\pagebreak

\begin{center}

Auto Vehicle

\end{center}

by Khemar Bryan, Jan Yalda, Bilal Al-fanous

Date: January 20, 2017

\pagebreak

Declaration of Joint Authorship

We, Jan Yalda, Bilal Al-fanous & Khemar Bryan confirm that this work submitted for assessment is our own and is expressed in our own words. Our best effort was made to split this project equally. Jan Yalda worked on the hardware and database. Bilal Al-fanous worked on and maintained the web site. Khemar Bryan worked on the mobile application. We all collaborated to ensure that all parts of this project were able to connect and communicate with each other. All direct or indirect sources used are acknowledged as references.

Date:

Approved Proposal

Proposal for the development of AutoVehicle

Prepared by Khemar Bryan, Jan Yalda, Bilal Al-Fanous
Computer Engineering Technology Students

Project Website : https://khemar1.github.io

Proposal Table of Contents

Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include distance sensors and actuators for controlling the movement of the vehicle. The database will store the mapped area. The mobile device functionality will include remote controlling the vehicle and displaying the mapped area and will be further detailed in the mobile application proposal. We will continue to work together this Winter semester as we all built similar hardware last term and worked on the mobile application. The hardware was completed in CENG 317 Hardware Production Techniques independently and the application was completed in CENG 319 Software Project. These will be integrated together this semester in CENG 355 Computer Systems Project as a member of this 3-student group.

Background

The problem solved by this project is how to have a robotic vehicle that will be controlled as an RC car with a mobile application, and which can work independently as an autonomous car.

The Hardware which consists of the vehicle's chassis where all the parts are either connected or mounted on which includes the actuators, the H-bridge driver, the ultrasonic sensors and the brain of it all the Raspberry Pi 3. The vehicle's sensors are used to avoid obstacles that are detected and at the same time send data to be used to map an area that will be displayed in the mobile application. Through Wi-Fi the ability to remote control the vehicle is possible using the mobile application.

We have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" [1] and have found and read the following three articles which provides insight into similar efforts.

- Algorithm Fusion for Feature Extraction and Map Construction from SONAR Data(Ismail & Balachandran, 2015)
- SLAM for robot navigation(Temeltas & Kayak, 2008)
- An Open-Source Scaled Automobile Platform for Fault-Tolerant Electronic Stability Control(Katzourakis, Papaefstathiou, & Lagoudakis, 2010)

In the Computer Engineering Technology program, we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java,(Ismail & Balachandran, 2015)
- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of databases from CENG 256 Internet Scripting; and,
- Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.
Phase 2 System integration.
Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration

The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that I have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	**Hrs**	**Notes**
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 12th, 2016 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.
Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 8th, 2017 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	
Material Estimates	**Cost**	**Notes**

Phase 1		
Raspberry Pi 3 Kit	$99.00	CanaKit
4WD Robot Platform	$42.94	Creatron Inc
L298N H-Bridge	$15.82	Creatron Inc
Mini Bread Board	$3.67	Creatron Inc
Jumper Wires(3-sets)	$6.86	Creatron Inc
HC-SR04 Ultrasonic Sensors (4)	$22.56	Creatron Inc
Standoffs F/F	$3.38	Sayal Elec
Portable Battery	$45.19	Scosche
Philips Head Screws	$5.64	Sayal Elec
Phase 1 Total	**<$200.00**	
Phase 2		
Materials to improve functionality, fit, and finish of project.		
Phase 2 Total	**TBD**	
Phase 3		
Off campus colocation	<$100.00	An example: [4].
Shipping	*TBD*	
Tax	*TBD*	
Duty	*TBD*	
Phase 3 Total	**TBD**	

Concluding remarks

This proposal presents a plan for providing an IoT solution for AutoVehicle. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project.

Abstract

So far human controlled robots have been great; they are able to complete many tasks instructed by the user that's controlling them. As technology advances in our world we have learned that a new form of robotic programming has been developed to allow the robot to complete tasks on their own, independent of the user. Artificial Intelligence(AI) has enabled developers to create robots that are capable of operating on their own, making them even more useful and easier to operate. To have a robot that is able to independently map an unknown area would be great in many cases. One such case would be: if an area that a user wants to map is small or has a small entrance where they may not be able to physically enter, it is useful to have a small robot that can complete that task on its own and return a visual map of that area to the user. The Auto Vehicle has the ability to autonomously map an unknown area with the press of a button on the mobile application. It independently avoids obstacles in its path while always having the knowledge of its position in the unknown area. The vehicle will record the data in the form of coordinates to be sent to a database which can be retrieved by the mobile & web applications. The coordinates, when retrieved by the mobile application, will then be displayed in the form of a 2D map of the area.

Table of Contents

List of Illustrations

Introduction

There are numerous areas such as: mines and sewer systems which are beyond the reach of humans. These areas may either be too rigid or have entrances that may be too small for an average sized person to reach. This report describes the process followed for the design and utilization of an autonomous vehicle with the ability to map areas that may represent an inconvenience for humans to enter. The autonomous vehicle in question will be fully autonomous and will use ultrasonic sensors to navigate and record coordinates in an area which will be sent to a database. This database can then be retrieved by website to display them to specific users when they log in. The mobile application will retrieve the coordinates from the database in addition to a drawn map of the area which it will be able to display to the user. The application will also be able to remotely control the car through the usage of an onscreen joystick, through a Bluetooth or WiFi connection. The information presented will be centered on the design of the autonomous vehicle in addition to the mapping algorithm which it will implement.

System Requirements Document

2.1 Introduction

2.1.1 Purpose The purpose of this small scale vehicle is to be able to autonomously navigate and map an area. Autonomous vehicles are the current trend in technology and this vehicle will be allow a user to make a map of an area without having to physically carry out the task.

2.1.2 Definitions An autonomous vehicle is a vehicle that is capable of sensing its environment and navigation without human input *add citation

2.1.3 Overview This vehicle is made using a combination of Raspberry Pi 3, Ultrasonic sensors, and a motor driver. These components are built onto smart car chassis which will be used to move. The ultrasonic sensors are used to detect obstacles and send the information back to the raspberry pi.

2.1.4 Target Audience

The target audience of this project is students with a background in Computer Engineering Technology who may have not taken part in a project of this type before. These students students will typically be in their final year of study.

2.2 Overall Description

The following section will have an overview of the whole system. The basic functionalities and how it interacts with the mobile application will be explained. Lastly the constraints and assumptions of the system will be discussed.

2.2.1 Work Breakdown

Hardware The Auto Vehicle's chassis has all the hardware connected or mounted on it that operates together to achieve the functionalities that are required from it. The most important piece of hardware is the Raspberry PI 3, the microcomputer which has all the other hardware connected to in where it receives data from and send data to, it makes it possible to connect to the mobile application as well to send and receive data. The hardware that sends the data to the microcomputer are the Ultrasonic Sensors which are distance sensors, they are used to detect objects that may appear in front of the vehicle then send a signal to the microcomputer where it process it then manipulates the actuators according to the data that it received, the actuators are DC Motors used to move the vehicle, but before that the microcomputer actually sends data to the H-Bridge the driver which based on the data controls the DC Motors direction.

(Developed by Jan Yalda)

Database The system requires two databases, one will be used to store the users that have signed up to use the mobile application and another to store data of the mapped area. The mobile application user's database is a MySQL database on a free hosting remote server, its main purpose is to store the information of users that sign up for the application. This information consists a username and an encrypted password which will allow the user enter the mobile application. The second database, which is still in development, will be used to receive data from the hardware while it is mapping an area. This data will be stored in the database which can be retrieved by the mobile application to display the mapped area to the user.

(Developed by Jan Yalda)

Web Interface A website will be developed to allow the user can login to his account. Each specific user will be able to see their previous data been collected using the AutoVehicle. One section of the web site will have a small bio for the developers of the project and their contributions. Another page will contain a step by step explanation for those who may wish to recreate the project on their own. In addition to those, a section will be added to allow visitors give us feedback and suggestions on how we may improve the project.

(Developed by Bilal Alfanous)

Mobile Application The AutoVehicle Application is used to communicate with the autonomous vehicle remotely. It has in total 12 java classes which includes 6 activities. The functions of the activities are as follows: Login Activity is the first page the user will see when they open the application, it is where the user enters their information to get their user specific information which is stored in the database; MenuActivity is the page the user sees after they successfully login, they can now choose whether they want to control the car or retrieve a map; The register activity is where new users sign up with their information to be stored in the database. After registering users will be taken back to the login page; The remote control activity has a joystick which the user can use to control the car; the settings activity is where the user can enter the ip address of their car so that they can connect to it; the mapping activity is where the user can see the map that their car has made, it will be blank for new user. The app will work in unison with both the database and hardware to be fully functional. In addition to these various features, the application also has support for both English and French.

(Developed by Khemar Bryan)

2.2.2 Product perspective The system mainly consists of a mobile application which will be used to control the hardware as an RC car, the other use will be to start and stop the mapping functionality of the hardware and to view the mapped area.

The mobile application has a login functionality which requires a user to be existing in the users database. The mobile application sends a request to the database to check if the user credentials provided matches one of the users that exists in the database then sends a response back to the mobile application which indicates if the user exists or not and therefore gives or denies access to the main functionalities of the application.

After a successful login to the application has been made the two main functionalities are presented. First the functionality to use the Auto Vehicle as a remote controlled car, with this functionality the mobile application communicates with the hardware using the internet and sends commands to be received by the hardware and acted upon which are just basic remote control functionalities. The second functionality the mapping of an area communicates with the hardware the same way, it just orders the hardware to do the mapping functionality instead.

2.2.3 Product functions Using the mobile application, users logged in will have two main choices on a menu first the remote control functionality and second the mapping functionality. When the users choose one of them they will be taken to the activity that is made for the functionality, where if its the users first time logging in gets an alert saying that an IP address is required which is the IP address of the Auto Vehicle, the user can just to add it right then or can ignore the alert and provide it another time.

When an IP address is provided users going back to the two activities will have the ability to actually connect to the Auto Vehicle and start any functionality that they wish to perform.

The remote control activity provides the users with the option to connect to the hardware using the connect button and start controlling it using the virtual joy stick provided in the remote control activity and finally when they no longer want to continue with this functionality they have the choice to disconnect from the hardware using the button provided.

The mapping activity has the connect and disconnect buttons that have the same functionality as in the remote control activity, but it also has two other buttons one to start the mapping process and the other to stop it, when the stop button is click the mapped will be send from the hardware to the mobile application and displayed in this activity.

2.2.4 User characteristics The end user of this product will be a person with a desire to map an area without having to carry out the task manually. They will also have access to a smart phone with working Bluetooth.

2.2.5 Constraints, assumptions and dependencies The main constraint on the system is that its dependent on the internet for connection to be made to the database when users want to sign up or login. A connection to the hardware through internet or Bluetooth is also required to be able to access the necessary functionalities from the mobile application. Using Bluetooth to connect also provides a distance issue, in that it requires the receiver to be within a certain range to be able to connect efficiently.

Another constraint is that the mobile application is only available in two languages which are English and French, therefore users that may not understand any of those two languages will not be able to use the application effectively.

An assumption of the system is that it the users to will have an android device that runs on Android API level 19 and above as a minimum requirement to use the mobile application on their smart phones. Users with devices that don't meet the requirements will not be able to download and use the mobile application.

2.3 Build Instructions

2.3.1 Introduction This content of this section will give detailed instructions on how to recreate the project.

Task	*Time Estimate*
Acquire parts	1 day
Assemble chassis	30 minutes
Print PCB(Optional)	2 hours
Solder components onto PCB (if using PCB)	1 hour
Setup sensors on breadboard (Optional if not using PCB)	10 minutes
Setup all wiring and connections to the Raspberry Pi	45 minutes
Configure Rasperry Pi with OS	1 hour
Setup Android Application	40 minutes
Setup Code for on raspberry p for car operation	20 minutes
Testing	1-2 hours

2.3.2 Time Commitment

2.3.3 Budget These prices may vary depending on which supplier you purchase them from. All these parts excluding the Raspberry Pi can be purchased from Creatron Inc. The raspberry pi can be ordered online through amazon. Parts with a "*" can be replaced with suitable alternatives.

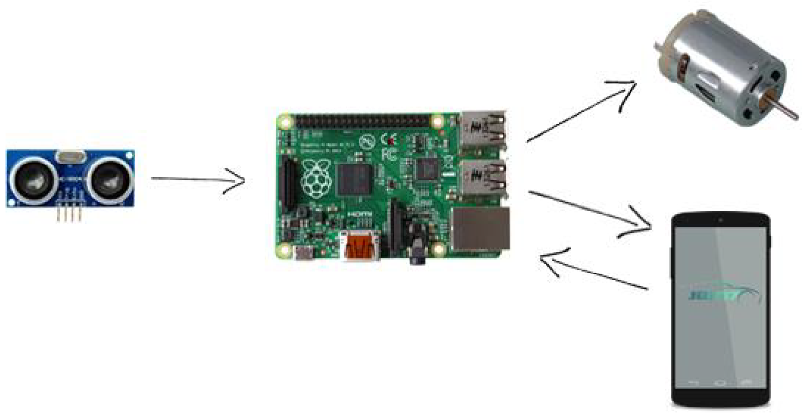

Figure 1:

Item	*Cost(Before Tax)*	*Tax*	*Cost(After Tax)*
Raspberry Pi Kit	$ 99.99	$ 13.00	$ 112.99
Hwydo 4WD Robot Smart Car Chassis*	$ 38.00	$ 4.94	$ 42.94
Energizer Max AA Batteries	$ 4.97	$ 0.65	$ 5.62
HC-SR04 Ultrasonic Sensors (x2)	$ 4.99	$ 0.65	$ 5.64
6“ Male to Male Jumper Wires (x10)	$ 2.97	$ 0.39	$ 3.36
6” Male to Female Jumper Wires (x10)	$ 2.98	$ 0.39	$ 3.37
6” Female to Female Jumper Wires (x10)	$ 2.99	$ 0.39	$ 3.38
L298 Dual Motor Driver-2A	$ 14.00	$ 1.82	$ 15.82
Total	$ 170.89		$ 193.11

2.3.4 System Diagram

2.3.5 Mechanical Assembly To start unbox all the parts that come in the 4WD chassis kit then start by:

- Putting the motor mounts on the chassis
- Putting the actual motors on to the mounts
- Putting the standoffs on the bottom chassis layer
- Then screwing the top layer to the standoffs
- Finally attach all the tires to the motors
- Then find good spots on the chassis to mount the raspberry pi, sensor circuit which is on a bread board and the H-driver
- Place the H-driver in a position close to the motors
- To connect the H-Driver to the motors, follow these steps:

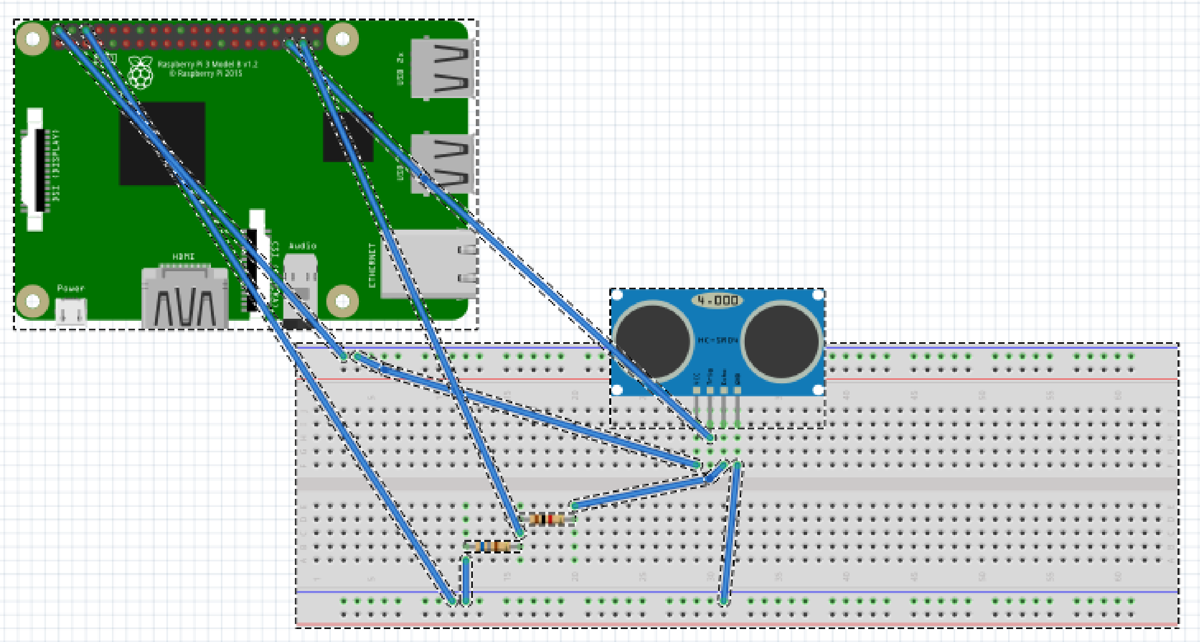

Figure 2:

  - Loosen the screws on the OUT ports (OUT1, OUT2, OUT3, OUT4)
  - Put the red and black leads from the right motors of the car into OUT2 and OUT1 respectively then tighten the screws.
  - Place the red and black wires from the left side into OUT4 and OUT3 respectively then tighten the screws.

- After the H-driver is connected to the motors you must now connect the driver to the raspberry pi and the power supply
  - Loosen the screws on the 5V, round and VCC ports
  - Take the red lead from the power supply and place it into the VCC port then take the black lead and push it into the ground port. Tighten the screws on the VCC port but leave the ground port open for now
  - a male to female wire and plug the header onto pin 2 on your raspberry pi (this is the 5V out pin) then push the male head into the 5V port on the motor driver.
  - Using another male to female wire, plug the header onto pin 6 on the raspberry pi and push the male head into the ground port on the motor driver.
  - You can now tighten the screws on the 5V and ground ports

- And finally place the sensor circuit somewhere on the edge of the chassis so nothing will be interfering with the sensors like the wires.

2.3.6 PCB/Soldering The files needed to recreate the PCB can be found at this link: [PCB layout]

Soldering onto the PCB is a simple task that even beginners can carry out but it is recommended that you have at least basic knowledge of circuitry. Practice before you solder onto your pcb to minimize mistakes

Step 1

Get a soldering iron, a PCB holder and some solder.

Step 2

Place the PCB into the PCB holder and plug the soldering iron in to heat it up

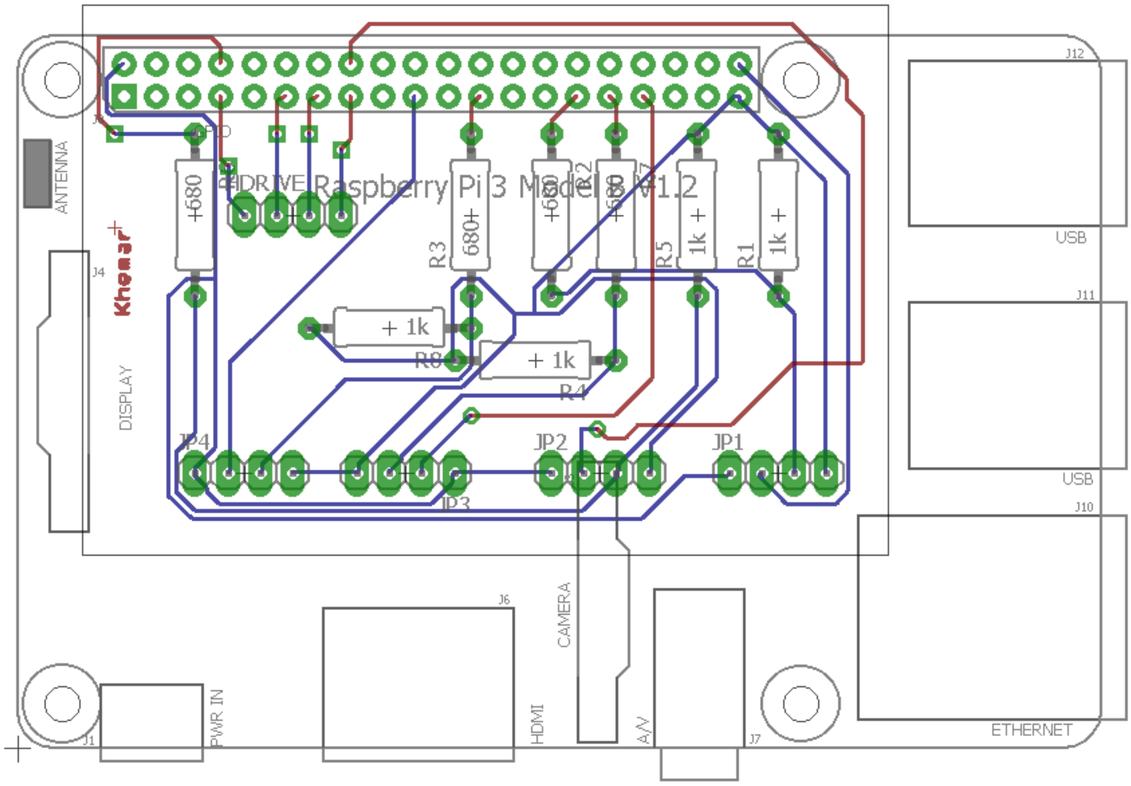

Figure 3:

Step 3

When the LED on the soldering iron has turned green this means you can begin soldering

Start by cleaning the putting some solder onto the iron and wipe it off on a sponge (this is done to clean the iron)

Step 4

Look at your schematic then begin soldering on the necessary parts. (MAKE SURE TO FOLLOW SCHEMATIC)

2.3.7 Power Up In this phase, you will be powering up your car and PCB(if you chose to use one)

Powering up the PCB

- The raspberrypi by default comes loaded with an OS but it is recommended you reinstall it
- Format the SD card then use http://sourceforge.net/projects/win32diskimager/ to write the image file https://downloads.raspberrypi.org/raspbian/images/raspbian-2016-09-28/2016-09-23-raspbian-jessie.zip onto it.
- Using the HDMI cable from the rapsberrypi kit connect the pi to a monitor
- Using an Ethernet cable connect the pi to a source of internet.(or use wifi)
- Put the sd card into your raspberrypi and power it on.
- When the raspberry pi is powered on look for the ip address (hover cursor over internet icon or use ifconifg in the terminal)
- Open remote pc, enter the ip address and connect to the pi.

- Go to the sixofour.github.io repository and look in the ModularSenseHatStripped folder for a file called traffic 2B.c. Download the file to your desktop.
- Put the PCB header onto the raspberry pi's pins
- Go to the terminal and run the traffic 2B.c program, the LED's on the PCB should now start changing colours.

Powering up the Car

- Put the batteries into the battery holder
- If the H Drive is receiving power from the batteries the LED should illuminate
- Remove one of the batteries from the battery holder to turn of the motors as you will need to test if the raspberry pi is powering the motor driver
- Plug in the raspberry pi and the LED on the motor driver should illuminate once again
- Now put the batteries back into the battery holder and all components should be powered up

Website Designing

Mobile Application

The files needed to recreate the android application can be found at the following link: https://github.com/bilfnous/AutonomousVehicle

You will need a physical device that runs on at least Android APK level 19(4.4 & above) to run this application. A virtual android device is not recommended as it does not have access to bluetooth.

- Follow the link above to go to the repository where the android code is stored.
- Download the repository or follow this direct link https://github.com/bilfnous/AutonomousVehicle/archive/master.zip
- Unzip the folder
- Open Android Studio
- Go to file, new, import project, locate the project you just downloaded and click OK
- After importing the project, wait for gradle to load then make sure that all the java classes from the repository can be found there.
- Before continuing, enable developer options on your android device by going to settings, software info
- Tap build number multiple times until a pop up appears saying you are now a developer
- Go back to the settings menu and enter developer options, from there turn on developer options and enable USB debugging
- Plug your device into your computer
- Run the project in android studio by click the green triangular button or use the run menu.
- After clicking the run button you will be met with a screen that asks you to choose which device you want to run the application on, choose your device

2.4 Specific Requirements

2.5 Progress Reports

February 3, 2017 Status Summary

With the remote-control part of the project completed we are now focused on the mapping functionality and making adjustments to improve the hardware and the software sides of the project. The android application is being modified to prevent crashing when an incorrect IP address is entered and ways of implementing the mapping functionality is being viewed to have a better understanding of how it works.

Project Overview

The mapping functionality is still in the research phase.
The android application is being modified to prevent crashing and incorporate Bluetooth connectivity.

The web site is under development.

Problems and Opportunities

Mapping is a very hard concept to comprehend and implement. We've gathered information from various sources which mainly include complex algorithms on how to do it but we're not quite sure how we can use it. This provides us with the opportunity to collaborate with someone who has experience with robotic mapping.

During the last semester, we used a WiFi/Internet connection to communicate with the car. This worked well from our homes but connecting to an enterprise WiFi proved to be an issue. This has provided us the opportunity to learn about Bluetooth connectivity and how we can incorporate that into our project.

Budget Overview

No changes to the budget thus far, we haven't had to add anything to the project.

Conclusions

Overall the project is still in development. Our goals for this month are to prevent the application from crashing, understand and implement the mapping functionality and have a skeleton for the website.

February 17, 2017 Status Summary

This week we met with our collaborator to discuss our options, as it relates to mapping. After meeting with the collaborator new solutions were introduced, and his ideas were taken into consideration. The rest of our time was focused on enabling connectivity on our car. A solution has been found to connect the vehicles to enterprise Wi-Fi. Project website started to take its overall design and outlines.

Project Overview

The mapping functionality is still in the research phase but will now see some progress after we take into consideration the options brought to us by our collaborator. The Bluetooth feature for now will be on hold, as we've found a solution for the Wi-Fi connectivity. The main page of the website is being coded and the final outline of the website is taking shape.

Problems and opportunities

Jan: A meeting with a collaborator in the mapping field took place, and some very interesting ideas were introduced. An electronic peripheral might be used, such as IMU to enable us to sense angles. Another solution known as the flood fill algorithm, is also being considered along with other ideas to do door mapping.

Khemar & Jan: Bluetooth proved to be a strenuous task and it will be on hold for the time being. While working on the Bluetooth functionality the Wi-Fi connectivity aspect was fixed in the application. In addition to that, a solution to our enterprise Wi-Fi issue was found by Jan. Moving forward, Wi-Fi will be our main way of connectivity until we learn more about Bluetooth.

Bilal: Website development has been going smoothly up to now, and there is not any major problem that worth mentioning, the major development was on the skeleton and overall design of the website.

Link for AutoVehicle's website:

www.munro.humber.ca/~n00994056/index.html

Budget Overview

After meeting with the mapping collaborator, we may have to adjust our budget. Among other things, we have taken into consideration an IMU, which should cost around 50$.

Conclusion

Overall, there has been major progress in regards to the connectivity aspect of the project, our goals have been met for this month with the app no longer crashing when and invalid IP address is entered. We now have a clear idea of the options we have to aid us in creating mapping functionality. The general outline of the website has been shaped and for the next month the website will continue to be in development, some initial mapping code and testing should take place.

2.6 Testing

Unit Testing *Testing the H Drive & Battery Power Up*

Place the batteries into battery holder and the LED should turn on and shine red

Testing H Drive & Raspberry Pi Power Up

Plug the raspberry pi into a power source and the LED should turn on.

Testing Raspberry Pi

- Plug the raspberry pi into a power source and connect it to a monitor.
- You should be greeted by a screen asking you to enter your login information
- If you haven't made any modifications then the default credentials are Username: pi Password: raspberry
- Open the browser and go to any website
  If the pi is connected to internet you should have no problem

Testing the motors

- Ensure all connections are made according to the build instructions
- Power up the raspberry pi and login
- Download fwdback.py
- Go to your terminal and go to the directory where you downloaded fwback.py
- Run the program using sudo python fwdback.py
- If all connections were made correctly then the wheels should move forward and backward

Testing the sensors

- To test the sensor the steps are similar to that of the motors.
- Power up the raspberry pi and login
- Download sensor2.py or copy the code below

```
import RPi.GPIO as gpio

import time
```

```
def distance(measure='cm'):

gpio.setmode(gpio.BOARD)

gpio.setup(12, gpio.OUT)

gpio.setup(16, gpio.IN)


gpio.output(12, False)

while gpio.input(16) == 0:

nosig = time.time()

while gpio.input(16) == 1:

sig = time.time()

tl = sig - nosig

if measure == 'cm':
distance = tl / 0.000058
elif measure == 'in':
distance = tl / 0.000148
```

```
else:

print('improper choice')

distance = None


gpio.cleanup()

return distance


print(distance('cm'))
```

- Open your terminal and go into the directory where you downloaded the program
- Run the program using the command: sudo python sensor2.py
- Place an object in front of the sensor and it should return the distance of the object to the screen.

Product Testing When unit testing has been complete and all problems resolved, you can now test the car as a whole. Follow these steps to test that all components work together:

- Power up the raspberry pi
- Put in the batteries
- Connect the raspberry pi to a monitor or use remote pc to view the desktop
- Login to the raspberry pi
- Download auto1.py
- Elevate the car so that it isn't touching the ground (You can place it on top of something)
- Go to your terminal and navigate to where you downloaded auto1.py
- Run the program using sudo python auto1.py
- The wheels should move forward continuously
- Wave your hand in front of the sensors and the wheels should change direction

You can further test the car by placing it in an open area and watching it go

- Follow the previous steps but use remote pc to connect instead of using a monitor. This is important because you want the car to move freely
- Place the car in the open area
- Run the program again
- The car should move forward continuously
- When the car senses an obstacle it should reverse and change direction(whether it be left or right. This is choses randomly by the program)
- If it senses an obstacle in the direction it turned to then it will turn once again.
- The car will continuously do this until it cannot find a path to continue

3. Conclusion

This autonomous vehicle is fully autonomous and uses ultrasonic sensors to navigate and record coordinates in an area which can be sent to a database. This database can then be retrieved by a website to display them to specific users when they log in. The mobile application retrieves the coordinates from the database in addition to a drawn map of the area which it will be able to display to the user. The application can also remotely control the car through the usage of an onscreen joystick, through a Bluetooth or WiFi connection. The information presented is be centered on the design of the autonomous vehicle in addition to the mapping algorithm which it implements.

4. Recommendation

As the project is now, it can be further improved in many aspects. A power bank can be used to power the motor instead of the AA batteries, this will allow the motor to run for longer periods and it can also be recharged when not in use. Currently the algorithm we use to do the mapping is very simple and specific to our needs. There are other options that will allow the mapping to be done more efficiently albeit with more complicated code.

5. References

Ismail, H., & Balachandran, B. (2015). Algorithm fusion for feature extraction and map construction from sonar data. In *IEEE Sensors Journal* (Vol. 15, pp. 6460–6471). https://doi.org/10.1109/JSEN.2015.2456900

Katzourakis, D. I., Papaefstathiou, I., & Lagoudakis, M. G. (2010). An open-source scaled automobile platform for fault-tolerant electronic stability control. *IEEE Transactions on Instrumentation and Measurement*, *59*(9), 2303–2314. https://doi.org/10.1109/TIM.2009.2034575

Temeltas, H., & Kayak, D. (2008). SLAM for robot navigation. In *IEEE Aerospace and Electronic Systems Magazine* (Vol. 23, pp. 16–19). https://doi.org/10.1109/MAES.2008.4694832