Team Rescue X:
Autonomous Firefighting Robot

Team Members
Project Manager: Shervin Haghgouy
Parts Manager: Shervin Nikravesh
Website Designer: Vincent Thomas
Repository Manager: Alexander Camp
Report Editor: Nico Enriquez
PowerPoint Editor: Jerin Givianrad

Sponsor: San Diego State University
ECE 490 / Fall 2016

Abstract

The objective for our design project is to build a firefighting robot capable of extinguishing a candle. It will be able to navigate a course using a Raspberry Pi camera, and will identify the candle using digital filters. The water-based extinguisher will put out the burning candle, all while avoiding obstacles.

Finally, the robot will be able to differentiate between the true candle and electronic imposters posing as decoys. The physical design will encompass an aluminum chassis, two DC motors with encoders, a water pump and reservoir, and a microcontroller that will act as the brain of the system.

Project Description

The overall goal of this assignment will be to design and fabricate a simplistic, firefighting robot to be entered into the Engineering Department's firefighting robot competition. Each team of engineers will be given a set budget of $800 to purchase parts and equipment necessary to build their entrant within the guidelines of the contest.

We will then use our robots to search a rudimentary obstacle course to extinguish a candle that will burning somewhere within. All teams will be given 100 mL of water in order to accomplish their task, and must also avoid several electronic decoy candles during this process.

Our team robot will be designed to hunt by sight using an eight megapixel Raspberry Pi camera that is hooked into a PIC32 microcontroller. This camera will be able to identify certain frequencies of light and detect object motion in order to locate the true candle. Using the camera in this way will also allow for object avoidance. Additionally, the microcontroller itself will

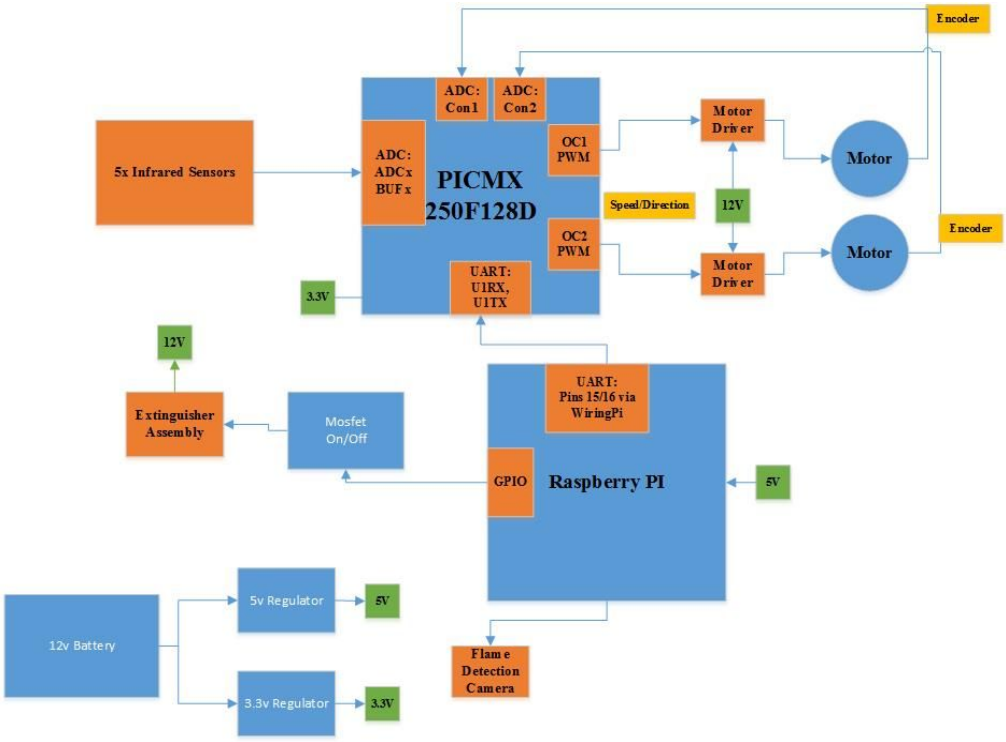
handle logical commands, and PWM (Pulse Width Modulation) to run the motors properly with respect to direction, speed, and positioning of the water pump.

The device will be powered with a simple motor driver carrier, two brushless DC motors, and a 11.1V rechargeable Lipo (Lithium Polymer) battery. Voltage regulators will be employed at strategic areas within the system to ensure that the components will not be damaged, along with capacitors and heatsinks to stave off voltage drops and hot spots respectively. The motor driver carrier will be powered with the same voltage as the microcontroller, while the motor will be fed a much higher voltage.

The physical appearance of the robot will be a little bigger than a notebook, with parts inside connected to a main motherboard for convenience. This will help us to save on wasted space inside the device and will reduce the overall weight as well. The robot will be equipped with a thin aluminum chassis, a front stabilizer wheel, two plastic wheels attached to the motors, a water reservoir, and a water pump to spray the target. Keeping the water pump and reservoir outside of the robot will be crucial to protect the delicate components inside.

Design

Block Diagram



Performance Requirements

Motors:
Min speed: 0.75 m/s
Max speed: 2 m/s
Gear ratio: 100:1
Specs at 12 V: 100 RPM and 300 mA free-run

Sensors:
Sharp GP2Y0A21YK0F Analog Distance Sensor
Detection range: 10 to 80 centimeter
Min operating voltage: 4.5 V
Max operating voltage: 5.5 V
Sampling rate: 26 Hz

Battery:
Minimum Capacity: 5000mAh
Constant Discharge: 20C
Configuration: 11.1V/ 3Cell

Raspberry Pi Camera:
IMX219 Camera Sensor Raspberry Pi Platform Evaluation Expansion Board
No infrared filter
Resolution: 1080P
Megapixels: 8

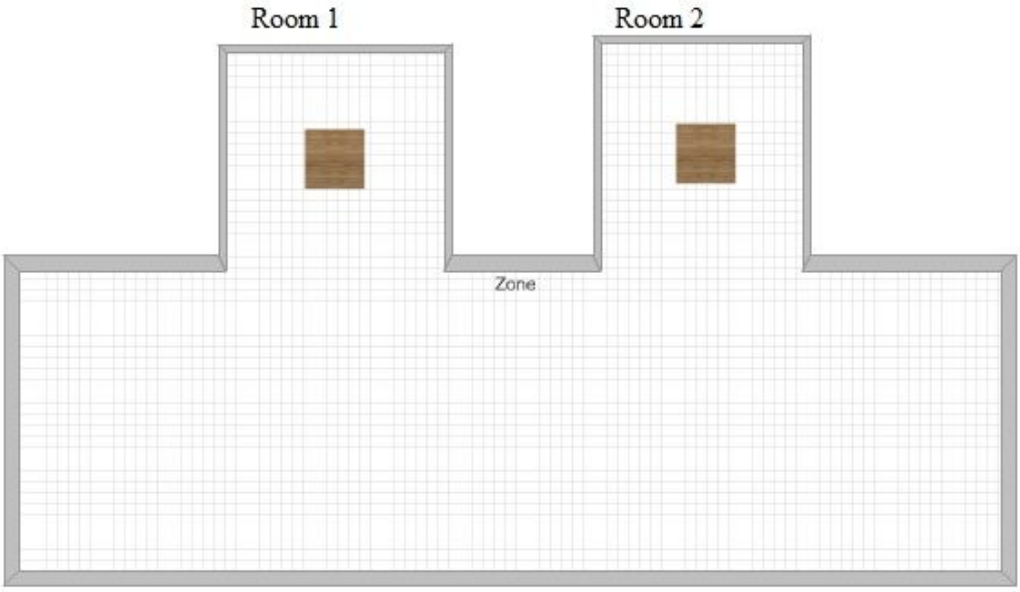
DC Water Pump:
Operating voltage: 12V
Power Consumption: 14.5W
Max Flow: 2 liters/minute

PIC32MX250F128D:
Current Consumption: 0.5 mA/MHz
Max Speed: 50 MHz
A/D Channels: 13
I/O pins: 33
I2C: 2
SPI: 2

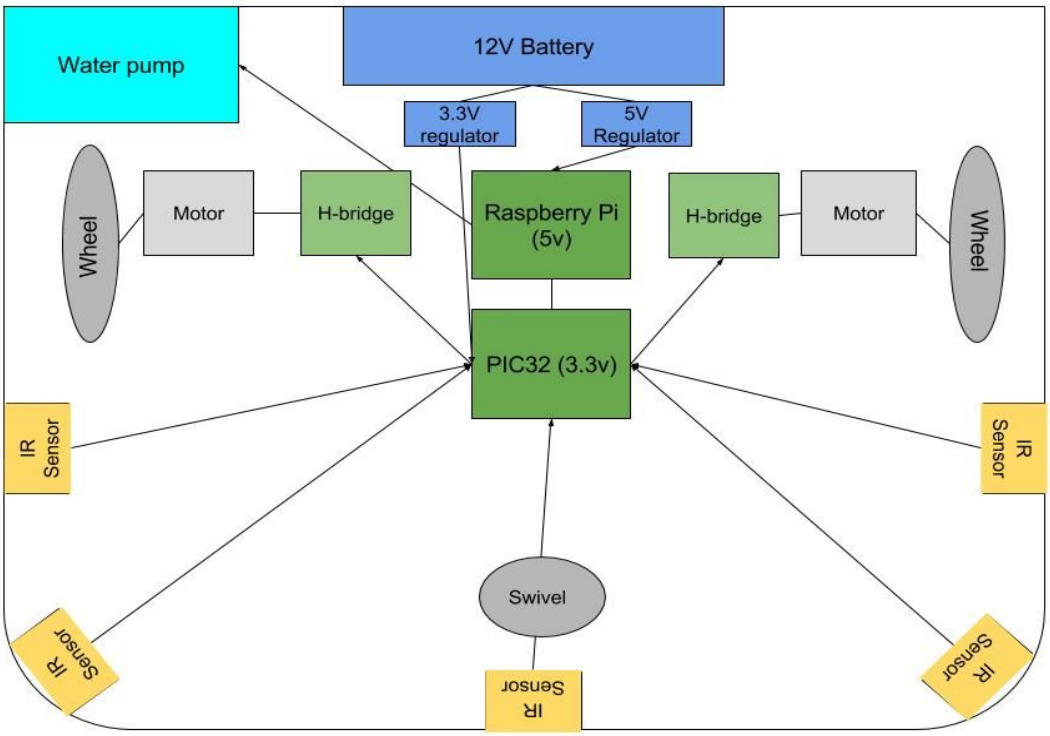
Fire Detection System - Optimized Python script using the open source computer vision library OpenCV and PIL (Python Imaging Library) that will execute on the Raspberry Pi. Using the Raspberry Pi camera module, the algorithm performs image processing on the frames captured through a real-time video stream by applying an RGB masking filter to block out non-infrared light frequencies. Significantly reducing the CPU processing power consumption is a byproduct of the algorithm because the camera will only display infrared light while masking out all other negligible data bits.

Mapping Algorithm & Course Traversal - Create a 2-dimensional matrix that will map the floor plan of the course. Data from the wheel encoders will allow the Raspberry Pi to calculate the distance traveled as well as its relative position in the course. In Round 1, after a successful flame detection the mapping system will save the candle's position in the floor plan and the robot will calculate the shortest path. In round 2, the robot will be able to maneuver directly to the candle.

The course:



Mock-up Illustration



Testing & Verification

Software

a. **Fire Detection**: Extensive testing will be performed on the fire detecting algorithm to ensure robustness and reliability. The picamera will be tested in different lighting conditions while applying several different masking features to analyze the frame data that is produced. To test the tracking accuracy of the camera, we will place the flame at the different positions within the camera's frame and calculate its coordinates relative to the robot. We are aiming for a 5% range within the nominal values.

b. **Mapping**: To test the mapping, we will execute a test program to have the robot traverse the entire course at a fixed speed. To execute turns, the robot will come to a complete halt then pivot at its current position. The robot will drive the wheel opposite of the desired direction in order to move in that direction. Success will be determined if the robot can make it through the entire course without colliding into the perimeters of the course.

c. **Proximity Sensing**: Our robot will be utilizing an infrared sensor in order to detect the distance between the robot and the closest object. To test, we will analyze the ADC data produced by the sensors and output strings of the calculated distance. Our threshold distance for obstacle avoidance is 10 centimeters.

Hardware

a. Water Pump & Reservoir: The water pump will be tested for its effective range as well as the widest angle of spread we can achieve. We will apply some basic physics such as angle of trajectory, initial velocity, and height relative to ground and the candle's height. As for the housing, we expect to mount it in a case such that if a leak were to occur, the water will not affect any of the circuitry.

b. **Motors**: The motors will be driven by various input voltages as well as varying PWMs to analyze their effects on the motor's performance to achieve the required movements. We will sample the wheel encoders to calculate the distance traveled. This information will be utilized in the mapping algorithm. Achieved 0.4 m/s at 12V with a 100:1 gear ratio.

c. **RPI/PIC Integration & Sensors**: Our minimum distance between an object and our robot will be 10-15 cm.This will be tested by the sensor being able to send the information to the PIC to trigger an interrupt which will be tested with an oscilloscope.

The infrared sensors will be connected to the PIC microcontroller via ADC. These sensors are in charge of proximity detection. The Raspberry Pi and the PIC32 microcontroller will be connected via the UART pins on both devices. The PIC32 will act as the main interrupt feature and when an interrupt occurs, the PIC will also interrupt the Pi.

Benchmarks

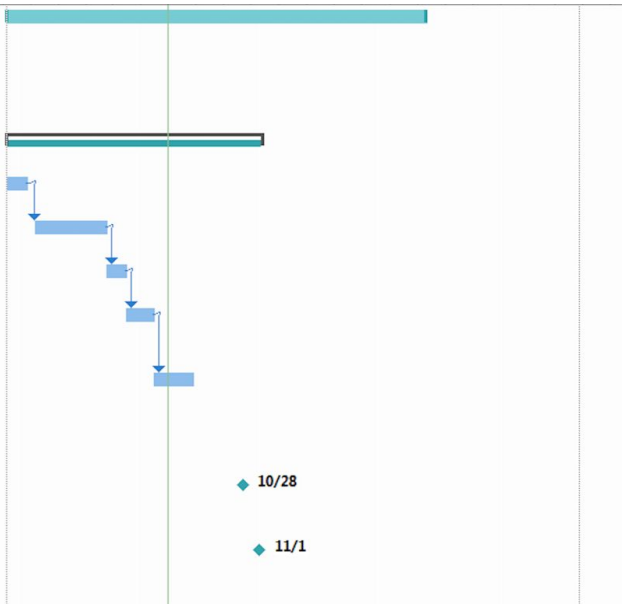
At completion, the robot will be verified to perform the following:

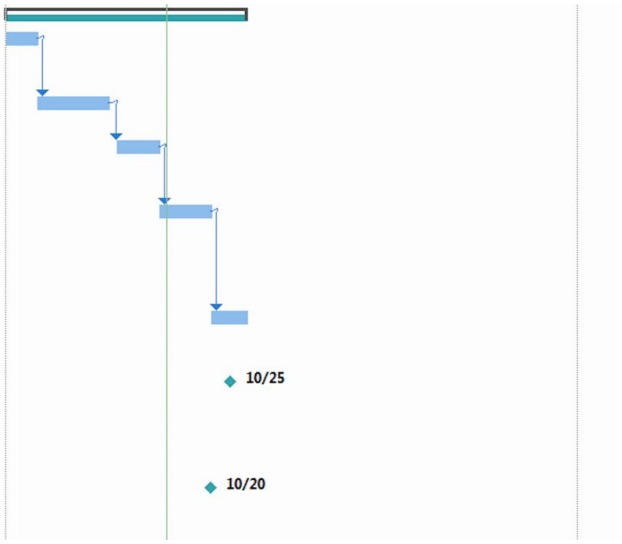
- Robot can traverse the complete course without colliding with obstacles using the proximity sensors.
- The fire detection system can distinguish between the real and fake fire, successfully identifying and extinguishing the real flame.
- The flame can be detected within a half meter distance.
- Achieve a maximum speed of 1 m/s, while establishing a balanced weight distribution on the chassis to prevent uncontrollable movements during acceleration and deceleration.
- Water pump aim is precise and accurately disperses water on the intended target.
- After the first round, the robot will calculate the short distance to the real flame's location and travel directly to it in the proceeding rounds.

Project Management

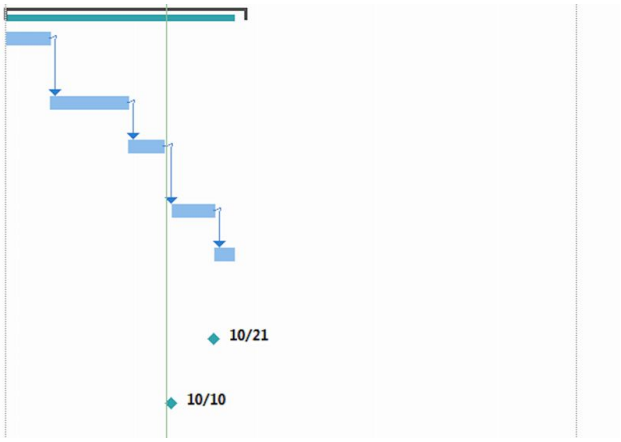
Gantt Chart:

	RESCUE X: Fire Fighting Robot	76 days	M 8/29/16	M 12/12/16	
	HARDWARE				
	Mobility	**47 days**	**M 8/29/16**	**T 11/1/16**	
	Begin researching	5 days	M 8/29/16	F 9/2/16	
	Order Part	14 days	M 9/5/16	T 9/22/16	8
	Begin testing motor	3 days	F 9/23/16	T 9/27/16	9
	Begin testing motor with H-Bridge	5 days	W 9/28/16	T 10/4/16	10
	Begin testing motor and H-bridge with PIC32 Microcontroller	8 days	W 10/5/16	F 10/14/16	11
	Have a robot capable of going straight	0 days	F 10/28/16	F 10/28/16	
	Robot capable of turning 360 degrees	0 days	T 11/1/16	T 11/1/16	
	Proximity Sensor	**45 days**	**M 8/29/16**	**F 10/28/16**	
	Begin researching different	6 days	M 8/29/16	M 9/5/16	
	Order parts	14 days	T 9/6/16	F 9/23/16	18
	Begin testing IR proximity Sensor	9 days	M 9/26/16	T 10/6/16	19
	Begin testing IR proximity Sensor with PIC32 Microcontroller	9 days	F 10/7/16	W 10/19/16	20
	Test IR sensor with PIC32 and Raspberry pi	7 days	T 10/20/16	F 10/28/16	21
	Determine different voltage levels for different distances	0 days	T 10/25/16	T 10/25/16	
	Microcontroller is able to send data to	0 days	T 10/20/16	T 10/20/16	

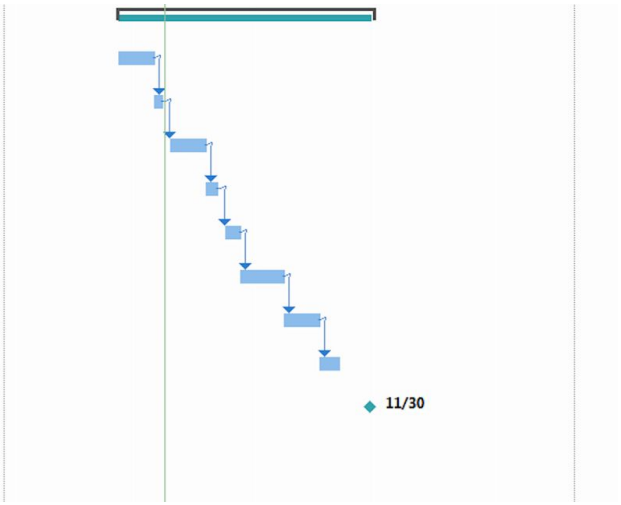




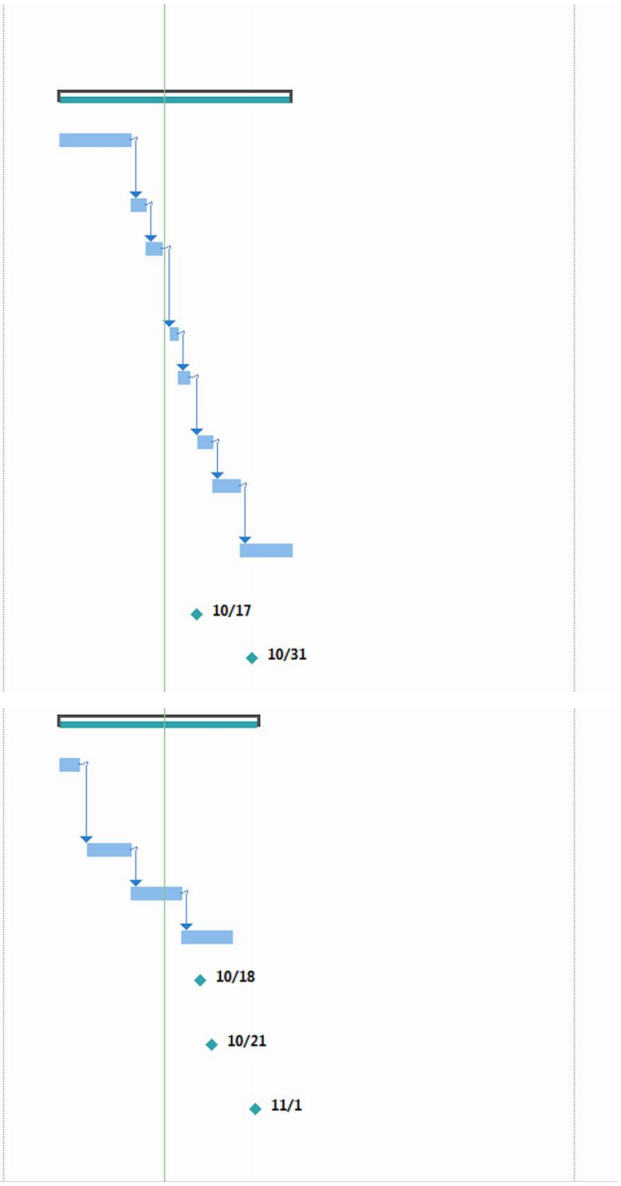
	Water Pump	45 days	M 8/29/16	F 10/28/16	
	Begin researching waterpump best	9 days	M 8/29/16	T 9/8/16	
	Order Part	14 days	F 9/9/16	W 9/28/16	28
	Begin testing pump for range capabilities	7 days	T 9/29/16	F 10/7/16	29
	Test pump with PIC32	9 days	M 10/10/16	T 10/20/16	30
	Verify that PIC32 sends data to Raspberry Pi	3 days	F 10/21/16	T 10/25/16	31
	Microcontroller capable of turning pump on	0 days	F 10/21/16	F 10/21/16	
	Turn pump on/off using	0 days	M 10/10/16	M 10/10/16	



	Solid Works	47 days	T 9/27/16	W 11/30/16	
	Design the bese for robot	7 days	T 9/27/16	W 10/5/16	
	Manufacture base for testing	2 days	T 10/6/16	F 10/7/16	38
	Begin draft design of chassis	7 days	M 10/10/16	T 10/18/16	39
	Manufacture the design	3 days	W 10/19/16	F 10/21/16	40
	Evaluate the design for issues	4 days	M 10/24/16	T 10/27/16	41
	Adjustment period	7 days	F 10/28/16	M 11/7/16	42
	Begin the final design	7 days	T 11/8/16	W 11/16/16	43
	manufacture final design	3 days	T 11/17/16	M 11/21/16	44
	Have a fully funtional robot with all components in place	0 days	W 11/30/16	W 11/30/16	



	SOFTWARE				
	Rasperry pi	**43 days**	**M 9/12/16**	**W 11/9/16**	
	Begin researching and about pin layout	14 days	M 9/12/16	T 9/29/16	
	Install OpenCV on Raspberry Pi	2 days	F 9/30/16	M 10/3/16	52
	Write a simple color detection algorithm with a JPEG image	4 days	T 10/4/16	F 10/7/16	53
	Add camera capture mode	2 days	M 10/10/16	T 10/11/16	54
	Verify that initial algorithm works on RPI w/	3 days	W 10/12/16	F 10/14/16	55
	Add live video stream function	4 days	M 10/17/16	T 10/20/16	56
	Perform Color detection in video	5 days	F 10/21/16	T 10/27/16	57
	Implement a mapping procedure	9 days	F 10/28/16	W 11/9/16	58
	Be able to track flame	0 days	M 10/17/16	M 10/17/16	
	Determine flame	0 days	M 10/31/16	M 10/31/16	
	Microcontroller	**37 days**	**M 9/12/16**	**T 11/1/16**	
	Research microstick development board	5 days	M 9/12/16	F 9/16/16	
	Learn the PWM pins and test	9 days	M 9/19/16	T 9/29/16	64
	Learn A/D pins and test	9 days	F 9/30/16	W 10/12/16	65
	Learn how to use interupt and test	9 days	T 10/13/16	T 10/25/16	66
	Test IR sensor and verify correct response	0 days	T 10/18/16	T 10/18/16	
	Write simple code to turn on motor	0 days	F 10/21/16	F 10/21/16	
	Contol motors individually to be able to do discrete turns	0 days	T 11/1/16	T 11/1/16	



Project Plan

Software Resource Requirement:

Effort	Task Title	Resource
7 Days	Track flame	Using the Raspberry pi and a camera, the software team will be able to detect the blue outline of flame.
14 Days	Test IR proximity Sensors	Using the PIC32 and Sharp IR proximity sensor, the software team will be able to determine the sensor's response from 10 - 80 cm.

7 Days	Write a simple code to control motor	Using the PIC32, the software team will be able to control the motor speed based on changes in Pulse Width Modulation.
5 Days	Turn water pump on and off	Using the PIC32, software team will be able to control turn the water pump on and off.

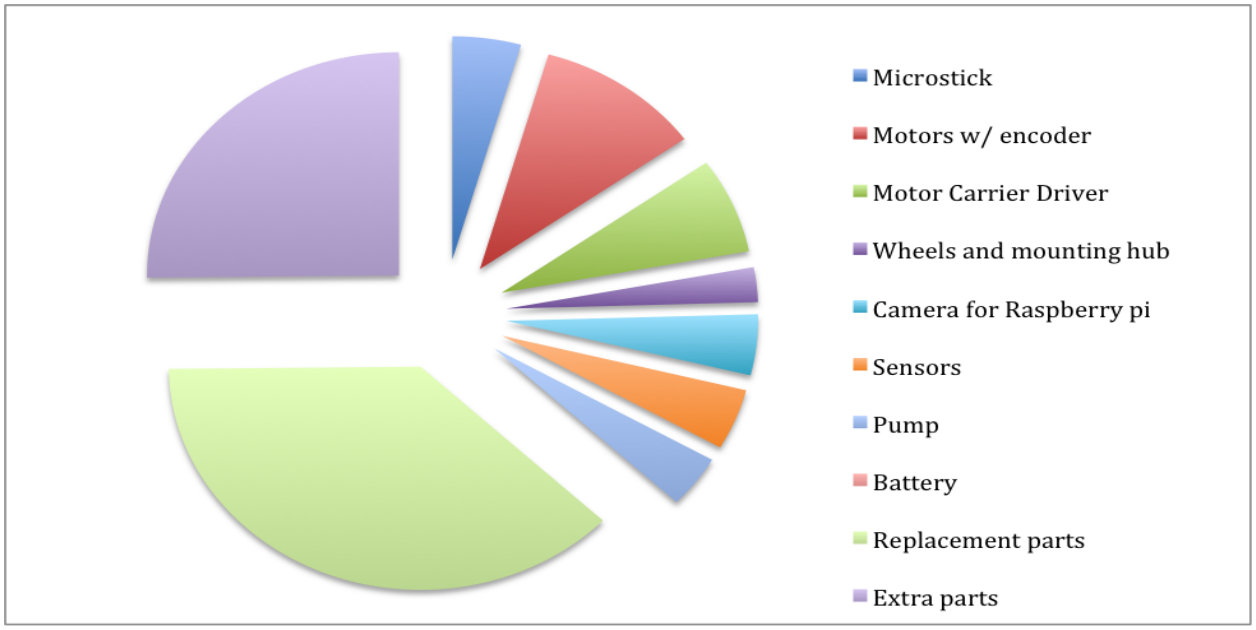
Hardware Resource Requirement:

Effort	**Task Title**	**Description**
14 Days	Robot capable of going straight	Combining the hardware and software developed by our team to move robot in a straight line.
7 Days	Robot capable of turning 360 degrees	Combining the hardware and software developed by our team to allow the robot to rotate 360 degrees.
60 Days	Have a fully functional Robot	Develop a robot capable of avoiding obstacles, detecting a flame, and extinguishing a flame.

Budget

Microstick	$35
Motors w/ encoder	$85
Motor Carrier Driver	$55
Wheels and mounting hub	$20
Camera for Raspberry pi	$35
Sensors	$35
Pump	$30
Battery	$0
Raspberry Pi	$40
Replacement Parts	$200
Extra Parts	$265

Total	$800


Microstick
Motors w/ encoder
Motor Carrier Driver
Wheels and mounting hub
Camera for Raspberry pi
Sensors
Pump
Battery
Replacement parts
Extra parts