HOWARD UNIVERSITY ECE DEPARTMENT
SPRING 2009 SENIOR DESIGN

AUTONOMOUS MAP FOLLOWER

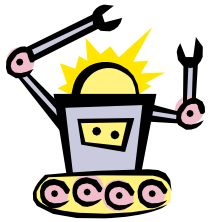
Progress Report

Ozioma Obiaka
Nicholas Baker
Clifton Lomax
Isaac Collins
Endor Cooper

Overview

- Introduction
- Problem Formulation
- Current Status of Art
- Selection Criteria
- Top Design Selection
- Final Solution Product
- Demo Prototype Product
- Progress
- Milestones vs. Outcomes
- Identified Risks and Control
- Future Plans
- Timeline
- Review

Introduction



- The objective of this project is to design a system that allows a vehicle to drive autonomously from one destination to another

- By replacing the human driver with a system that can autonomously drive the vehicle, it is likely that driving can be made safer for both commuters and pedestrians.

- The team will deliver a scaled version of the prototype which will mirror the full functionality of the design

Problem Formulation

- Calculate and drive a route
- Stop at destination within 5% error margin
- Alert the user upon arrival at destination
- Allow the user to abort on-going destination navigation
- Change desired destination during on-going navigation
- Automatically calculate and return to departure location
- Turn off remotely

Current Status of Art

GPS

- Strength- With GPS it is proven that the idea of an autonomous car can follow a map and reach its destination. i.e. DARPA
- Weakness- The map follower could not be an independent part of the vehicle and it must be integrated with other components in the vehicle for it to successfully drive autonomously from a start destination to a final destination.

Electric Compass

- Strength- When the external interferences are eliminated the electric compass becomes more accurate with navigation.
- Weakness- Sensitivity to external interferences of the magnetic field, the electric compass itself is not accurate enough to be used for localization

Multilateration

- Strength- It is commonly used in civil and military surveillance applications to accurately locate an aircraft, vehicle or stationary emitter.
- Weakness- Integrating it with software to help navigate the vehicle instead of locate or track it.

Selection Criteria

- Ease of use: this was one of the most important factors we considered as it would determine how user friendly the design solution was.
- Accuracy: this was also one of the most important factors in choosing a design as it was directly related to the function of the system in moving the vehicle from one location to another.
- Safety: was also an important attribute for us to consider in making a decision and so we gave it a weight of 30
- Price: we considered price because we were working with a budget constraint as determined by the ECE department

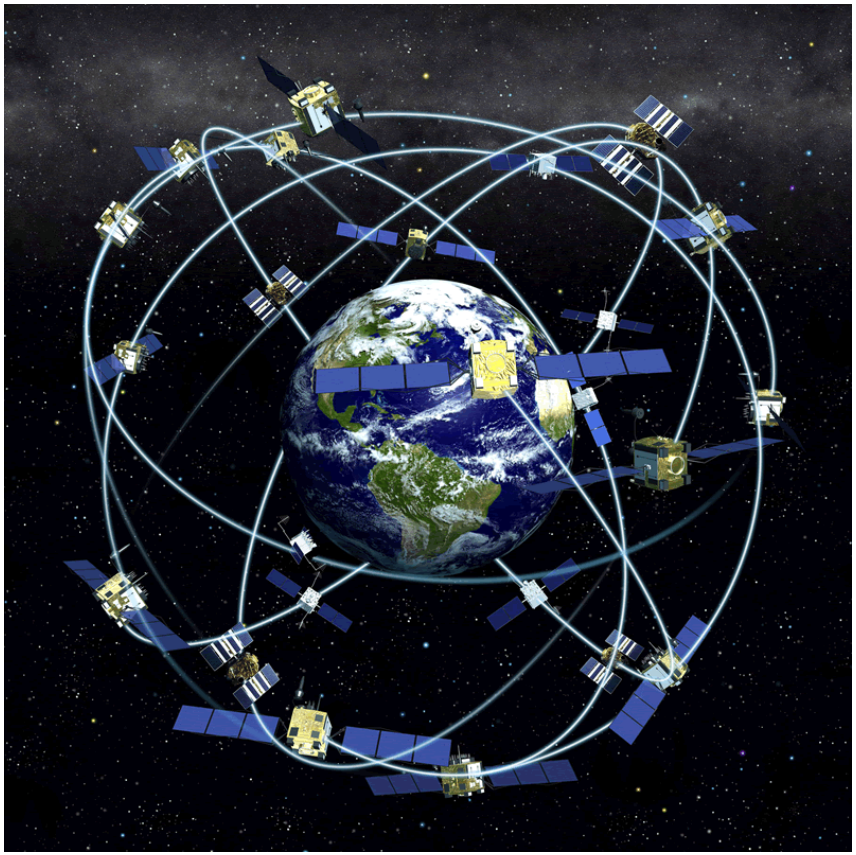
Top Design Selection

	DECISION MATRIX						
		Solution 1 GPS Guided System		Solution 2 Triangulation System		Solution 3 Electric Compass System	
Selection Criteria	Weight	Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
Ease of Use	35	7	2.04	6	1.75	4	1.17
Accuracy	35	8	2.33	6	1.75	5	1.46
Safety	30	8	2.00	6	1.50	6	1.50
Price	20	8	1.33	7	1.17	7	1.17
Total Score			**7.7**		**6.2**		**5.3**
Rank			**1**		**2**		**3**

- Design choice is Solution 1 using the GPS Guided System

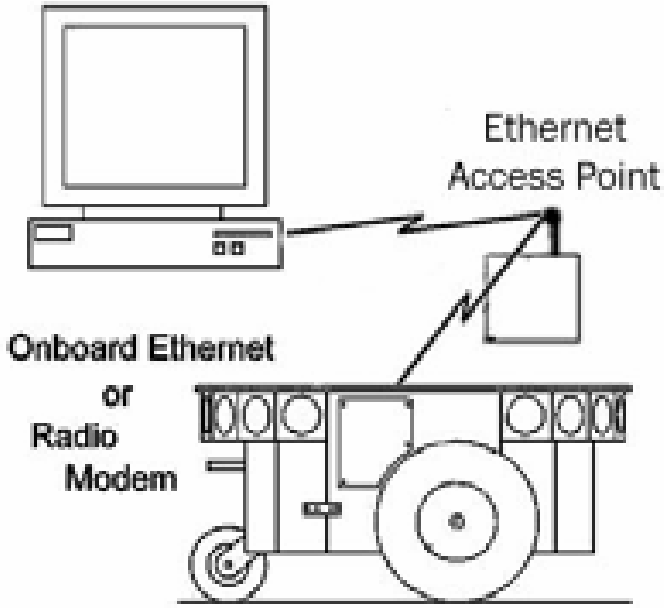
Final Solution Product

- Contains a computer, and two microcontrollers.
- One microcontroller in charge of receiving, interpreting GPS signal
- The computer will be the core part of the system that will interpret, plan, translate given instructions
- The final microcontroller will be in charge of the motors and will move the vehicle.

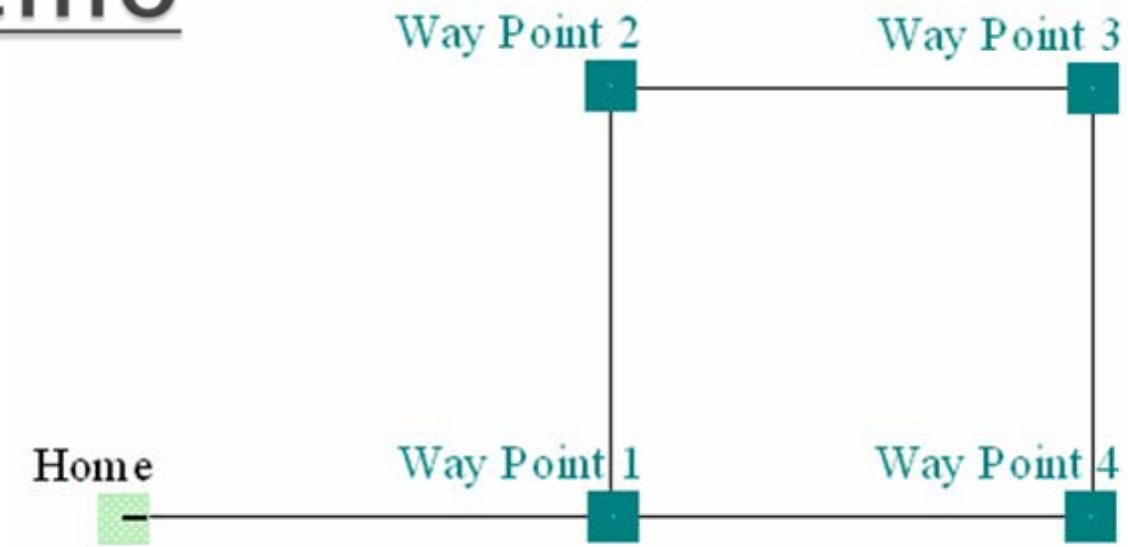


Demo Prototype Product

- Design Solution uses GPS to help vehicle navigate autonomously
- Prototype design selected will utilize the P3-AT robot as a vehicle



Active Demo



- We will communicate with the P3 via wireless router
- The robot will follow the line moving to way point 1 on through the rest of the way points and finally return home autonomously.
- The distances between points in the demo are 3 meters.

Progress: P3-AT Vehicle

Highlights and Results

- A lot of progress has been made in learning the ARIA API
- While coding these actual subroutines, we've also been able to implement a program that drives the vehicle autonomously over a fixed distance
- The Top Level code has been completed allowing us to view the sequence of functions which will be required to achieve the autonomous navigation

Lowlights and Issues

- Implementing the other behaviors has taken a little longer than expected. The behaviors will require proficiency in C++ language data structures. As a result, the team has intensified individual efforts to develop better programming skills

Progress: GPS/Microprocessor

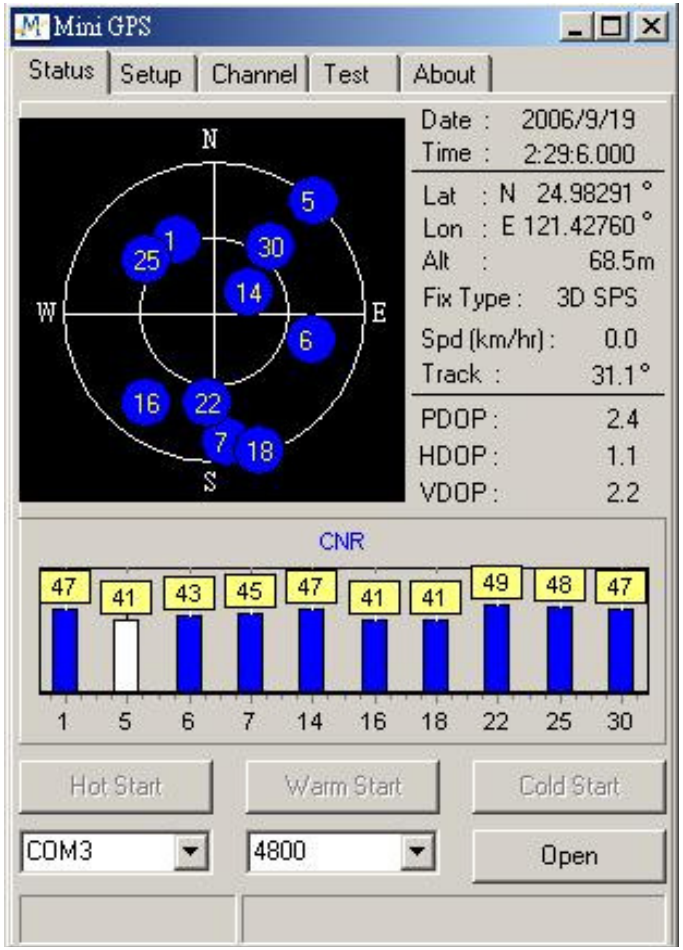
Highlights and Results

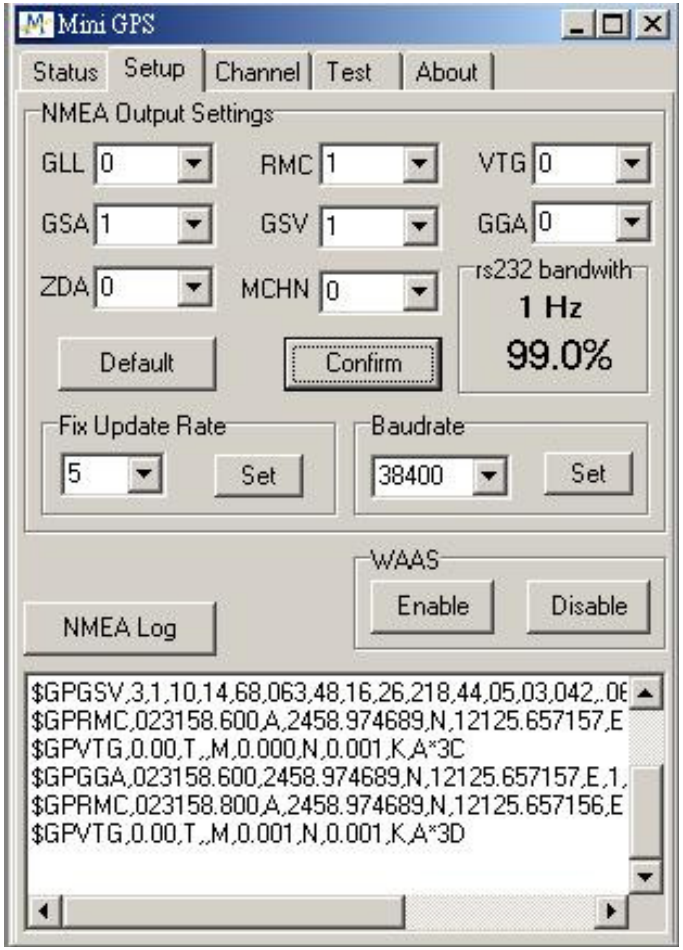
- We received full funding for the required parts and placed the orders
- Progress has been made in learning the C language needed for coding the board

Lowlights and Issues

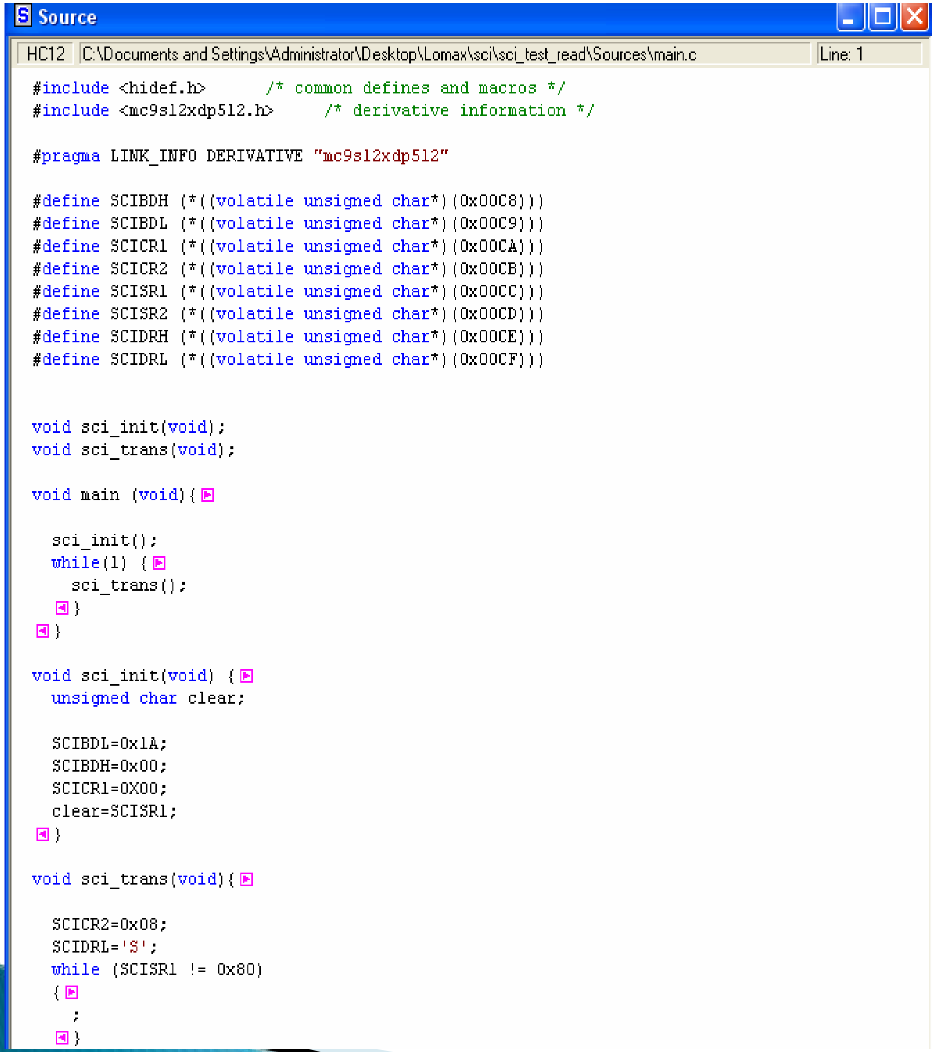
- Communication link for interfacing with the vehicle has not been finalized. This is due to some latency in receiving the GPS signal through the serial port

Progress: GPS/Microprocessor

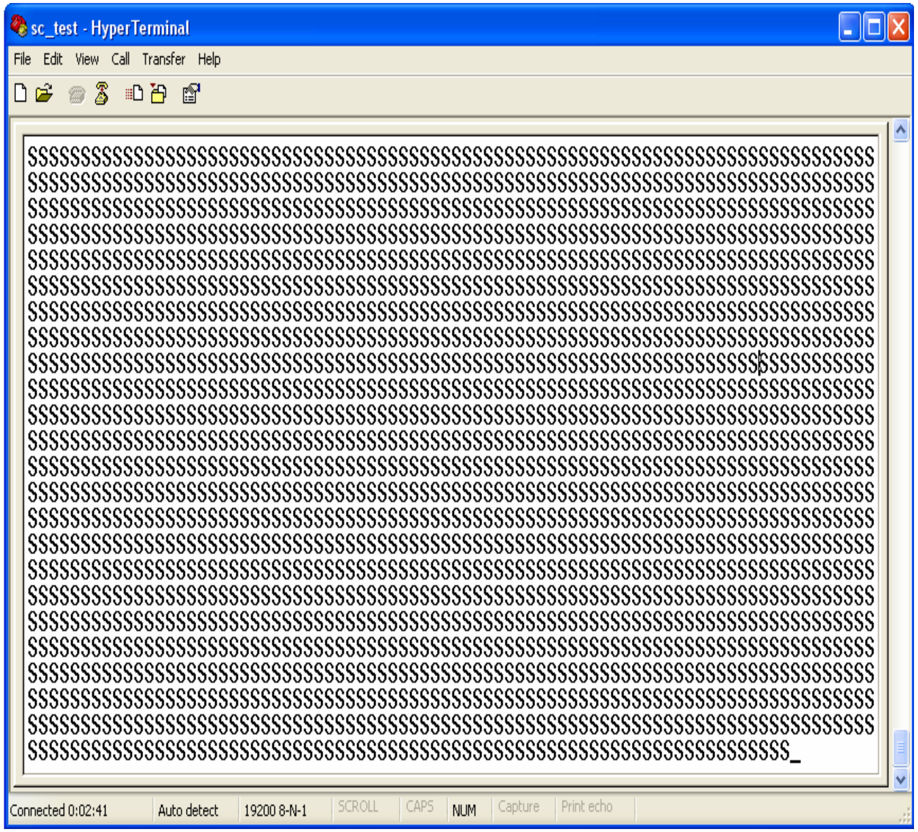




Progress: GPS/Microprocessor



```c
#include <hidef.h>      /* common defines and macros */
#include <mc9s12xdp512.h>     /* derivative information */

#pragma LINK_INFO DERIVATIVE "mc9s12xdp512"

#define SCIBDH (*((volatile unsigned char*)(0x00C8)))
#define SCIBDL (*((volatile unsigned char*)(0x00C9)))
#define SCICR1 (*((volatile unsigned char*)(0x00CA)))
#define SCICR2 (*((volatile unsigned char*)(0x00CB)))
#define SCISR1 (*((volatile unsigned char*)(0x00CC)))
#define SCISR2 (*((volatile unsigned char*)(0x00CD)))
#define SCIDRH (*((volatile unsigned char*)(0x00CE)))
#define SCIDRL (*((volatile unsigned char*)(0x00CF)))


void sci_init(void);
void sci_trans(void);

void main (void){

  sci_init();
  while(1) {
    sci_trans();
  }
}

void sci_init(void) {
  unsigned char clear;

  SCIBDL=0x1A;
  SCIBDH=0x00;
  SCICR1=0X00;
  clear=SCISR1;
}

void sci_trans(void){

  SCICR2=0x08;
  SCIDRL='S';
  while (SCISR1 != 0x80)
  {
    ;
  }
```



```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS_
```

Milestones vs. Outcomes

	MILESTONES	OUTCOMES
1.	Order and Receive Parts	DONE
2.	Learn ARIA API for programming the robot	IN PROGRESS: Still Learning base libraries. Starting to understand code, and how robot is controlled through C++. Learning to create to our own file executable files.
3.	Learn C programming for HCS12X board	IN PROGRESS: Learning to write our own libraries as well as utilizing existing libraries
4.	Learn NMEA Standards for GPS signals	DONE
5.	Send and receive information over RS-232	IN PROGRESS-Learning interrupt function of RS-232 serial connection. Having Latency Problems over RS-232.
6.	Code behaviors • DistanceCalc • Navigate_Route • Notification_Arrival • Destination_Select • Traffic_Light • Top Level Code	IN PROGRESS: Top level code has been completed and other behavior functions are proceeding as scheduled.
7.	Create a voltage step down unit and find a source to power GPS module remotely	IN Progress: In order to make robot completely autonomous we need a (battery)source to power unit PENDING: We need a driver circuit to regulate voltage levels to protect and provide power to our modules.
8.	Interface with Robot	PENDING

Identified Risks and Control

- *Risk that the ARIA API will turn out to be very complex to learn*
- Control: The team working on the ARIA API will review C++ textbooks and also consult with faculty in the Computer Science department on the best approach for working with the API.

- *There is a risk that the prototype will not function as desired in meeting all the functional requirements of the design solution*
- Control: All functional requirements will be reviewed to ensure that the programming and communication links are fast enough to meet all performance requirements.

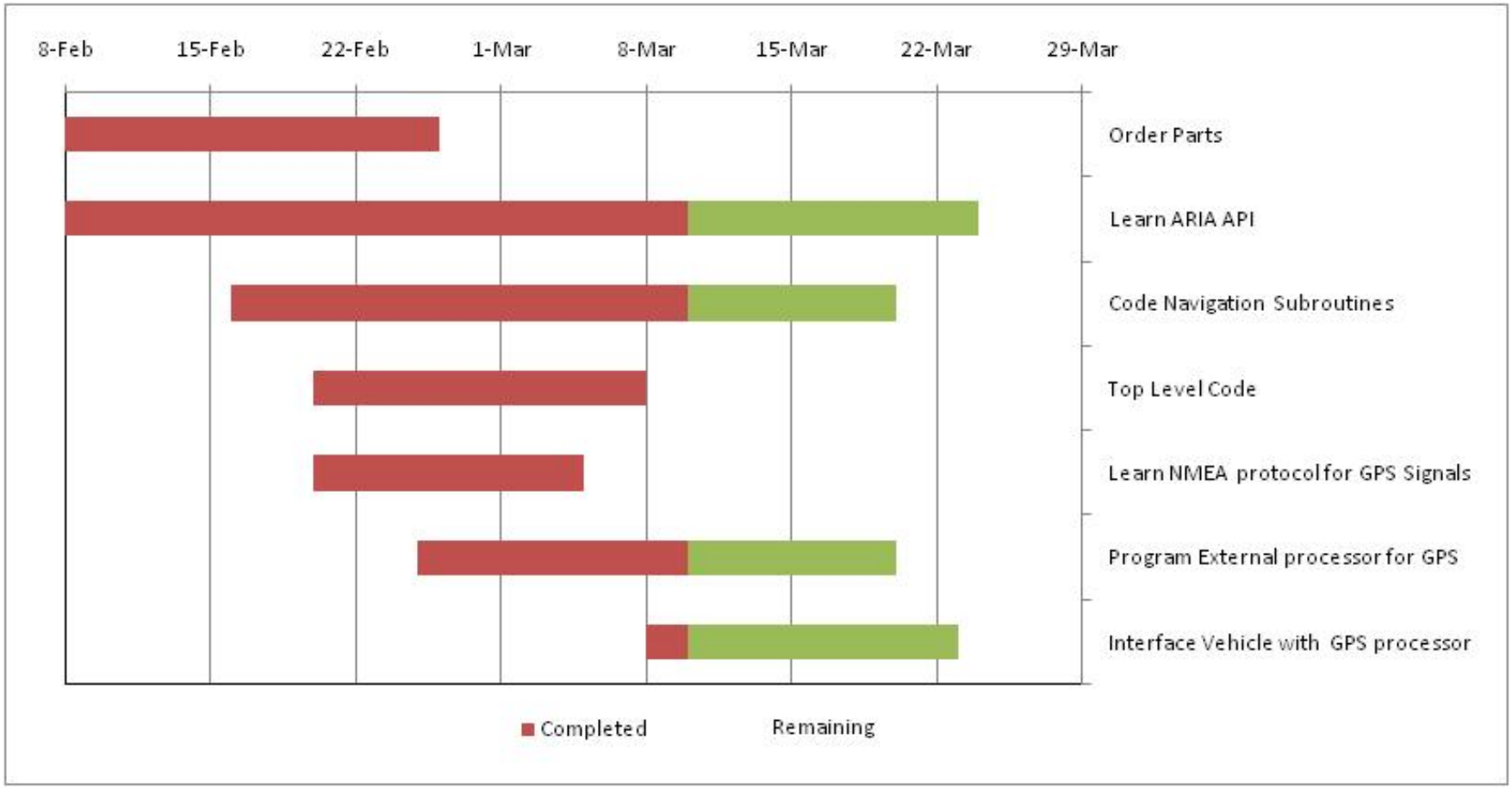
- *There is a risk of compatibility issues between the external microprocessor and the P3-AT system*
- Control: The microprocessor chosen will be based on available research on its applications and also the available connections for it to interface with the P3-AT vehicle.

- *There is a risk of overspecialization as the project has been split into various modules with different teams*
- Control: Regular meetings are held to update every team member on progress in the various modules

Future Plans

- Continue learning ARIA API and C++ language using online tutorials and C++ textbooks
- Complete subroutines for P3-AT vehicle to accurately select and navigate route
- Develop C++ algorithm to accurately read data being received from the external processor
- Configure communication link between host (P3-AT) and device (GPS module) to allow for adequate date transfer
- Timeline will be closely monitored to ensure that the project continues on schedule

Timeline


8-Feb
15-Feb
22-Feb
1-Mar
8-Mar
15-Mar
22-Mar
29-Mar
Order Parts
Learn ARIA API
Code Navigation Subroutines
Top Level Code
Learn NMEA protocol for GPS Signals
Program External processor for GPS
Interface Vehicle with GPS processor
Completed
Remaining

Review

- Introduction
- Problem Formulation
- Solution Generation
- Top Design Selection
- Final Solution Product
- Demo Prototype Product
- Progress
- Milestones vs. Outcomes
- Identified Risks and Control
- Future Plans
- Timeline
- Questions?