



# Vehicle Intersection Control

McMASTER UNIVERSITY

Draft Component Design

SE 4G06

GROUP 6

Alex Jackson	-	1302526
Jean Lucas Ferreira	-	1152120
Justin Kapinski	-	1305257
Mathew Hober	-	1228607
Radhika Sharma	-	1150430
Zachary Bazen	-	1200979

# Table of Contents

<b>1</b>	<b>Revisions</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Document Purpose . . . . .	4
<b>3</b>	<b>Module Guide</b>	<b>4</b>
3.1	Module Overview . . . . .	4
3.2	Intersection Software Module Description . . . . .	5
3.3	Vehicle Software Module Description . . . . .	6
3.4	Vehicle Hardware Component Description . . . . .	7
<b>4</b>	<b>Module Specifications</b>	<b>7</b>
4.1	Intersection Module Interface Specification . . . . .	7
4.2	Intersection Module Internal Design . . . . .	8
4.3	Vehicle Module Interface Specifications . . . . .	9
4.4	Vehicle Module Internal Design . . . . .	10
4.5	Vehicle Hardware Modules . . . . .	11
<b>5</b>	<b>Scheduling</b>	<b>11</b>
<b>6</b>	<b>Data Dictionary (if necessary)</b>	<b>11</b>
<b>7</b>	<b>References</b>	<b>11</b>

## List of Tables

1	VIC Table of Revisions . . . . .	3
11	ICM.1 DecisionMaker . . . . .	7
12	ICM.2 VehicleDetection . . . . .	7
13	ICM.3 Communication . . . . .	7
14	ICM.4 IC_Main . . . . .	8
21	VCM.6 ImageProcessing . . . . .	9
22	VCM.7 VehicleNavigation . . . . .	9
23	VCM.8 Communication . . . . .	9
24	VCM.9 VC_Main . . . . .	10

## List of Figures

# 1 Revisions

Table 1: VIC Table of Revisions

Date	Revision Number	Authors	Comments
January 23, 2017	Revision 0	Alex Jackson Jean Lucas Ferreira Justin Kapinski Mathew Hober Radhika Sharma Zachary Bazen	N/A

**Possible Headings - Taken from suggested content. May need additional headings**

## 2 Introduction

### 2.1 Document Purpose

Insert Text Here.

#### 2.1.1 System Scope

Insert Text Here.

#### 2.1.2 Document Overview and Intended Audience

Insert Text Here.

#### 2.1.3 Acronyms

Insert Text Here.

#### 2.1.4 Definitions

Insert Text Here.

## 3 Module Guide

### 3.1 Module Overview

#### 3.1.1 Intersection Controller Modules

ID	Name	Responsibilities	Secrets
ICM.1	DecisionMaker	Determine order of car progression	Scheduling algorithm
ICM.2	VehicleDetection	Know when a car is on top of one of the intersection sensors, and the corresponding sensor	Relationship between magnetic sensor and car
ICM.3	Communication	Interpret receiving car signals and sending signals to a car	Communication protocol
ICM.4	IC_Main	Control information flow of intersection controller	Manages intersection modules

#### 3.1.2 Vehicle Controller Hardware Modules

ID	Name	Responsibilities	Secrets
VCM.1	SignalConverter	Convert a software signals to a physical signal, and vice versa	How to convert signal
VCM.2	SpeedConverter	Convert wheel rotation count to a speed value	Speed calculation algorithm

VCM.3	ServoController	Set a physical wheel angle	How to convert a software value to a PWM (Pulse Width Modulation) signal
VCM.4	MotorSpeedController	Control PWM signal	How to convert speed into a PWM signal
VCM.5	MotorHBridge Controller	Setting H bridge gates	Which gates correspond to which action of the motor

### 3.1.3 Vehicle Controller Software Modules

ID	Name	Responsibilities	Secrets
VCM.6	ImageProcessing	Interpret image into environment state	Image processing algorithm
VCM.7	VehicleNavigation	Control the navigation of the car	How the car navigates on the track
VCM.8	Communication	Interpret signal from Intersection Controller. Prepare and send signal to the Intersection Controller	Communication Protocol
VCM.9	VC_Main	Control information flow of the car	Manage car modules

## 3.2 Intersection Software Module Description

### Design Notes

<Description goes here for all the intersection software modules>

<ID and Name 1>

### Behavioural Description

<Description goes here.>

### Inputs

<List one after the other>

### Outputs

<List one after the other>

### Initialization Description

<Describe here>

### Derived Timing Constraints

<Describe here>

<ID and Name 2>

### Behavioural Description

<Description goes here.>

**Inputs**

&lt;List one after the other&gt;

**Outputs**

&lt;List one after the other&gt;

**Initialization Description**

&lt;Describe here&gt;

**Derived Timing Constraints**

&lt;Describe here&gt;

### 3.3 Vehicle Software Module Description

**Design Notes**

&lt;Description goes here for all the intersection software modules&gt;

&lt;ID and Name 1&gt;

**Behavioural Description**

&lt;Description goes here.&gt;

**Inputs**

&lt;List one after the other&gt;

**Outputs**

&lt;List one after the other&gt;

**Initialization Description**

&lt;Describe here&gt;

**Derived Timing Constraints**

&lt;Describe here&gt;

&lt;ID and Name 2&gt;

**Behavioural Description**

&lt;Description goes here.&gt;

**Inputs**

&lt;List one after the other&gt;

**Outputs**

&lt;List one after the other&gt;

**Initialization Description**

&lt;Describe here&gt;

**Derived Timing Constraints**

&lt;Describe here&gt;

### 3.4 Vehicle Hardware Component Description

#### Design Notes choices

<Description goes here for all the intersection software modules>

<ID and Name 1>

#### Inputs

<List one after the other>

#### Outputs

<List one after the other>

<ID and Name 2>

#### Inputs

<List one after the other>

#### Outputs

<List one after the other>

## 4 Module Specifications

There are two main components to VIC: the intersection component and the vehicle component. The following module specifications are grouped in this way.

### 4.1 Intersection Module Interface Specification

These may be subject to change

Table 11: ICM.1 DecisionMaker

ICM.1 DecisionMaker	
DecisionMaker()	Constructor to initialize the scheduling algorithm
getSchedule(cars[ ]) : carQueue	When function is called, it will return a queue of cars in the order which they should proceed. Expects an array of car objects when called

Table 12: ICM.2 VehicleDetection

ICM.2 VehicleDetection	
VehicleDet()	Constructor to initialize the detection of vehicles at the intersection.
getSignalsState( ) : bool[ ]	Returns the state of the sensors at the intersection when the function is called. Returns an array of boolean values signifying if the sensors have been tripped or not.

Table 13: ICM.3 Communication

ICM.3 Communication	
RecieveRequest() : Request	Function to allow the controller recieve a request to be scheduled from the car.
SendResponse(car c) : void	Function that allows the intersection to send a car the response to proceed through the intersection.

Table 14: ICM.4 IC\_Main

ICM.4 IC_Main	
Main( )	Main Function for VIC.

## 4.2 Intersection Module Internal Design

double check whats included in the MID'S

<ID and Name 1>

<Description goes here.>

### Variables

<Variable 1 >	<Description>
<Variable 2>	<Description>
<Variable 3>	<Description>
<Variable 4>	<Description>

### Objects

<Object 1 >	<Description>
<Object 2>	<Description>
<Object 3>	<Description>
<Object 4>	<Description>

### Methods

<Method 1 >	<Description, parameters and return>
<Method 2>	<Description, parameters and return>
<Method 3>	<Description, parameters and return>
<Method 4>	<Description, parameters and return>

<ID and Name 2>

<Description goes here.>

### Variables



<Variable 1 >	<Description>
<Variable 2>	<Description>
<Variable 3>	<Description>
<Variable 4>	<Description>

## Objects

<Object 1 >	<Description>
<Object 2>	<Description>
<Object 3>	<Description>
<Object 4>	<Description>

## Methods

<Method 1 >	<Description, parameters and return>
<Method 2>	<Description, parameters and return>
<Method 3>	<Description, parameters and return>
<Method 4>	<Description, parameters and return>

## 4.3 Vehicle Module Interface Specifications

Table 21: VCM.6 ImageProcessing

VCM.6 ImageProcessing	
ImgProc( )	Function to capture images of the track environment from a webcam and process it into information that can be analysed by software.
getImageInfo( ) : ADT	Function to relay image information when called.

Table 22: VCM.7 VehicleNavigation

VCM.7 VehicleNavigation	
VehicleNav( )	Function to signal to the vehicle if there is a change in the navigation, and if so, what changes should be made.
GetCarState( ) : enum	Function to relay the car state. Will return the states as an enum. Exact states will be determined later.
driveThroughIntersection( ) : void	Function to signal the car to proceed through the intersection.

Table 23: VCM.8 Communication

VCM.8 Communication
---------------------

SendRequest(Request r) : void	Function to allow the car to send a request to the intersection controller.
Recieve Response( ) : Car	Function to allow the vehicle to revice a response to proceed from the intersection controller.

Table 24: VCM.9 VC\_Main

<b>VCM.9 VC_Main</b>	
VC_Main	Function to control all software aspects of the vehicle control.

## 4.4 Vehicle Module Internal Design

double check whats included in the MID'S

<ID and Name 1>

<Description goes here.>

### Variables

<Variable 1 >	<Description>
<Variable 2>	<Description>
<Variable 3>	<Description>
<Variable 4>	<Description>

### Objects

<Object 1 >	<Description>
<Object 2>	<Description>
<Object 3>	<Description>
<Object 4>	<Description>

### Methods

<Method 1 >	<Description, parameters and return>
<Method 2>	<Description, parameters and return>
<Method 3>	<Description, parameters and return>
<Method 4>	<Description, parameters and return>

<ID and Name 2>

<Description goes here.>

### Variables

<Variable 1 >	<Description>
<Variable 2>	<Description>
<Variable 3>	<Description>
<Variable 4>	<Description>

### Objects

<Object 1 >	<Description>
<Object 2>	<Description>
<Object 3>	<Description>
<Object 4>	<Description>

### Methods

<Method 1 >	<Description, parameters and return>
<Method 2>	<Description, parameters and return>
<Method 3>	<Description, parameters and return>
<Method 4>	<Description, parameters and return>

## 4.5 Vehicle Hardware Modules

All things hardware related. We didn't do "MIS" for the hardware yet. Note the id numbers may get you confused but they follow the module guide

## 5 Scheduling

Unsure what this is"

## 6 Data Dictionary (if necessary)

Insert Text Here.

## 7 References

Insert Text Here.