Vehicle Intersection Control

McMaster University

Draft Component Design

SE 4G06

Group 6

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

Table of Contents

List of Tables

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 Hobers Radhika Sharma Zachary Bazen	N/A

Possible Headings - Taken from suggested content. May need addtional 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	TrafficController	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	PWM	Convert a software signals to a physical PWM signal	How to convert a software value to PWM
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

3.1.3 Vehicle Controller Software Modules

ID	Name	Responsibilities	Secrets
VCM.6	ImageProcessing	Interpret image into environment state	Image processing algorithm
VCM.7	VehicleNavigaton	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
The Vluetooth communication implementation

ICM.1 - TrafficController

Behavioural Description
Will receive input from IC_ Main only when IC_ Main have input from Communication and Vehicle Detection. Traffic Controller will then assess the input and decide if the vehicle may proceed or not. If more than one car is approaching the intersection, Traffic Controller will notify the vehicles whether they may proceed or not based on the intersection conditions.

Inputs The module will get information about the current intersection state, and the list of cars dictating which direction they are coming from, and where they are going.

Outputs
TrafficController will output whether or not the vehicle can proceed

Initialization Description
TrafficController will be initialized once, when the IC_Main has been initialized. It does not require any specific initialization procedure.

Derived Timing Constraints
The TrafficController will have a soft deadline of 1 second, however a hard deadline will not be enforced.

ICM.2 - VehicleDetection

Behavioural Description
Vehicle Detection will make use of a camera to view the intersection, from a bird's eye view. It will create

and update a data structure which will hold the state of the intersection.

Inputs
This module will have no inputs from other modules.

Outputs
VehicleDetection will output a data structure containing the state of the intersection.

Initialization Description
IC_Main will initialize this module. Upon initialization, VehicleDetection will turn on the web camera.

Derived Timing Constraints
The car will be travelling at maximum 1.4 m/sThe length of the intersection is about 0.6m. We want to detect the car at every 1/6 of the intersection. Therefore, the VehicleDetection must be able to update within the time constraint setup by IC_Main.

ICM.3 - Communications

Behavioural Description It will be responsible for the two-way communication from the car to the intersection controller. It will create a list of communications coming in and coming out.

Inputs
A Bluetooth signal containing information from the requesting vehicle. Such as the car the orientation and destination of the car, and communication information.

Outputs
The module will have two outputs. One will the information of inbound cars to the IC_Main, the other will be a response to the cars that it can proceed or not.

Initialization Description
The inbound and outbound list of cars will be initialized empty, and the module will remain running continuously.

Derived Timing Constraints
Limited to the speed of the Bluetooth connection.

ICM.4 - IC_Main

Behavioural Description
It will be responsible for controlling the flow of information between all components of the intersection controller. Also maintains a list of the current traffic state, and updated accordingly.

Inputs
Inbound communication from the Communications module, the current state of the intersection from the VehicleDetection module, and the response from the TrafficController.

Outputs
The output will be cars going into the outbound list of cars that may proceed through the intersection.

Initialization Description
Ensure the list of active cars in the intersection is cleared. It also initializes all other modules connected to it.

Derived Timing Constraints

The timing constraint is based on the speed of the car, and the length of the intersection. In order to detect when a car has entered and left the intersection, we must process the current intersection state within this given time.

$$processingTime = \frac{intersectionSegmentLength}{carSpeed}$$

$$processingTime = \frac{\frac{1}{6} * 0.6m}{1.4m/s}$$

$$processingTime = 0.07seconds$$

3.3 Vehicle Software Module Description

VCM.6 - ImageProcessing

Behavioural Description The image processing module of the vehicle controller is responsible for detecting lane positioning and obstacles. Once these factors have been detected it must produce digital information pertaining to the state of the track as seen in the current image. It must then pass this information on to the vehicle navigation module which will make appropriate adjustments.

Inputs
A path to the image produced by the webcam installed on the car. The image will be taken just prior to calling the image processing module.

Outputs
Digitally interpreted information regarding the state of the track.

Initialization Description
The image processing module will be initialized once the car has been turned on. It does not have any specialized initialization procedure.

Derived Timing Constraints
The time constraint for the imageProcessing is bounded by the time constraint in VC_Main.

VCM.7 - VehicleNavigation

Behavioural Description The vehicle navigation module of the vehicle controller is responsible for ensuring that the vehicle follows the correct lane on the track, stops at intersections, and avoids obstacles. This module will apply any necessary adjustments in trajectory to ensure that these responsibilities are met.

Inputs
Information regarding the current state of the track.

Outputs
Adjustments to the vehicle's steering, acceleration, and braking necessary to continue following the track.

Initialization Description
The vehicle navigation module will be initialized once the vehicle has been turned on. The vehicle's steering angle, speed, and acceleration with be initialized at zero (i.e. standing still with the steering pointed directly forwards).

Derived Timing Constraints
The vehicle navigation module has negligible processing time constraints.

VCM.8 - Communication

Behavioural Description Responsible for establishing communication from the car to the intersection controller, when it begins to approach an intersection. It will also receive the response from the intersection controller if it can proceed through the intersection.

Inputs
The state of the car, such as the current orientation and intended direction. It will also receive input from the intersection controller if it is allowed to proceed through the intersection.

Outputs
One output will be the information about the car to the intersection communication module. The other will be the 'go-ahead' signal to the car controller.

Initialization Description
Will be initialized when the car is initiated, and paired to the intersection controller.

Derived Timing Constraints
Limited to the speed of the Bluetooth connection.

VCM.9 - VC_Main

Behavioural Description This module will assist the vehicle in navigating the intersection. It will work in conjunction with the imaging processing module. Further more, VC_Main will make navigation decisions about the intersection environment and instruct the car on how to proceed. VC_Main will also control the communication for the vehicle.

Inputs The inputs to this module will be the "go ahead" signal from the intersection controller as well as a struct from ImageProcessing.

Outputs This module will output the "request to go" signal to the Communication module as well as the navigation instructions to the VehicleNavigation Module.

Initialization Description Upon initialization, VC_Main will start up the VehicleNavigation module, the ImageProcessing module and the Communication Module.

Derived Timing Constraints The exact car speed is unknown, but it will be estimated to be 1.4 m/s, and we will require to get an update on the track status every 3cm to maintain accuracy. Therefore, at this speed and this update rate, the IC_Main must process all necessary information within the following time:

$$processingTime = \frac{minDistanceUpdateRate}{carSpeed}$$

$$processingTime = \frac{0.03m}{1.4m/s}$$

$$processingTime = 0.02\ seconds$$

a picture must be processed every 0.02 seconds.

3.4 Vehicle Hardware Component Description

Design Notes
The reason Hall Effect sensors were chosen to measure the speed of the vehicle is because they are simple to use and they are cheap.

VCM.1 - PWM

Inputs
The duty cycle and frequency for the PWM.

Outputs
Physical PWM signal on a GPIO pin.

VCM.2 - SpeedConverter

Inputs
Signal from Hall Effect sensor mounted next to wheel.

Outputs
The approximate speed of the vehicle.

VCM.3 - ServoController

Inputs
The desired angle for the servo in software.

Outputs
A physical signal on a GPIO pin telling the servo to go to the desired angle.

VCM.4 - MotorSpeedController

Inputs
The desired speed of the vehicle in software.

Outputs
A physical signal on a GPIO pin telling the motor to go at a specific speed.

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 17: ICM.1 - TrafficControl

ICM.1 TrafficControl

TrafficControl()	Constructor to initialize the scheduling algorithm
getTrafficPermission(Queue<cars>, intersectionState[ ]) : Boolean	When called, it will evaluate all active cars and the current intersection state, then make a decision whether the most recent car can proceed or not.

Table 18: ICM.2 VehicleDetection

ICM.2 VehicleDetection	
VehicleDetection()	Constructor to initialize the detection of vehicles at the intersection.
getIntersectionState( ) : Boolean[ ]	Returns the state of the intersection when the function is called. Returns an array of boolean values signifying where a car is positioned in the intersection.

Table 19: ICM.3 Communication

ICM.3 Communication	
RecieveRequest() : Request	Function to allow the controller recieve a request to be scheduled from the car.
0SendResponse(car c) : void	Function that allows the intersectrion to send a car the response to proceed through the intersection.

Table 20: ICM.4 IC_Main

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

4.2 Intersection Module Internal Design

ICM.1 - TrafficControl

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>
<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

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

ICM.2 - VehicleDetection

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>
<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

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

ICM.3 - Communication

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>

<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

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

ICM.4 - IC_Main

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>
<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

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

4.3 Vehicle Module Interface Specifications

Table 37: VCM.6 ImageProcessing

VCM.6 ImageProcessing	
ImageProcessing() : ImageData	Function to capture images of the track environment from a webcam and process it into information that can be analysed by software. Will return a struct called ImageData.

Table 38: VCM.7 VehicleNavigation

VCM.7 VehicleNavigation	
VehicleNavigation(struct *ImageData, struct *communicationResponse)	Function to signal to the vehicle if there is a change in the navigation, and if so, what changes should be made.

Table 39: VCM.8 Communication

VCM.8 Communication	
SendRequest(struct *request) : void	Function to allow the car to send a request to the interection controller.
RecieveResponse( ) : struct* response	Function to allow the vehicle to revice a response to proceed from the intersection controller.

Table 40: VCM.9 VC_Main

VCM.9 VC_Main	
VC_Main	Function to control all software aspects of the vehicle control.

4.4 Vehicle Module Internal Design

<ID> - <Name>

<Description goes here.>

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>
<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

<Method 1 >	<Desciption, parameters and return>
<Method 2>	<Desciption, parameters and return>
<Method 3>	<Desciption, parameters and return>

<Method 4>	<Desciption, parameters and return>

<ID> - <Name 2>

<Description goes here.>

Variables

<Variable 1 >	<Desciption>
<Variable 2>	<Desciption>
<Variable 3>	<Desciption>
<Variable 4>	<Desciption>

Objects

<Object 1 >	<Desciption>
<Object 2>	<Desciption>
<Object 3>	<Desciption>
<Object 4>	<Desciption>

Methods

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

4.5 Vehicle Hardware Module Interface Specification

Table 49: VCM.1 PWM

VCM.1 PWM	
set_PWM(pin_number, duty_cycle) : void	Used to output a PWM signal to a given GPIO pin at a given duty cycle.

Table 50: VCM.2 SpeedConverter

VCM.2 SpeedConverter	
get_speed( ) : double	Returns the current speed of the vehicle as measured by the Hall Effect sensor.

Table 51: VCM.3 ServoController

VCM.3 ServoController

set_angle(angle : double) : void	Outputs a physical signal to the servo to go to the specified angle.

Table 52: VCM.4 MotorSpeedController

VCM.4 MotorSpeedController	
set_speed(speed : double) : void	Outputs a physical signal to the motor to go at a specified speed.

5 Vehicle Hardware Module Internal Design

VCM.1 - PWM

Generate PWM signals on GPIO pins of the Raspberry PI.

Variables
None.

Methods

set_PWM(pin_number, duty_cycle) : void	Used to output a PWM signal to a given GPIO pin at a given duty cycle.

VCM.2 - SpeedConverter

Convert the physical signal from the Hall Effect sensor mounted next to the wheel into a velocity.

Variables
None.

Methods

get_speed( ) : double	Returns the current speed of the vehicle as measured by the Hall Effect sensor.

VCM.3 - ServoController

Generate physical signals that will control the steering servo of the vehicle.

Variables

current_angle : double	Stores the current direction of the servo so it can continuously send the correct angle to the servo.

Methods

set_angle(angle : double) : void	Outputs a physical signal to the servo to go to the specified angle.

VCM.4 - MotorSpeedController

Generate physical signals that will control the motor of the vehicle.

Variables
None.

Methods

set_speed(speed : double) : void	Outputs a physical signal to the motor to go at a specified speed.

6 Scheduling

Unsure what this is"

7 Data Dictionary (if necessary)

Insert Text Here.

8 References

Insert Text Here.