Vehicle Intersection Control

McMaster University

Draft System 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
December 21, 2016	Revision 0	Alex Jackson Jean Lucas Ferreira Justin Kapinski Mathew Hobers Radhika Sharma Zachary Bazen	N/A

2 Introduction

2.1 Document Purpose

The purpose of this document is to provide insight into the system design of VIC (Vehicle Intersection Controller). VIC is a system that allows autonomous cars to proceed through stop sign intersections when the vehicles arrive simultaneously.

2.2 System Scope

VIC will focus on solving the aforementioned problem on a controlled indoor track. 1/10 scale autonomous vehicles will be used to simulate real world autonomous cars. To prevent damage of hardware, the autonomous vehicles will be able to detect obstacles. VIC will ignore situations involving non-autonomous cars.

2.3 Document Overview and Intended Audience

This document will outline the module guides, the module interface specification, and the component descriptions. Furthermore, the document will provide a behaviour overview, context diagrams and system component diagrams. The intended audience for this document is Sean Marshall (the engineering team leader at GM) who proposed the problem, Dr. Alan Wassyng and the teaching assistants as supervisors of the project, and ourselves as designers of the system.

2.4 Acronyms

Table 2: Acronyms

VIC	Vehicle Intersection Control
IC	Intersection Controller
VC	Vehicle Controller

2.5 Definitions

Table 3: Definitions

VIC	The entire system including the intersection controller, the vehicles, and their corresponding controllers.
IC	The Intersection Controller is the system that tracks the arrival and departure of the vehicles, as well as determining the order in which the vehicles must proceed through the intersection.
VC	The Vehicle Controller is the system that will allow the 1/10 scale RC car to follow lanes, maintain a desired speed, steer itself, and send requests to the intersection controller.

2.5.1 Naming Conventions

Table 4: Naming Conventions

m_ic_variableName	Monitored variable for intersection controller
c_ic_variableName	Control variable for intersection controller
m_vc_variableName	Monitored variable for autonomous vehicle controller

c_vc_variableName	Control variable for autonomous vehicle controller
ICD#	Intersection Controller Design Component ID
ICM#	Intersection Controller Module Guide ID
VCD#	Vehicle Controller Design Component ID
VCM#	Vehicle Controller Module Guide ID

3 Monitored Variables

3.1 Intersection Controller

Table 5: Intersection Controller Monitored Variables

m_ic_readSensor	Boolean [8]
m_ic_carSignal	Byte [4][ ]

3.2 Autonomous Vehicle Controller

m_vc_videoCapture	Bytes [ ][ ]
m_vc_frontDistance	Double
m_vc_speedSignal	Boolean
m_vc_hallEffect	Double
m_vc_vehicleOrientation	Character

4 Controlled Variables

4.1 Intersection Controller

c_ic_carProceedSignal	Boolean

4.2 Autonomous Vehicle Controller

c_vc_wheelAngle	Double
c_vc_carSpeed	Integer
c_vc_vehicleBrake	Boolean
c_vc_requestIC	Byte[ ]

5 System Overview

5.1 Behavior Overview

5.2 Context Diagrams

Figure 1: Car Controller Context Diagram

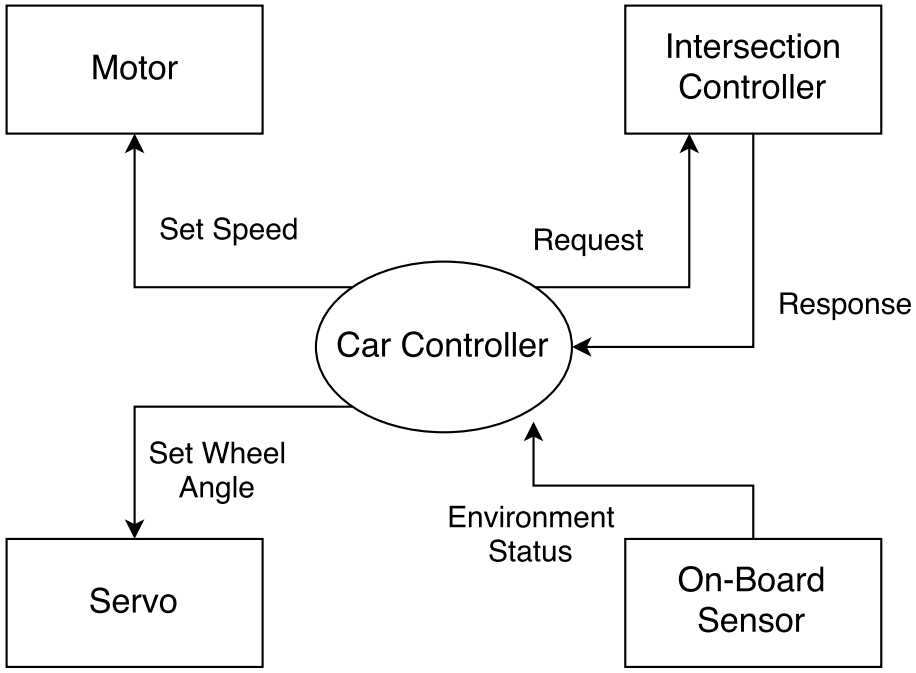
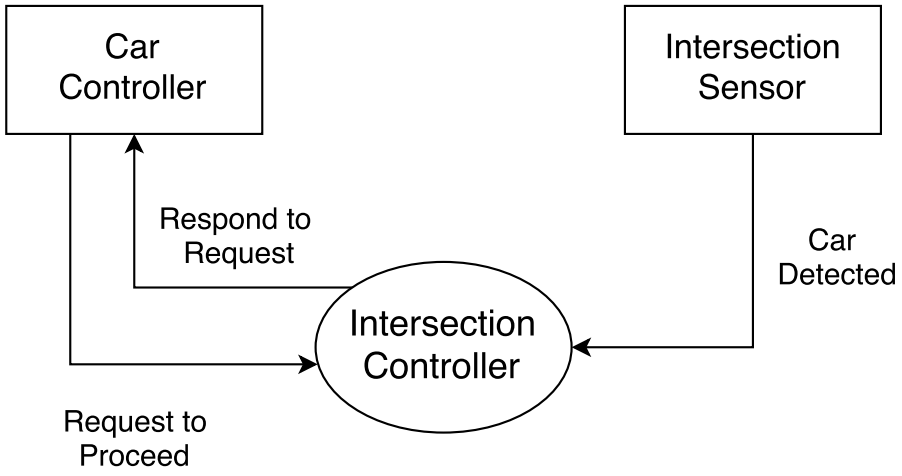


Figure 2: Intersection Controller Context Diagram



5.3 System Component Diagrams

Insert Text or Image Here.

6 System Components

6.1 Intersection Control Component

IDC1	
Description	The Intersection Controller directs traffic at an intersection by communicating with vehicles and determines which order they should proceed
Inputs	m_ic_carSignal[4]
	m_ic_readSensors[8]
Outputs	c_ic_carSignal[4]
Timing Constraints	1 second intersection arrival decision
	1 second intersection schedule
	0.5 second intersection departure decision
Deadline	Decisions must be made before the next intersection arrival poll
Initialization	Connect to autonomous vehicles over Bluetooth communication
	Clear all intersection arrival queues

6.2 Vehicle Controller Component

VCD1	
Description	A 1:10 scale RC car will be controlled by the Vehicle Controller. The vehicle will be able to follow lanes and stop at intersections. It will communicate with an Intersection Controller and proceed through the intersection after receiving the appropriate signal from it
Inputs	m_vc_videoCapture[x][y]
	m_vc_frontDistance
	m_vc_hallEffect
	m_ic_carProceedSignal
Outputs	c_vc_ wheelAngle
	c_vc_carSpeed
	c_vc_vehicleBreak
	c_vc_requestTheIC
Timing Constraints	Process images within 20 ms
Initialization	Initialize all speed controls to zero
	Initialize wheel angle to zero
	Connect to intersection over Bluetooth communication

7 Module Guide

7.1 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

7.2 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

7.3 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

8 Module Interface Specification

Table 14: 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 15: 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 16: 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 intersectrion to send a car the response to proceed through the intersection.

Table 17: ICM.4 IC_Main

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

Table 18: 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 19: 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 20: VCM.8 Communication

VCM.8 Communication	
SendRequest(Request r) : void	Function to allow the car to send a request to the interection controller.

Recieve Response( ) : Car	Function to allow the vehicle to revice a response to proceed from the intersection controller.

Table 21: VCM.9 VC_Main

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

9 References

Possible References Here