Firefighter Bot Proposal

Antonio Santos
Guhan Iyer

TEJ4MI-01

2022-09-26

Introduction

The purpose of this project is to create a robot capable of navigating through several rooms in a maze to locate and extinguish a candle lit on fire. During its run in a maze, the bot also has to be able to avoid several obstacles, such as pits and walls. Success in this project relies on effective use of the bots' several subsystems, such as line, wall and flame detection. Additionally, good code, circuit and bot design, as well as overall creative thinking are qualities that are needed to complete this project successfully.

Description

Challenge Statement

When designing our bot we posed some questions to ourselves. The main question we came back to was: “can we find a flame before we enter a room?” With our design, we have tried to make it possible to do so. Physically, we will utilize a pop-up flame detection sensor, and the ability of the GM8 motors to turn in place. This will be combined with programming that will allow our bot to stop outside each room, scan for an infrared source from the candle, and make an appropriate decision on whether or not to enter the room. Our bot, component and material wise, does not stray far from the typical design of past firefighter bots. We will be using stock GM8 motors and wheels, and normal materials like wood. The uniqueness of our bot comes from our sensor placement and clever programming.

Line Detection

Between the sumo bot and firefighter bot, the line detection system functions identically. A super bright LED positioned towards the ground flashes, and its light is either absorbed by the black ground or reflected by white surfaces. This is interpreted by a phototransistor positioned alongside the LED as either a high voltage “1” or a low voltage “0” respectively. However, between the sumo bot and firefighter bot, the purpose of the line detection system has changed. While in the sumo bot the line detection system was used to stop the sumo bot from exiting the ring, the firefighter bot uses the line detection system to indicate whether it has entered a room or not. In the two mazes, the entrances of most rooms are denoted by a painted white stripe on the floor. Once the bot crosses over this line, a coded-in counter is added to by one. Our bot will have subroutines specific to each room that will take effect according to the current line detection counter. The use of the line detection system is not only to indicate whether or not the bot has entered a room, but also to activate other subroutines necessary for each room. For example, once the bot has entered a room, the counter would increase by one. As such, subroutines involving flame searching and extinguishing can then be activated as necessary.

Wall Detection

The wall detection system of the firefighter bot is used to detect walls, which pose an obstacle to the bot. It utilizes two wall sensors, one positioned at the front of the bot and one on the left side of the bot. There are a variety of uses for the wall detection system. An obvious use is to purpose the wall detection system to avoid walls so the bot does not run into them, but another use is to maneuver the bot inside rooms in order to locate the flame better. This use of the wall detection system would be quite efficient when used in combination with the flame detection system.

Flame Detection and Extinguishing

Flame detection and extinguishing are two of the foremost subsystems of the firefighter bot. They are the systems that accomplish the primary goal of the firefighter bot: to locate and extinguish a fire. The flame detection system itself is a repurposed version of the bot detection system from the sumo bot project, as it functions off of similar logic. In the bot detection system of the sumo bot, an infrared emitting LED consistently blinks until it reflects off of another bot and gets received by the Vishay 4038 sensor. Following this, the bot moves appropriately. With the firefighter bot, the goal is now to detect a source of infrared light, being the candle lit on fire. To do so, the bot utilizes the QSD123 phototransistor to specifically detect the fire's emitted infrared light. Based on the reading of the QSD123 phototransistor, either a low voltage "0" or a high voltage "1", the bot will move towards the infrared source, and the RB-SPA-718 fan will turn on as necessary to extinguish the flame. As previously mentioned, we aim to have our flame sensor on a mechanism that will allow it to pop up above the walls of a room, scan for an infrared source, and make an appropriate decision regarding whether or not to enter the room.

Motors and Wheels

The firefighter utilizes the GM8 motors, the wheels, and brackets that come with it, which are nearly identical to the GM2 motors, wheels, and brackets that were used in the sumo bot. They have identical dimensions and weight, but the GM8 motors provide nearly double the speed of the GM2 motors. For our bot, we will be using stock versions of the GM8 motors and wheels, and only using two motors. We will be heavily utilizing the ability of the motors and wheels to turn in place, in order to maximize our speed and efficiency when it comes to going through each room in the mazes.

Design and Selected Materials

The design of our bot is intended to be minimal and sleek. Our primary concerns when designing the bot were efficiency and speed. To do so, our bot is quite small, with dimensions of

15 x 15 x 21 cm. Our design does not deviate far from the typical design of past firefighter bots. We have three levels: the base level where the motor board is located, the second level where the main board is located, and the top level where the sensor board is located. Our bot is entirely constructed out of wood. This includes the supports between levels, meaning we have no bolts or washers. Due to the wide availability of wood and its lightweight, we aim to use it as much as possible. The bolts that would be needed to support all three levels are heavy, dragging down the speed of our bot, and would perhaps require majorly changing our design in order to account for the height of the bolts. To avoid the difficulties that bolts would bring to our bot, we will be using wooden supports that will be nailed into the levels.

Material List

Part	Specification	Quantity	Source	Unit Cost	Total Cost
Battery	-	-	-	-	-
Resistors					
	220 Ω	1	Abra	$0.03	$0.03
	1K Ω	1	Abra	$0.07	$0.07
	4.7K Ω	1	Abra	$0.37	$0.37
	10K Ω	2	Abra	$0.07	$0.14
	10K Ω Variable	1	Abra	$1.92	$1.92
	47K Ω	1	Abra	$0.05	$0.05
	150K Ω	1	Abra	$0.03	$0.03
Capacitors					
	0.1 µF	8	Abra	$0.09	$0.72
	1000 µF	3	Abra	$0.38	$1.14
Chips and Microcontrollers					
	PIC16F887	1	Abra	$5.53	$5.53
	L293D	1	Abra	$3.85	$3.85
	16 Pin Mechanical Socket	1	Abra	$0.65	$0.65
	40 Pin Mechanical Socket	1	Digi-Key	$0.78	$0.78
Wall Detection					
	GP2Y0A02YK0F Sensors	2	Abra	$13.85	$27.70
Line Detection					
	Phototransistor	1	Abra	$2.74	$2.74

	Superbright LED	1	Abra	$0.43	$0.43
Motors and Wheels					
	GM8 Motor and Wheels	2	Solarbotics	$10.21	$20.42
	GM8 Brackets	2	Abra	$2.65	$5.30
Miscellaneous					
	TIP120 Transistor	1	Abra	$0.57	$0.57
	LCD-MOD-11 Display	1	Abra	$8.95	$8.95
	1N4148 Diode	1	Abra	$0.03	$0.03
	RB-SPA-718 Fan	1	RobotShop	$10.26	$10.26
	QSD123 Phototransistor	1	Abra	$0.65	$0.65
	ON/OFF Switch	1	Abra	$0.45	$0.45
	Terminal Blocks	5	Abra	$3.46	$17.30
	Wood	-	-	-	-
	Screws	-	-	-	-
	Tape	-	-	-	-
	Bolts	-	-	-	-
	Wire	-	-	-	-
Total Cost					$110.08

Timeline

September (5 Working Days)				
Monday	Tuesday	Wednesday	Thursday	Friday
26	27	28	29	30
DEADLINE: Proposal Flash LED: Gathering materials and begin breadboarding	Flash LED: Test if breadboard is working. If not, get a new one	Flash LED: Troubleshooting. Finish Flash LED.	BREADBOARD DEADLINE: Flash LED Motors: Gathering materials, Soldering capacitors to	Motors: Program PIC chip for motor testing, finish breadboarding, begin troubleshooting if necessary.

			motors, begin breadboarding	

October (19 Working Days)				
Monday	Tuesday	Wednesday	Thursday	Friday
3	4	5	6	7
Motors: Finish troubleshooting, Finish Motors	BREADBOARD DEADLINE: Motors running PIC controlled Line Detection: Gather materials, start breadboarding.	Line Detection: Program PIC chip for line detection testing, begin troubleshooting if necessary.	BREADBOARD DEADLINE: Line Detection Line Detection: Finish troubleshooting and line detection.	LCD: Gathering materials, finish breadboarding, begin any troubleshooting.
10 (Thanksgiving Monday)	11 (Guhan absent: Student Council retreat)	12	13	14
	BREADBOARD DEADLINE: LCD LCD: Finish troubleshooting. and finish LCD.	Wall Detection: Program PIC chip and breadboard left wall sensor. Begin troubleshooting left wall sensor.	Wall Detection: Breadboard front wall sensor. Begin troubleshooting front wall sensor.	BREADBOARD DEADLINE: Wall Detection Wall Detection: Finish troubleshooting both left and front wall sensors. Finish wall detection.
17	18	19	20	21
Flame Detection: Gathering materials, program PIC chip, begin and finish breadboarding. If necessary, begin troubleshooting	BREADBOARD DEADLINE: Flame Detection Flame Detection: Finish troubleshooting. Finish flame detection	Extinguisher: Gathering materials, program PIC chip begin and finish breadboarding, begin troubleshooting	BREADBOARD DEADLINE: Extinguisher, All Firefighter Breadboarding Labs Extinguisher: finish troubleshooting,	Antonio: Design motor control board. Guhan: Design main board.

			finish extinguisher	
24 (PD Day)	25	26	27	28
	Antonio: Design sensor and fan board. Guhan: Print and etch motor and main boards.	Antonio: Print and etch sensor board and fan board. Guhan: Scrub and wash motor and main boards; start drilling holes in main and motor board.	Antonio: Scrub and wash sensor and fan board; start drilling holes. Guhan: Finish drilling holes in main and motor board, collect components for main board and start soldering.	Antonio: Finish drilling holes in sensor and fan board, collect components and begin soldering. Guhan: Finish soldering main board.
31				
Antonio: Finish soldering sensor and fan boards. Guhan: Collect components for the motor board, start soldering.				

November (19 Working Days)				
Monday	Tuesday	Wednesday	Thursday	Friday
	1	2	3	4
	Antonio: Program PIC chip for motor testing. Guhan: Finish soldering motor board.	Antonio & Guhan: Create a skeleton for the bot chassis, finish soldering any boards left.	CIRCUIT DEADLINE: Firefighter circuit boards. Antonio & Guhan: Finish chassis construction and finish	Antonio & Guhan: Connect motors to PIC chip and begin testing. Begin troubleshooting if necessary.

			programming PIC chip for motor testing.	
7 (Guhan absent: OSLC)	8 (Guhan absent: OSLC)	9 (Instructional Support Day)	10 (Instructional Support Day)	11
CIRCUIT DEADLINE: Motors running PIC controlled Antonio: Finish motor troubleshooting, and finish motors. Mount motors onto chassis.	CIRCUIT DEADLINE: Line Detection. Antonio: Troubleshoot line detection. Finish line detection. . Start and finish LCD. Begin troubleshooting.			Antonio: Program PIC chip for LCD testing. Make sure Guhan is up to speed. Guhan: Mount sensors, fan and line detection onto chassis; connect components to respective boards.
14	15	16	17	18 (PD Day)
CIRCUIT DEADLINE: LCD Antonio & Guhan: Finish and test LED.	Antonio & Guhan: Program PIC chip for wall detection. Troubleshoot wall detection.	CIRCUIT DEADLINE: Wall Detection Antonio: Finish troubleshooting left wall detection. Guhan: Finish troubleshooting front wall detection.	Antonio & Guhan: Program PIC chip for flame detection. Begin troubleshooting flame detection.	
21	22	23	24	25
CIRCUIT DEADLINE: Flame Detection	CIRCUIT DEADLINE: Extinguisher	Antonio: Create test program for an initial run in	Antonio & Guhan: Finish and troubleshoot	Antonio & Guhan: Run the firefighter in

Antonio & Guhan: Finish troubleshooting flame detection.	Antonio & Guhan: Program PIC chip for extinguisher. Troubleshoot extinguisher	maze 1. Guhan: Check constructed design for any flaws or errors, fix as necessary.	test program, install onto PIC chip.	maze 1, room 1. Note observations and any possible improvements.
28 (Guhan absent: Musical matinee)	29 (Guhan absent: Musical matinee)	30 (Guhan absent: Musical matinee)		
Antonio: Refine code according to previously taken observations. Install code onto the PIC chip and run the bot in maze 1, room 1.	Antonio: Run the firefighter in maze 1, room 2. Note results and any observations.	Antonio: Run the firefighter in maze 1, room 3. Note results and possible improvements.		

December (16 Working Days)				
Monday	Tuesday	Wednesday	Thursday	Friday
			1	2
			Antonio & Guhan: Catch Guhan up with recent progress. Run the firefighter in maze 1, room 4. Update code for optimal run (sub 20 seconds) in maze 1. Install onto PIC chip.	Antonio & Guhan: Make final code or design improvements specific to maze 1. Run the firefighter in maze 1.
5	6	7	8	9
Antonio & Guhan: Perform an initial run of the firefighter in	Antonio & Guhan: Run the firefighter in maze 2, room 1.	Antonio & Guhan: Make necessary changes to code	Antonio & Guhan: Complete maze 2, room 1. Run	Antonio & Guhan: Make necessary changes to code

maze 2. Note the results and observations.	Note results and any possible improvements.	or design as needed.	maze 2, room 2 and note observations.	or design as needed.
12	13 (Guhan potentially absent: vacation)	14 (Guhan potentially absent: vacation)	15 (Guhan potentially absent: vacation)	16 (Guhan potentially absent: vacation)
Antonio & Guhan: Run the firefighter in maze 2, room 2. Note results and any possible improvements.	Antonio: Make necessary changes to code or design as needed.	Antonio: Run the firefighter in maze 2, room 3. Note results and any possible improvements.	Antonio: Make necessary changes to code or design as needed.	Antonio: Run the firefighter in maze 2, room 4. Note results and any possible improvements.
19 (Guhan potentially absent: vacation)	20 (Guhan potentially absent: vacation)	21 (Guhan potentially absent: vacation)	22 (Guhan potentially absent: vacation)	23 (Antonio & Guhan potentially absent: vacation)
Antonio: Optimize code for all of maze 2.	Antonio: Optimize code for all of maze 2.	Antonio: Optimize code for all of maze 2.	Antonio: Optimize code for all of maze 2.	

January (16 Working Days)				
Monday	Tuesday	Wednesday	Thursday	Friday
9	10	11	12	13
Antonio: Catch Guhan up with anything he may have missed. Guhan: Look over any observations Antonio made.	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).
16	17	18	19	20
Antonio & Guhan: Optimize bot for ideal results in	Antonio & Guhan: Optimize bot for ideal results in	Antonio & Guhan: Optimize bot for ideal results in	Antonio & Guhan: Optimize bot for ideal results in	Antonio & Guhan: Optimize bot for ideal results in

maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).
23	24	25	26	27
Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).	Antonio & Guhan: Optimize bot for ideal results in maze 1 (sub 20 seconds) and maze 2 (sub 25 seconds).
30 (Final Day of Semester 1)				
Antonio & Guhan: Attempt final runs in maze 1 and maze 2.				