 Software flowchart
The following is a list of the functions and a brief description of each. (Complete listing of source code is available in appendix B.)

*unsigned char ADC(unsigned char chnum)*

This function returns the value of a channel number passed to it from the analog-to-digital converter. Because the Lego board uses an 8-bit analog-to-digital converter, the values range from 0 to 255, and an unsigned char is an appropriate data type for the return value.

*void initializeCounter(void)*

*initializeCounter* simply sets up the timers and the pulse width modulation (PWM) for the interrupt of the encoder. It enables the programmable counter array module 3 to be used for detecting a hardware interrupt caused by some external event, in this case the encoder.

*void cupPlacement(void)*

To use the servo motor to place the cup over the candle, the timers used for the operation have to be set up, this occurs initially in this function. The servo motor is then moved into position over the candle, and is returned to its original position. *getBackHomeFromRoom1 (or 2, 3, 4 depending on the room the candle is located)* is then called to return the robot to the home area.

*void Delay(void)*

This is a simple delay function consisting of a nested *for* loop that is used for timing in the positioning of the servomotor.

*void start(void)*

*start* loops idly until the filter acknowledges the proper starting signal, detected on port 1.3. This function is called immediately after *initializeCounter* in the main code.

*void lightFind(void)*

This function guides the robot to the candle while periodically checking the distance sensor to ensure the candle is not too close. Positioning the robot the proper distance directly in front of the candle is accomplished by moving forward any time the center light sensor *sees* the light from the candle. If the light is sensed by either the right or the left sensor exclusively, the trajectory of the robot is altered to the appropriate direction. If no light is detected by any of the sensors, the robot will scan in alternating directions, at increasing lengths. For example, the first time the light from the candle is undetected, the robot will turn approximately 10 degrees to the left. If the light is not found during that scan, or the light is again lost, the car will turn approximately 20 degrees in the opposite direction. This pattern keeps the robot from getting forced to make a complete 360 degree turn simply because the candle is just 10 degrees in the opposite direction, if it isn’t found in one short turn, it will be found during the next turn in the opposite direction.
The entire time the robot is moving toward the candle, it looks at the distance sensor. If this value increases to a predetermined intensity for the candle to be in front of the car at the proper distance, and at least one of the light sensors sees a light, the robot is stopped and cupPlacement is called.

```c
void room1(void), void room2(void), void room3(void), void room4(void)
These functions are designed to use the line sensors and the encoder to get the robot to the doorway of the proper room. Each room was given a number and each function corresponding to that number puts the robot at the doorway to that room. These functions are called only once per run by the main code and are almost identical with the exception of different distance values to be traveled from the home area to the desired room. Once this function is called, it simply looks at the state of the counter, incremented by the PCA interrupt caused by the wheel rotation, and the lower light sensors facing the floor of the course. In each possible state the motors can be in within this function, a direction bit is set. This global variable is then considered each time the encoder causes an interrupt. If the direction bit is set to 1, meaning the motors are both set to forward, the interrupt routine increments the counter called straightCounter. If the direction bit is set to 0, the interrupt knows that both motors are not set to forward and straightCounter will not be incremented (see figure 3.6.2). This increases the accuracy of the distance calculations greatly. Generally speaking, when both of the motors are not set to move forward very little, if any, forward progress is achieved, so the variable used to gauge distance will not be altered in the interrupt. In summary, this function follows the white line on the floor of the course for a specified distance, then returns.

```c
PulseTime() interrupt 6 using 1
{
  if(CCF3 == 1)
  {
    counter++; // This counter is incremented regardless of the direction the motors are set to
    if(direction == 1) // If both motors are moving forward
      counterStraight++; // Increment the counter for straight movement
    TF2 = 0; // This resets the timer 2 overflow bit
    // This is the 'watchdog' timer
    // Each time this interrupt occurs it keeps the timer counting. It is important to do this here because we know the wheels are turning
    // so as long as the wheels turn, the watchdog timer interrupt will not occur
    CCF3=0; // reset the module 3 interrupt flag
  }
}
```

Figure 3.6.2 PCA interrupt

Fire-fighting Robot Merima Jahic and Christopher Elpers
void getHomeFromRoom1(void), void getHomeFromRoom2(void), void getHomeFromRoom3(void), void getHomeFromRoom4(void)

These functions maneuver the robot from the position it deploys the cup back to the home area using the line sensors and the encoder. Each is called only once by cupPlacement. Each getHomeFromRoom function is fundamentally identical to its corresponding room function. The only difference is that at the beginning of this function, the car is still located inside the room the candle is in. Subsequently, this function backs the robot up until the white line in the doorway is detected. After this, the robot continues to move in reverse a specified distance to get itself into a better position to make a 90 degree turn to orient itself parallel with the white line in the hallway. It then makes the aforementioned 90 degree turn (either right or left depending on the room it traveled to) and commences following the line in the manner the functions room1, room2, room3, and room4 do. Finally, once this distance is achieved, the motors are turned off and the code enters a forever loop (this is because there is no no-op instruction on the 89c51RD2) and no further instructions execute. One important fact we would like to mention is that the official rules for the IEEE hardware competition state that the robot is considered to be in the home area as long as some part of it has crossed over the tape enclosing the home area. As a result, there is a large margin for error on the trip back to the home area, the robot can stop short or go past the home area as long as some portion is inside the tape enclosing the home area.

**Watchdog Timer**

If the encoder interrupt does not occur in approximately 13 seconds, the watchdog timer backs the robot up slightly and makes a right or left turn. The direction of this turns alternates and gets increasingly larger each time the interrupt occurs. This keeps the robot from getting trapped against the wall for more than about 13 seconds and gets it into a better position complete the run.

void SetMotor(unsigned char, int) and void AllStop(void)

These two functions are used to control the speed and the rotation direction of the two DC motors. SetMotor sets up the Pulse Width Modulation in the PCA. It initializes the PCA and by accepting the unsigned char value between 0 and 255 it sets up the length of the time interval of the pulse sent to the DC motors. This determines the speed of the motor. AllStop simply sends zeros to both of the motors, which does not send any pulses to the motors. This stops their rotation.

4. Results

The robot performs well in the course. It follows the white line accurately and it never gets off track or comes in contact with the wall. The distance sensor is better at detecting the distance from the front of the car to the LED “candle”, than the distance from the walls. The walls are painted flat black which does not reflect the infrared light as well as the candle. Originally we had planned on using the distance sensor to avoid coming into
contact with any walls. However, the fact that the distance sensors does a poor job sensing the black walls made it difficult for it to be useful, but the encoder determines the distance the robot travels precisely, allowing the robot to know exactly when to turn and avoid the walls. Detecting the light from the candle is another task which the robot performs well. The room in which the candle resides must be known before hand so it can be programmed with the proper code.

5. Conclusion

Working on this project helped us develop many important engineering skills, such as: troubleshooting, low level programming, hardware-software interaction, circuit design, and control design using microcontrollers. This experience will prove beneficial for our future design work.

6. Future work

To improve the design of the fire-fighting robot, more sensors could be added improving its ability to find the candle. Additionally, light sensors could be added to help distinguish the white line. The spacing of only three sensors can lead to some confusion as the edges of the white line do in fact reflect light better than the black floor. Consequently, when a sensor is over the black floor just beyond the edge of the white line, the sensors detect more light than they would if they were over the black floor far from the line. If more sensors could have been used here, a much clearer view of the orientation with respect to the white line could be known. An additional analog-to-digital converter could be a possible method of solving this problem using the Lego v6 board.

7. References

1. Infrared Emitter and Detector Pair Specifications
   Available at http://support.radioshack.com/support_supplies/doc31/31145.htm

2. OMRON EE-SX 1081 Specifications
   Available at http://rocky.digikey.com/WebLib/Omron/Web%20Data/eesx1035_1081.pdf

3. SOUTHEASTCON 2003 Hardware Competition Rules
   Available at http://www.ewh. ieee.org/r3/jamaica/southeastcon/robot.html

4. University of Evansville Department of Electrical Engineering and Computer Science, Datasheet Archive
   Available at http://cecsfp/resources/schematics/

5. University of Evansville Lego 101 Project Home Page
   Available at http://csserver.evansville.edu/~lego101
8. Cost

<table>
<thead>
<tr>
<th>Item</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools</td>
<td>$60</td>
</tr>
<tr>
<td>Lego Board</td>
<td>$100</td>
</tr>
<tr>
<td>Lego kit</td>
<td>$200</td>
</tr>
<tr>
<td>Parts (encoder, sensors, emitters, etc.)</td>
<td>$50</td>
</tr>
<tr>
<td>Supplies (batteries, glue, etc.)</td>
<td>$40</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$450</strong></td>
</tr>
</tbody>
</table>