Vehicle Intersection Control

MCMASTER UNIVERSITY

Development Process and Implementation

SE 4G06

GROUP 6

Alex Jackson
Jean Lucas Ferreira
Justin Kapinski
Matthew Hobers
Radhika Sharma
Zachary Bazen

Contents

List of Tables

1 Revisions

Date	Revision Number	Authors	Comments
October 22, 2016	Revision 0	Alex Jackson Jean Lucas Ferreira Justin Kapinski Matthew Hobers Radhika Sharma Zachary Bazen	N/A

Table 1: VIC Table of Revisions

2 Overall Process Workflow

2.1 Project Steps and Order

Step	Task	Deadline
0	Develop high level project requirements and potential parts list	Week of October 25
1	Acquire two, one tenth ($\frac{1}{10}$) scale RC cars	Week of November 1
2	Acquire hardware to automate cars - micro controllers, sensor(s), camera(s) and mounting hardware	Week of November 1
3	Develop a track plan	Week of November 1
4	Integrate car hardware, micro-controllers and computer	Week of November 15
5	Look for open source lane following and obstacle detection	Week of November 15
6	Lane following and obstacle detection integrated and tested	Week of November 28
7	Intersection and Detection Algorithms Developed Concurrently	End of December
8	Algorithm simulations	End of January
9	Integrate decision algorithms and intersection detection with vehicles	February
10	Algorithm Testing and Refinement	March
11	Final Implementation and Documentation	Beginning of April

Table 2: VIC Project Steps

Note: Appropriate documentation will be developed as appropriate with each step.

2.2 Step Inputs and Outputs

Step	Input	Output
0	N/A	High-level Design Document
1	-	-
2	-	-
3	-	-
4	Hardware and Micro-controllers	Integrated hardware/software system
5	-	-

6	-	-
7	-	-
8	-	-
9	-	-
10	-	-
11	-	-

Table 3: VIC Project Steps Input and Output

2.3 Step Output Acceptance Criterion

Step	Output Acceptance Criterion
0	-
1	-
2	-
3	-
4	Cars are able to be controlled from central control and micro controllers
5	-
6	-
7	-
8	-
9	-
10	-
11	-

Table 4: VIC Project Steps Output Acceptance Criterion

3 Step Completion Information

3.1 Tools and Versions

TBD

3.2 Tool Setting and Use

TBD

3.3 Standards

Volere and IEEE for software requirements specification
Coding standards and conventions, for the programming language at hand.

3.4 Work Assignments

Ideally we should create two subgroups (HW and SW), but we would still discuss both aspects as a whole group.
When it comes to implementation subgroups might be more efficient.

Step	Task	Assignment
0	Develop high level project requirements and potential parts list	VIC
1	Acquire two, one tenth ($\frac{1}{10}$) scale RC cars	Name
2	Acquire hardware to automate cars - micro controllers, sensor(s), camera(s) and mounting hardware	Name
3	Develop a track plan	VIC
4	Integrate car hardware, micro-controllers and computer	HD
5	Look for open source lane following and obstacle detection	HD
6	Lane following and obstacle detection integrated and tested	SE
7	Intersection and Detection Algorithms Developed Concurrently	VIC
8	Algorithm simulations	SE
9	Integrate decision algorithms and intersection detection with vehicles	VIC
10	Algorithm Testing and Refinement	SE
11	Final Implementation and Documentation	VIC

Table 5: VIC Project Assignments

Table Key
SE: Software Team (Name, Name Name)
HD: Hardware Team (Name, Name, Name)
VIC: Whole VIC Team

Note: These assignments are tentative and subject to change as project advances.

4 Version Control Information

The version control of choice for this project is GitHub. Two repositories will likely be required: one for documentation and miscellaneous information and another for source code, libraries and dependencies.

5 Project Evolution

5.1 Bug and Change Tracking

Any issues with the project (i.e. bugs) will be posted on GitHub via the Issues panel. When a issue is posted, the appropriate members will take responsibility to fixing the bug. Once fixed, the issue will be closed. Appropriate members will include: developers, software team, hardware team or both.

5.2 Project Change Documentation

VIC will log project changes through GitHub version control logs, personal log books and VIC documentation.

5.3 Project Change Classification

VIC will classify changes in the following ways: Global change, Software change and Hardware change.

A global change constitutes a change that affects both hardware and software aspects of the system. This type of change would fundamentally alter the functionality of the system.

A software change only affects the software aspects of the system.

A hardware change only affects the hardware aspects of the system.

5.4 Making Project Change Decisions

Change Type	Change Severity	Decision Assignment	Change Documentation
Global	Software	VIC Team	Logged in VIC documents
	Hardware	VIC Team	Logged in VIC documents
	Other	VIC Team	Logged in VIC documents
Software	Local	Developer	GitHub Commit Log
	API Change	Software Team	Logged in VIC documents
Hardware	Local	Developer	Log Books
	Interface Change	Hardware Team	Logged in VIC documents