[illegible]

โดย

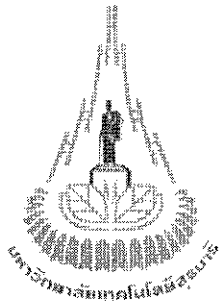
[illegible]

[illegible]

[illegible]

[illegible]

CONTRIBUTION



ระบบ A.I. (Artificial Intelligence) เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบไฟจราจร

โดย

นายนพดล	ลัยนันท์	B4503552
นายธัชกร	ไกรธนานันต์	B4503644
นายพงศกร	คงสมบูรณ์	B4505983

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 427494 โครงงานศึกษาวิศวกรรมโทรคมนาคม และ 427499 โครงงานวิศวกรรมโทรคมนาคม

ประจำภาคการศึกษาที่ 1 และ 3 ปีการศึกษา 2548

หลักสูตรวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุงพ.ศ. 2541

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

หัวข้อโครงงาน	ระบบปัญญาประดิษฐ์ (*A.I.* หรือ *Artificial Intelligence*) เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบสัญญาณไฟจราจร	
นักศึกษา	นาย นพดล ลัยนันท์	รหัส B4503552
	นาย ธัชกร ไกรธนานันต์	รหัส B4503644
	นาย พงศกร คงสมบูรณ์	รหัส B4505983
ปริญญา	วิศวกรรมศาสตร์บัณฑิต	
พ.ศ.	2548	
อาจารย์ที่ปรึกษาโครงงาน	อ.ดร.วิภาวี อุสาหะ	

บทคัดย่อ

โครงงานฉบับนี้เป็นการนำระบบปัญญาประดิษฐ์*(Artificial Intelligence หรือ A.I.)*มาเพื่อใช้ในการวิเคราะห์และการแก้ไขปัญหาระบบสัญญาณไฟจราจร เนื่องจากระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา(Fixed Time) ที่ใช้อย่างแพร่หลายอยู่ในปัจจุบันนั้น ไม่สามารถให้อัตราผ่านของรถต่อช่วงเวลาในแต่ละเส้นทางได้ดีเพียงพอสำหรับทุกช่วงเวลา จึงทำให้ไม่สามารถใช้ประโยชน์จากสัญญาณไฟจราจรได้อย่างเต็มประสิทธิภาพ ดังนั้นคณะผู้จัดทำจึงได้ทำโครงงานระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ ขึ้นมาเพื่อใช้ในการแก้ไขปัญหาดังกล่าว จากการเขียนโปรแกรมภาษา *C++* และในโครงงานนี้ได้ทำการเลือกใช้วิธี *Q-Learning* ซึ่งเป็นส่วนหนึ่งของเทคนิคที่เรียกว่า การเรียนรู้แบบรีอินฟอร์สเมนท์*(Reinforcement Learning* หรือ *RL)* โดยที่วิธีการดังกล่าวนั้นจะทำให้คอมพิวเตอร์สามารถทำการเรียนรู้ได้ด้วยตัวเอง จากการกำหนดสิ่งที่เราสนใจและทำการเปลี่ยนให้อยู่รูปแบบของข้อมูลที่โปรแกรมสามารถทำการวิเคราะห์และคำนวณได้ ซึ่งในที่นี้สิ่งที่เราสนใจคือ ปริมาณของยานพาหนะบนท้องถนน แต่ในการทดสอบผลการทดลองเนื่องจากข้อจำกัดในการทดลอง จึงได้ทำการใช้รถจักรยานยนต์ในการทำการทดสอบผลแทน โดยการใช้เซ็นเซอร์แบบอินฟาเรดทั้งภาครับและภาคส่งในการตรวจนับปริมาณรถที่มีอยู่ ในแต่ละเส้นทาง ณ ช่วงเวลานั้น แล้วจึงนำข้อมูลดังกล่าวไปทำตามขั้นตอนและกระบวนการตามเงื่อนไขของวิธี *Q-Learning* ด้วยสมการทางคณิตศาสตร์ เพื่อทำการตัดสินใจที่ถูกต้องต่อการเลือกเปลี่ยนแปลงสัญญาณไฟจราจรผ่านตัวไมโครโพรเซสเซอร์เพื่อให้ได้อัตราผ่านที่ดีที่สุด จึงสามารถทำการเปลี่ยนแปลงสัญญาณไฟจราจรให้สอดคล้องกับสภาพการจราจรที่เป็นอยู่ในขณะนั้นได้ดียิ่งขึ้น และทำการเปรียบเทียบประสิทธิภาพที่ได้จากอัตราผ่านของรถ เปรียบเทียบกับระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา และทำการเปรียบเทียบกับการใช้มนุษย์ในการตัดสินใจ เพื่อแสดงให้เห็นถึงประสิทธิภาพในการนำระบบปัญญาประดิษฐ์มาใช้ในการควบคุมระบบสัญญาณไฟจราจรได้อย่างชัดเจนมากยิ่งขึ้น ซึ่งจากผลการทดลองโครงงานพบว่าการใช้ระบบปัญญาประดิษฐ์การควบคุมสัญญาณไฟจราจรนั้นสามารถทำงานได้ดีใกล้เคียงกับมนุษย์ และมีประสิทธิภาพดีกว่าระบบควบคุมสัญญาณไฟจรจาแบบตั้งเวลาที่ใช้อยู่ในปัจจุบัน

กิตติกรรมประกาศ

คุณงามความดีอันใดที่เกิดจากโครงงานฉบับนี้ ขอมอบแด่บิดา มารดา และพี่ ๆ ของข้าพเจ้า ที่คอยห่วงใย ให้โอกาส ให้กำลังใจ และให้การสนับสนุนทางการศึกษามาโดยตลอด

โครงงานเล่มนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาจากอาจารย์ที่ปรึกษา อ.ดร.วิภาวี อุสาหะ ผู้ที่เป็นเจ้าของแนวคิดเริ่มแรกของโครงงาน ระบบปัญญาประดิษฐ์*(Artificial Intelligence หรือ A.I)* เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบสัญญาณไฟจราจร ที่ให้ความช่วยเหลือทางด้านแนวคิด การดูแลเอาใจใส่ติดตามงาน ชี้แนะข้อบกพร่อง ตลอดจนฝึกฝนและสนับสนุนข้าพเจ้าให้มีความสามารถในการทำโครงงานจนสามารถนำเสนอผลงานให้เป็นที่รู้จักและยอมรับได้

ขอขอบคุณผู้ที่เกี่ยวข้องอื่นๆดังนี้

ขอขอบคุณ คุณมณีรัตน์ ทุมพงษ์ เลขานุการประจำสาขาวิชาวิศวกรรมโทรคมนาคมที่ให้ความสะดวกในการติดต่อกับอาจารย์ และการค้นหาเอกสาร คุณ คุณประพล จาระตะคุ วิศวกรประจำอาคารเครื่องมือ 3 ที่ช่วยเป็นธุระในการสั่งซื้ออุปกรณ์ คุณฉัตรชัย ถาจอหอ เจ้าหน้าที่ดูแลอุปกรณ์ห้องปฏิบัติการโทรคมนาคมที่ช่วยอำนวยความสะดวกในการเบิกอุปกรณ์ พี่ๆนักศึกษาปริญญาโทวิศวกรรมโทรคมนาคมทุกๆคนที่ให้การสนับสนุนและคำปรึกษา เพื่อนๆนักศึกษาสาขาวิชาโทรคมนาคมทุกๆคนที่เป็นกำลังใจให้มาโดยตลอดมา และรุ่นน้องนักศึกษาสาขาวิชาโทรคมนาคม รุ่นที่ 12 ที่ให้ความร่วมมือในการทดสอบผลการทดลองโครงงานนี้

คณะผู้จัดทำ

นายนพดล ลัยนันท์

นาย ธัชกร ไกรธนานันต์

นาย พงศกร คงสมบูรณ์

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูปภาพ	จ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ของโครงงาน	4
1.3 ขอบเขตของโครงงาน	4
1.4 ขั้นตอนการดำเนินงาน	4
บทที่ 2 ทฤษฎี อุปกรณ์ และการออกแบบ	5
2.1 ทฤษฎี	5
2.2 อุปกรณ์การทดลอง	8
2.2.1 INFRARED RECEIVER	8
2.2.2 INFRARED TRANCEIVER	9
2.2.3 วงจรประมวลผลสัญญาณจราจร	14
2.3 การออกแบบ	17
2.4 การเขียนโปรแกรม	20
2.4.1 การเขียนโปรแกรมจำลองระบบ	20
2.4.2 การเขียนโปรแกรมจำลองระบบในไมโครคอนโทลเลอร์	21
บทที่ 3 การทดลอง	22
3.1 การทดลองในส่วนของโปรแกรมจำลองระบบ	22
3.1.1 วิธีการทดลอง	22
3.1.2 ผลการทดลอง	23
3.1.3 วิเคราะห์ผลการทดลอง	25
3.1.4 สรุป	26

สารบัญ (ต่อ)

3.2 การทดลองในสถานการณ์จริง 27

3.2.1 วิธีการทดลอง 27

3.2.2 ผลการทดลอง 42

3.2.3 วิเคราะห์ผลการทดลอง 47

3.2.4 สรุป 48

บทที่ 4 บทสรุป และข้อเสนอแนะ 49

4.1 สรุปผลโครงงาน 49

4.2 สิ่งที่ได้รับจากโครงงาน 49

4.3 ปัญหาและอุปสรรคระหว่างการทดลอง 49

4.4 ปัญหาหลังการทดลอง 51

4.5 แนวทางการแก้ไข 51

4.6 ข้อเสนอแนะ 51

บรรณานุกรม 53

ประวัติผู้เขียน 54

ภาคผนวก 55

ภาคผนวก ก. โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ 56

ภาคผนวก ข. โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา 67

ภาคผนวก ค. โปรแกรมระบบการทดลองในสถานการณ์จริงโดยใช้ระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา 77

ภาคผนวก ง. โปรแกรมระบบการทดลองในสถานการณ์จริงที่ใช้ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ 79

ภาคผนวก จ. ตารางข้อมูลในการนับจำนวนรถ (จักรยานยนต์, รถยนต์) ที่บริเวณสามแยกหน้าส่วนกิจการนักศึกษาทางเข้าเรียนรวม 89

ภาคผนวก ฉ. ตารางข้อมูลดิบที่ได้จากการทดลองในสถานการณ์จริง 101

ภาคผนวก ช. Datasheet AT89S8252 104

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 แสดงปริมาณผู้รถบนท้องถนน 2

รูปที่ 1.2 แสดงปริมาณยานพาหนะที่เพิ่มมากขึ้นทุกๆปี 3

รูปที่ 1.3 แสดงผลจากการเกิดปัญหาการจราจรติดขัด 3

รูปที่ 2.1 ภาพแสดงการทำงานของ การเรียนรู้แบบรีอินฟอร์สเมนท์ 6

รูปที่ 2.2 ภาพเซ็นเซอร์ อินฟาเรด ภาครับ 10

รูปที่ 2.3 ภาพวงจรเซ็นเซอร์ อินฟาเรด ภาครับ 11

รูปที่ 2.4 ภาพเซ็นเซอร์อินฟาเรดภาคส่ง 12

รูปที่ 2.5 ภาพวงจร เซ็นเซอร์อินฟาเรดภาคส่ง 13

รูปที่ 2.6 ภาพวงจรประมวลผลสัญญาณจราจร 15

รูปที่ 2.7 ภาพไมโครคอนโทลเลอร์ในการควบไฟจราจร 16

รูปที่2.8 ภาพแสดงรูปแบบของถนน 17

รูปที่2.9 ภาพแผนการจำลองสถานการณ์ *Q-Learning* 18

รูปที่2.10 แผนภาพ Flowcharts ในการเขียนโปรแกรมจำลองระบบใน C++ 20

รูปที่2.11 แผนภาพ Flowcharts ในการเขียนโปรแกรมจำลองระบบในไมโครคอนโทลเลอร์ 21

รูปที่ 3.1 ภาพแผนภูมิเปรียบเทียบอัตราผ่านในรูปแบบต่างๆ 23

รูปที่ 3.2 ภาพผลการทดลองโปรแกรมจำลองระบบทั้ง 3 รูปแบบเมื่อกำหนดอัตรารถเข้าเท่ากับ 20, 20 และ 20 คันต่อนาทีตามลำดับ 23

รูปที่ 3.3 ภาพแผนภูมิเปรียบเทียบอัตราผ่านในรูปแบบต่างๆ 24

รูปที่ 3.4 ภาพผลการทดลองโปรแกรมจำลองระบบทั้ง 3 รูปแบบเมื่อกำหนดอัตรารถเข้าเท่ากับ 20, 10 และ 10 คันต่อนาทีตามลำดับ 24

รูปที่ 3.5 ภาพจำลองแสดงการติดตั้งอุปกรณ์ในการทดลองจากสถานที่จริง 28

รูปที่ 3.6 ภาพแยกการทดลองประตู 3 29

รูปที่3.7 ภาพแสดงการติดตั้งตัวไมโครคอนโทลเลอร์ 30

รูปที่3.8 ภาพการติดตั้งสัญญาณไฟจราจร 31

รูปที่ 3.9 ภาพบริเวณแยกประตู 3 ก่อนการทดลอง 32

รูปที่ 3.10 ภาพแสดงการควบการปล่อยรถ เส้นที่ 1 33

รูปที่ 3.11 ภาพแสดงการควบการปล่อยรถ เส้นที่ 2 34

รูปที่ 3.12 ภาพแสดงการควบการปล่อยรถ เส้นที่ 3 35

รูปที่3.13 ภาพแสดงการปล่อยรถตามจำนวนที่ระบุในการทดลอง 36

สารบัญรูปภาพ (ต่อ)

รูปที่ 3.14 ภาพการทดลองโดยใช้มนุษย์ในการตัดสินใจ 37
รูปที่ 3.15 ภาพแสดงการปฏิบัติตามกฎจราจรในการทดลอง 38
รูปที่ 3.16 ภาพแสดงพฤติกรรมในการใช้เส้นทาง 39
รูปที่ 3.17 ภาพกลุ่มผู้ร่วมทดสอบปฏิบัติตามไฟจราจรที่ปรากฏ 40
รูปที่ 3.18 ภาพแสดงการเก็บข้อมูลและจับเวลา 41
รูปที่ 3.19 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง เมื่อความเร็วในการทดลองเป็น 20 กิโลเมตรต่อชั่วโมง 43
รูปที่ 3.20 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง เมื่อความเร็วในการทดลองเป็น 30 กิโลเมตรต่อชั่วโมง 44
รูปที่ 3.21 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง เมื่อความเร็วในการทดลองเป็น 40 กิโลเมตรต่อชั่วโมง 45
รูปที่ 3.22 ภาพแผนภูมิแสดงอัตราผ่านรวมทั้งหมดของรถในแยกการทดลองที่ความเร็วต่างๆ 46
รูปที่ 4.1 ภาพการใช้คนช่วยในการบอกสัญญาณไฟจราจร 50
รูปที่ 4.2 ภาพแสดงบริเวณที่แยกต่างๆอยู่รวมกันและแยกที่มีความซับซ้อน 52

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงการตั้งจำนวนนับทำโดยที่เราป้อนแรงดันไฟฟ้าที่ขาอินพุทแบบเลขฐาน 2 8

ตารางที่ 2.2 แสดงกำหนดเวลาของไฟเขียวในการปล่อยรถทั้ง 3 เส้นทาง 17

ตารางที่ 3.1 แสดงผลการนับจำนวนรถที่ผ่านสัญญาณไฟจราจรรวม 42

บทที่ 1
บทนำ

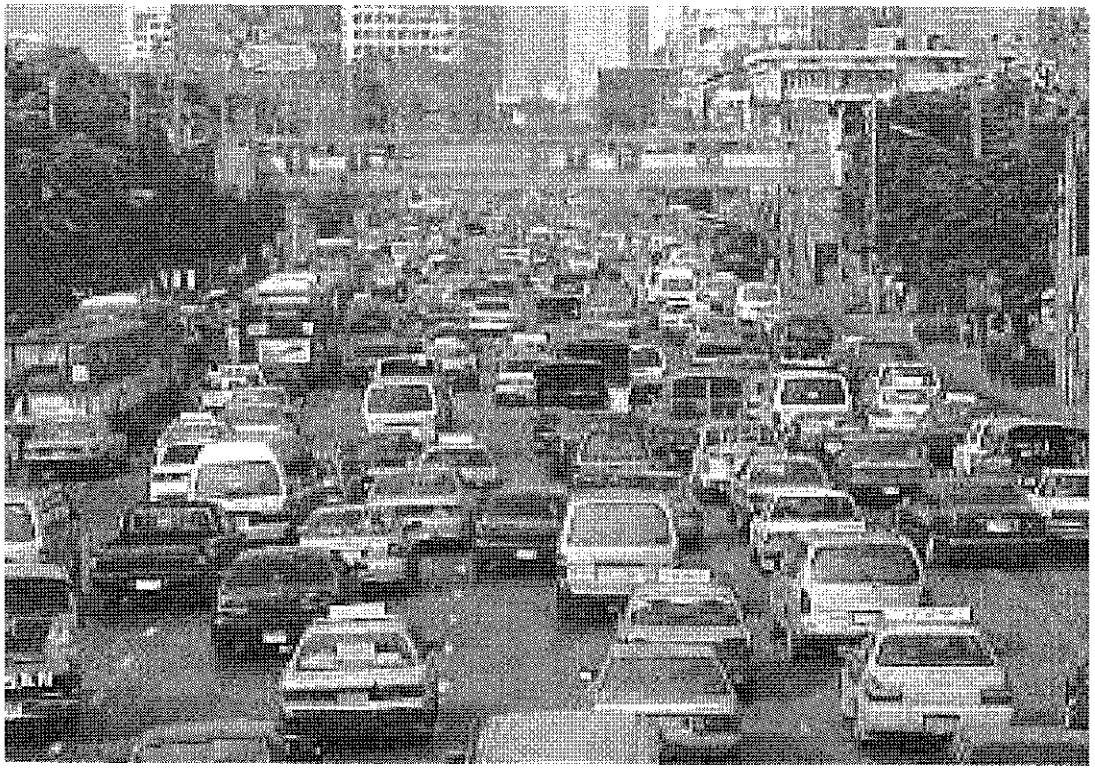
1.1 กล่าวนำ

ปัจจุบันนี้ระบบคมนาคมเริ่มเข้ามามีบทบาทสำคัญต่อการใช้ชีวิตประจำวันมากยิ่งขึ้น หรืออาจจะกล่าวได้ว่า ระบบคมนาคมได้เป็นส่วนหนึ่งของการใช้ชีวิตประจำวัน โดยที่เราสามารถเลือกการเดินทางได้ทั้ง ทางบก ทางอากาศ และทางน้ำ แต่การเดินทางที่ได้รับความนิยมและใช้งานมากที่สุดคือ การจราจรทางบก โดยจะเป็นการใช้รถยนต์และรถจักรยานยนต์เดินทางเป็นยานพาหนะหลัก ทั้งเป็นการเดินทางโดยรถยนต์นั่งส่วนบุคคล รถโดยสาร รวมถึงการขนส่งสินค้าและการบริการ โดยที่ปริมาณของผู้ใช้ยานพาหนะที่มีเพิ่มมากขึ้น และการที่มีปริมาณยานพาหนะที่เพิ่มขึ้นทุกๆปี แต่จำนวนเส้นทางของถนนที่ใช้ในการสัญจรไปมา ยังคงไม่เพียงพอต่อความต้องการของผู้ใช้ยานพาหนะ ซึ่งผลกระทบที่พบตามมาคือ ปัญหาการจราจรติดขัดและการเกิดอุบัติเหตุ จึงต้องมีการแก้ปัญหาดังกล่าวโดยการควบคุมระบบการจราจร โดยใช้สัญญาณไฟจราจรในการควบคุมปริมาณการใช้รถในแต่ละเส้นทาง ซึ่งจะติดตั้งบริเวณที่เป็นจุดที่เส้นทางแต่ละสายมาพบกันหรือบริเวณแยกต่างๆ เพื่อให้การเดินทางมีความราบรื่นมากยิ่งขึ้น

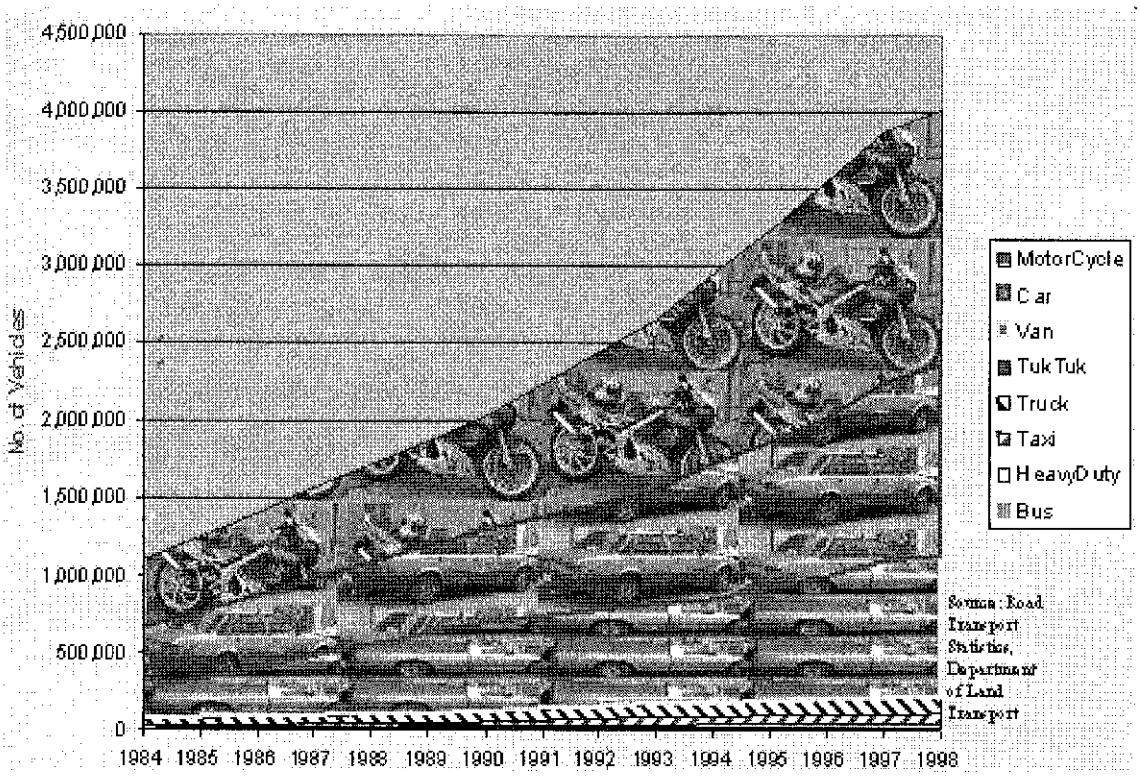
สิ่งที่พบตามมาภายหลังคือ การที่ระบบสัญญาณไฟจราจรไม่สามารถทำงานได้เต็มประสิทธิภาพอันเนื่องมาจากปริมาณการใช้ยานพาหนะในแต่ละช่วงเวลาไม่เท่ากัน ดังเช่นในชั่วโมงเร่งด่วน จะพบว่ามีปริมาณการใช้งานในบางเส้นทางมากกว่าปกติ จึงเกิดการติดขัดในการระบายยานพาหนะในเส้นทางที่ต้องการไม่ทัน และส่งผลไปยังเส้นทางอื่นๆที่อยู่ใกล้เคียงลักษณะแบบลูกโซ่ เนื่องจากระบบสัญญาณไฟจราจรที่ใช้งานอยู่ในปัจจุบันนั้น เป็นระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา*(Fixed Time)* ซึ่งจะต้องทำการกำหนดการตั้งปริมาณเวลาที่ต้องการใช้ควบคุมในแต่ละเส้นทางตามความความต้องการ โดยที่ระบบจะทำการเปลี่ยนสัญญาณไฟจราจรเมื่อถึงเวลาที่ทำการกำหนดไว้เท่านั้น ดังนั้นจะพบว่าเมื่อถึงช่วงเวลาเร่งด่วนดังกล่าวจะต้องมีการใช้ตำรวจจราจรในการควบคุมสัญญาณไฟจราจรหรือใช้สัญญาณมือช่วยโบกบนกลางถนน ซึ่งเป็นการแก้ปัญหาเฉพาะหน้าดังกล่าวได้ในระดับหนึ่ง แต่จะพบว่าวิธีการดังกล่าวอาจก่อให้เกิดปัญหาขึ้นได้เช่นกัน เมื่อมีปริมาณยานพาหนะจำนวนมากๆในหลายๆเส้นทางพร้อมๆกัน และหากเกิดการตัดสินใจเปลี่ยนสัญญาณไฟจราจรที่ไม่เหมาะสมจะยิ่งทำให้เกิดปัญหาการจราจรติดขัดมากยิ่งขึ้น ทั้งอาจเกิดอุบัติเหตุขึ้นแก่เจ้าหน้าที่ตำรวจจราจรขณะกำลังปฏิบัติงานบนกลางถนนด้วยเช่นกัน

ดังนั้นคณะผู้จัดทำจึงจัดทำโครงงาน ระบบ *A.I. (Artificial Intelligence)* เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบไฟจราจรขึ้น เพื่อทำการสร้างระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์*(A.I. หรือ Artificial Intelligence)* โดยระบบดังกล่าวเป็นระบบควบคุมสัญญาณไฟจราจรที่สามารถพัฒนาตนเองให้เข้ากับสภาพที่ใช้งานจริงในแต่ละสถานที่ที่ทำการติดตั้ง อันเนื่องมาจากการนำกระบวนการทางคณิตศาสตร์มาใช้ในการวิเคราะห์และตัดสินใจ โดย

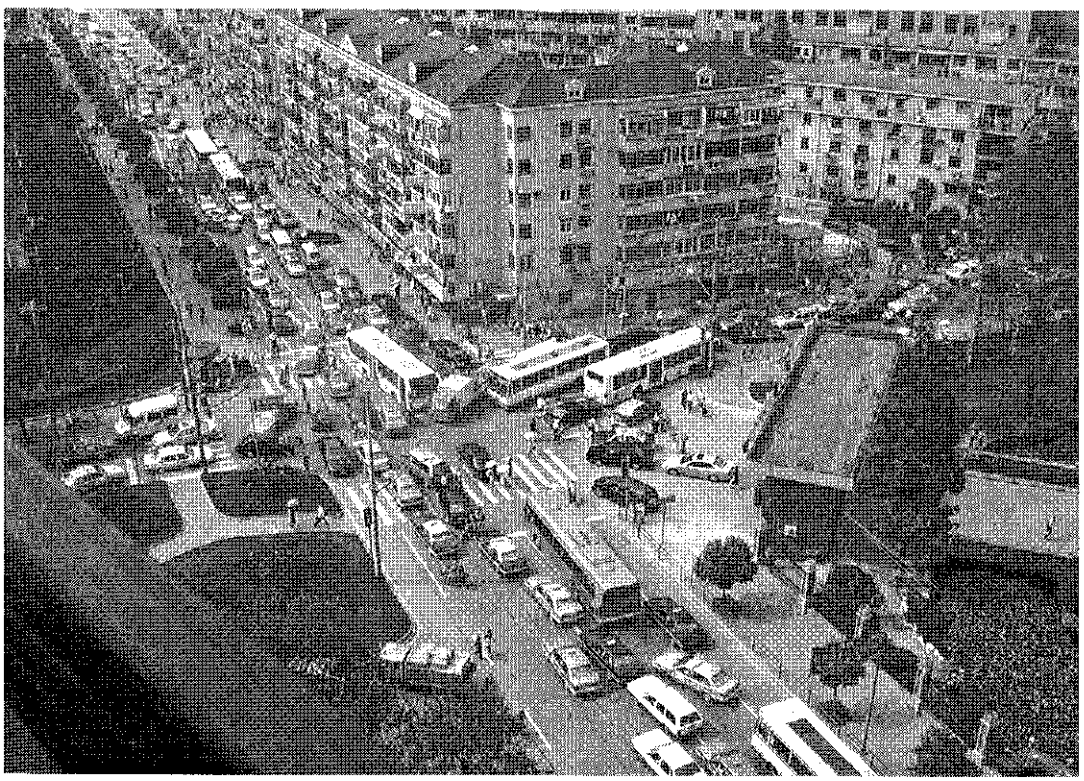
คำนึงถึงค่าอัตราผ่านของรถต่อช่วงเวลาที่มีค่ามากที่สุด เพื่อให้ได้การตัดสินใจที่ดีที่สุดในการทำการเลือกเปลี่ยนสัญญาณไฟจราจรในแต่ละครั้ง ซึ่งตัวโปรแกรมนั้นจะสามารถเรียนรู้และพัฒนาได้ด้วยตัวเองจากการตัดสินใจที่ได้เคยกระทำลงไปแล้ว และเก็บผลดังกล่าวไว้เป็นฐานข้อมูลในการตัดสินใจในครั้งต่อๆไป โดยที่ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์จะสามารถทำการตัดสินใจในการเปลี่ยนสัญญาณไฟจราจรได้อย่างเหมาะสม ณ ช่วงเวลานั้นมากยิ่งขึ้น ส่งผลให้อัตราผ่านของรถต่อช่วงเวลาที่บริเวณดังกล่าวจะมีค่าที่สูงมากขึ้น ซึ่งในโครงงานเล่มนี้นี้ทางคณะผู้จัดทำโครงงานได้นำเสนอเกี่ยวกับทฤษฎี รายละเอียดในกระบวนการคิดวิเคราะห์ สมการทางคณิตศาสตร์ อุปกรณ์ในการทดลอง และตัวต้นแบบระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์*(A.I. หรือ Artificial Intelligence)* จากสถานที่จริง รวมอยู่ในโครงงานฉบับนี้



รูปที่ 1.1 แสดงปริมาณผู้รถบนท้องถนน



รูปที่ 1.2 แสดงปริมาณยานพาหนะที่เพิ่มมากขึ้นทุกๆปี



รูปที่ 1.3 แสดงผลจากการเกิดปัญหาการจราจรติดขัด

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษากลไกการทำงานของระบบ *A.I. (Artificial Intelligence)* ที่เรียกว่าเทคนิค *Reinforcement Learning*

1.2.2 เพื่อศึกษาการประยุกต์ใช้งานทางเทคนิค *Reinforcement Learning* ในการแก้ปัญหาทางระบบสัญญาณไฟจราจร

1.2.3 เพื่อศึกษาและออกแบบระบบควบคุมระบบสัญญาณไฟจราจร โดยใช้เทคนิค *Reinforcement Learning*

1.3 ขอบเขตของโครงงาน

โครงงานฉบับนี้เป็นการนำเสนอระบบ *A.I. (Artificial Intelligence)* เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบสัญญาณไฟจราจร ซึ่งใช้เทคนิค ที่เรียกว่า *Reinforcement Learning* โดยการประยุกต์ใช้โปรแกรมภาษา *C++*

ในการทำโครงงาน *A.I. (Artificial Intelligence)* เพื่อการวิเคราะห์และการแก้ปัญหาทางระบบสัญญาณไฟจราจร มีขอบเขตดังนี้

1.3.1 เขียนโปรแกรมภาษา *C++* ในการคำนวณสมการทางคณิตศาสตร์ให้รับค่าพารามิเตอร์ต่างๆที่สำคัญในการวิเคราะห์รูปแบบของระบบสัญญาณไฟจราจรได้

1.3.2 ทำโปรแกรมการ *Simulate* สถานที่จริงลงในคอมพิวเตอร์

1.3.3 นำโปรแกรม *Simulate* ที่สมบูรณ์มาทำการวิเคราะห์ และสรุปผล

1.3.4 ออกแบบและทำการต่ออุปกรณ์เพื่อใช้ทดลองในสถานการณ์จริง

1.3.5 ตรวจเช็คความถูกต้องโดยเปรียบเทียบกับแบบจำลองทางคอมพิวเตอร์เพื่อวัดประสิทธิภาพในการใช้งานจริง

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาระบบการทำงานของระบบจราจร

1.4.2 ศึกษาสมการทางคณิตศาสตร์และกลไกการทำงานของเทคนิค *Reinforcement Learning*

1.4.3 ศึกษาวิธีการใช้งานโปรแกรมภาษา *C++* เพื่อออกแบบและทำการเขียนโปรแกรมภาษา *C++*

1.4.4 ออกแบบและทำการต่ออุปกรณ์เพื่อใช้ทดลองใน สถานการณ์จริง

1.4.5 นำโปรแกรมที่สมบูรณ์แล้วมาใช้งานร่วมกับอุปกรณ์ในการทดลอง

1.4.6 วิเคราะห์และสรุปผลการทดลอง

บทที่ 2
ทฤษฎี อุปกรณ์ และการออกแบบ

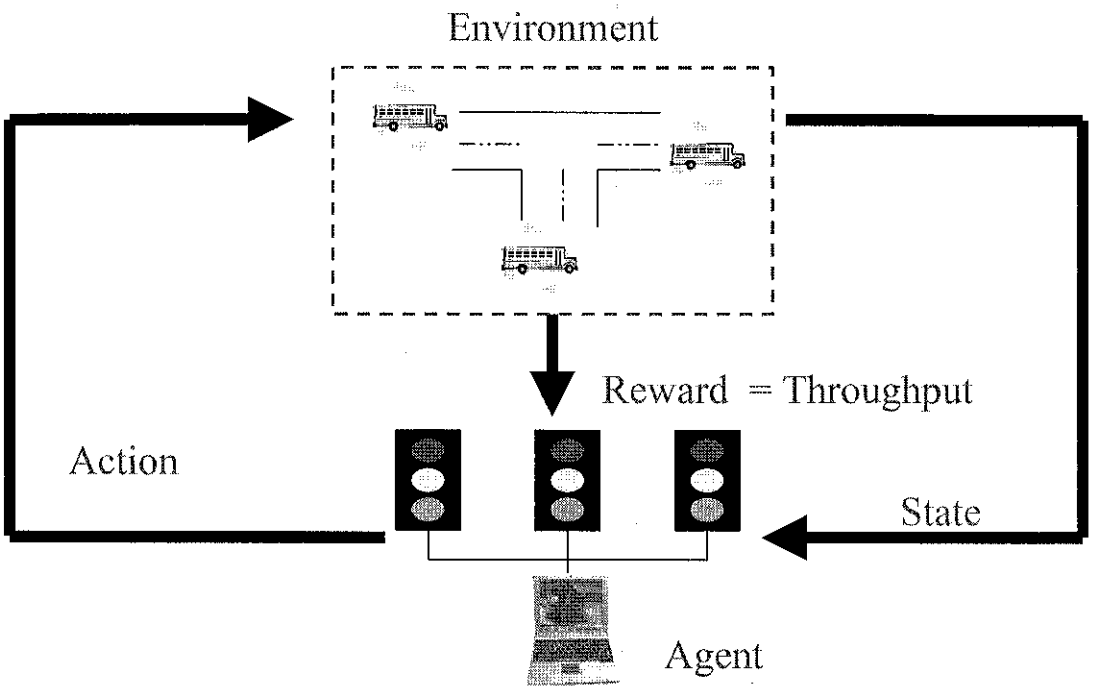
2.1 ทฤษฎี

เมื่อเราได้ทำการศึกษาในส่วนของทฤษฎีพื้นฐาน ซึ่งสามารถนำมาใช้งานเกี่ยวกับการเลือกตัดสินใจต่อเหตุการณ์ที่ต้องการของระบบปัญญาประดิษฐ์ ด้วยความสามารถในการเรียนรู้ต่อสิ่งที่เราสนใจ แล้วสามารถตัดสินใจได้ถูกต้องมากขึ้นตามลำดับ และมีการเรียนรู้จากข้อผิดพลาดในการตัดสินใจ จึงนำข้อได้เปรียบในส่วนดังกล่าวมาทำการวิเคราะห์และออกแบบการควบคุมระบบสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ ซึ่งมีความละเอียดซับซ้อนในส่วนของการตัดสินใจเพื่อทำการระบายปริมาณยานพาหนะบนท้องถนนในแต่ละทางแยกให้มีประสิทธิภาพมากและอัตราผ่านของรถต่อช่วงเวลาให้มีค่าที่มากที่สุด โดยการให้พิจารณาถึงปริมาณของยานพาหนะที่ทำการตรวจสอบจำนวนได้ในแต่ละเส้นทาง มาทำการวิเคราะห์ด้วยกระบวนการทางคณิตศาสตร์ในตัวโปรแกรม จะพบว่า เงื่อนไขของปัญหาที่ได้ทำการวิเคราะห์เป็นเงื่อนไขแบบ *Markov decision process (finite MDP)* ดังนั้นผลลัพธ์ที่ได้ออกมาจากการวิเคราะห์ในแต่ละช่วงเวลาจะมีค่าเปลี่ยนไปเรื่อยๆตามสถานการณ์ที่เป็น จะอยู่ในรูปแบบที่เราสามารถทำการกำหนดรูปแบบของการกระทำและเงื่อนไขให้อยู่ในเชิงจำกัด โดยกระบวนการดังกล่าวเราจะต้องเริ่มต้นจากการพิจารณาถึงบริเวณแยกถนนที่เราสนใจทำการทดลอง และทำการออกแบบตัวโปรแกรมให้สอดคล้องกับสภาพความเป็นจริง หลังจากนั้นจึงทำการเลือกกระบวนการในการแก้ปัญหาและด้วยเงื่อนไขจากข้างต้นนั้นคณะผู้จัดทำจึงเลือกวิธีการ *Q-Learning* ซึ่งเป็นวิธีการที่ตรงกับปัญหาที่ทำการวิเคราะห์ในข้างต้นมากที่สุด เนื่องจากเป็นวิธีการที่สามารถตัดสินใจได้รวดเร็ว และสามารถตรวจสอบผลลัพธ์ที่ได้ว่ามีความถูกต้องเหมาะสมหรือไม่ สามารถนำไปปรับปรุงในการตัดสินใจครั้งถัดไปในระบบการควบคุมระบบสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์(A.I. หรือ *Artificial Intelligence*)ได้

ในการเขียนโปรแกรมภาษา *C++* ของระบบปัญญาประดิษฐ์ด้วย วิธี *Q-Learning* ซึ่งเป็นรูปแบบหนึ่งของเทคนิคการเรียนรู้แบบรีอินฟอร์สเมนท์ *(Reinforcement Learning หรือ RL)* และวิธีการดังกล่าวจะเป็นระบบที่มีการเก็บสะสมค่าที่ต้องการทำการวิเคราะห์ แล้วหาค่าเฉลี่ยเป็นค่าผลลัพธ์ที่มีความสัมพันธ์เกี่ยวข้องกับ *Reward* แล้วนำค่าที่ได้จากการวิเคราะห์มากำหนดกระบวนการถัดมา โดยในระบบการเรียนรู้แบบรีอินฟอร์สเมนท์ ในส่วนของวิธี *Q-Learning* จะประกอบไปด้วยส่วนต่างๆ ดังต่อไปนี้

1. *State* คือ สถานะของระบบ
2. *Action* คือ การกระทำของระบบ
3. *Reward* คือ ผลจากการกระทำของ *Action* โดยที่จะมีค่าเป็นตัวเลข
4. *V(s)* คือ ค่าของ *State*
5. *Q(s, a)* คือ ค่าของ สถานะของการกระทำของระบบ หรือ ค่าของการกระทำ
6. *Policy* คือ แนวทางในการปฏิบัติเมื่อดูผลของ *Reward*
7. ε – *greedy* คือค่าความน่าจะเป็นในการเลือกค่า *Q(s, a)*

โดยที่เราสามารถอธิบายการเรียนรู้แบบรีอินฟอร์สเมนท์ ในส่วนของวิธี *Q-Learning* ดังรูป



รูปที่ 2.1 ภาพแสดงการทำงานของ การเรียนรู้แบบรีอินฟอร์สเมนท์

สมการคณิตศาสตร์ที่ใช้ในการคำนวณเพื่อใช้ในส่วนของการเลือกตัดสินใจของระบบปัญญาประดิษฐ์คือ

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right].$$

จากสมการข้างบนสามารถอธิบายได้ว่า วิธีการ *Q-Learning* นี้เป็นการนำค่าของการกระทำของระบบในปัจจุบันมาหาความแตกต่างระหว่างค่าของการกระทำของระบบในช่วงเวลาถัดไปมาบวกเพิ่มเข้ากับค่าของการกระทำของระบบที่ช่วงเวลาปัจจุบัน แล้วทำการนำค่าที่ได้มาใส่ไว้เป็นค่าของการกระทำของระบบ เพื่อใช้ค่าดังกล่าวในการเลือกการกระทำครั้งต่อๆไปในช่วงเวลาถัดไป ซึ่งเราสามารถทำการเขียนโปรแกรมภาษา *C++* ด้วยเงื่อนไขของวิธีการ *Q-Learning* ดังตัวอย่างในต่อไปนี้ เพื่อใช้ในส่วนของการเขียนโปรแกรมจำลองระบบ

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g., ε-greedy)
    Take action a, observe r, s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$
    $s \leftarrow s'$;
  until s is terminal

หลังจากทำการเขียนโปรแกรมภาษา *C++* ด้วยเงื่อนไขดังกล่าวนี้ และนำไปใช้ในส่วนของโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจร โดยส่วนของเงื่อนไขนี้จะทำให้โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรจะมีการคิดวิเคราะห์และแก้ปัญหาด้วยระบบปัญญาประดิษฐ์ โดยจะได้โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบ่งออกเป็น 2 รูปแบบคือ โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ และโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ที่มีโครงสร้างพื้นฐานของโปรแกรมเดียวกัน ซึ่งสามารถนำโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรที่ได้ไปทำการรันโปรแกรมเพื่อใช้ในการทดลองผลของโครงงานต่อไป

2.2 อุปกรณ์การทดลอง

2.2.1 INFRARED RECEIVER

ในวงจรภาครับสัญญาณอินฟาเรดนี้จะประกอบด้วยวงจร 3 ส่วนด้วยกันคือ

2.2.1.1.วงจรภาครับสัญญาณและฟิลเตอร์

อุปกรณ์ที่ทำหน้าที่รับสัญญาณคือ ไดโอด D1 สัญญาณที่ได้ตกคร่อม D1 จะป้อนผ่านวงจรฟิลเตอร์ซึ่งความถี่ 5 KHz จะผ่านไปได้ โดยที่ความถี่ต่ำ ๆ วงจรจะมีอัตราขยายเท่ากับ 1 ส่วนความถี่ที่5KHz จะมีอัตราขยายหลายพันเท่า

2.2.1.2.วงจรจัดรูปคลื่น

สัญญาณที่รับได้อาจจะมีรูปคลื่นที่ไม่เหมาะสม จึงได้จัดรูปคลื่นใหม่ที่ทีการเปลี่ยนแปลงอย่างรวดเร็วทั้งช่วงขึ้นและช่วงตก โดยที่ ใช้ IC LM1458 เปรียบเทียบแรงดันไฟที่ขา 6 ด้วย VR1 หากแรงดันที่เข้ามาสูงกว่าแรงดันไฟที่ขา 6 จะได้แรงดันไฟเอ้าท์พุทสูง หากแรงดันที่เข้ามาต่ำกว่าแรงดันไฟที่ขา 6 จะได้แรงดันไฟเอ้าท์พุทเท่ากับ 0 และเราสามารถรู้ว่ามีสัญญาณจากตัวส่งได้จาก LED D4

2.2.1.3.วงจรหารความถี่

วงจรนี้ทำงานโดยใช้ ไอซีหารความถี่ชนิดโปรแกรมได้ตั้งแต่หาร 2-256 ด้วยการจัดแรงดันไฟฟ้าที่ขาคอนโทรลอินพุท ซึ่งมีอยู่ 8 ขา คือขา 4, 5, 6, 7, 10, 11, 12, 13 การจัดโปรแกรมนี้ใช้แบบเป็นเลขฐาน 2 โดยให้ 1= ไฟบวก และ 0= กราวด์ โดยที่จะทราบว่ามีเอ้าท์พุทออกจากวงจรหารความถี่ได้จาก LED D5

การตั้งจำนวนนับทำโดยที่เราป้อนแรงดันไฟฟ้าที่ขาอินพุทแบบเลขฐาน 2 ตามจำนวนนับที่ต้องการ ซึ่งที่ขาต่างจะแสดงจำนวนนับดังนี้

ขาที่	4	5	6	7	10	11	12	13
กำลัง 2 =	0	1	2	3	4	5	6	7
ค่าฐานสิบ	1	2	4	8	16	32	64	128

ตารางที่ 2.1 แสดงการตั้งจำนวนนับทำโดยที่เราป้อนแรงดันไฟฟ้าที่ขาอินพุทแบบเลขฐาน 2

2.2.1.4 การสร้างและการทดสอบ

นำวงจรทั้งหมดประกอบลงบนแผ่นปริ้นท์ ป้อนแรงดันไฟฟ้า 9 โวลต์ให้กับชุดส่ง จากนั้นทดสอบโดยต่อแรงดันไฟฟ้า +12V ให้กับชุดรับสัญญาณ ปรับVR1 จนกระทั้ง LED D4 จะติด วงจรภาครับสัญญาณอินฟาเรดนี้สามารถวางห่างวงจรภาคส่งสัญญาณได้ในระยะประมาณ 10 เมตร เมื่อเรานำสิ่งกีดขวางผ่านไปผ่านมาระหว่างวงจรภาคส่งสัญญาณและวงจรภาครับสัญญาณ จะสามารถได้ยินเสียงรีเลย์ทำงานด้วย

2.2.2 INFRARED TRANCEIVER

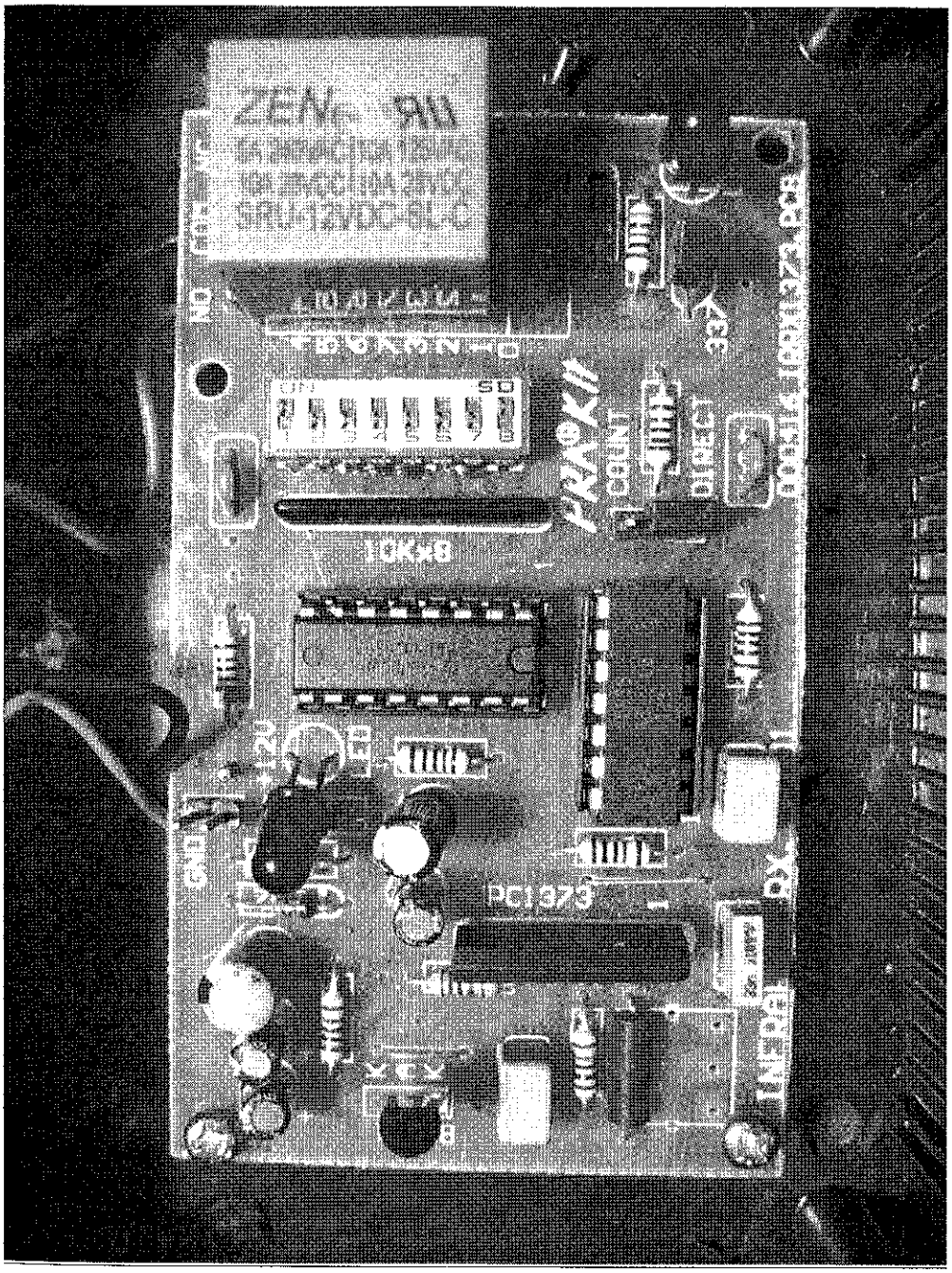
2.2.2.1 วงจรภาคส่งสัญญาณอินฟาเรด

วงจรนี้ที่เด่นชัดคือ ไอซี 555ที่ทำหน้าที่เป็นตัวกำเนิดความถี่ 5KHz โดยที่ค่าความถี่ของวงจรกำหนดโดยค่า VR1, R1, R2 และ C1 ทั้งนี้สามารถปรับแต่งความถี่ได้ด้วยการปรับค่า VR1

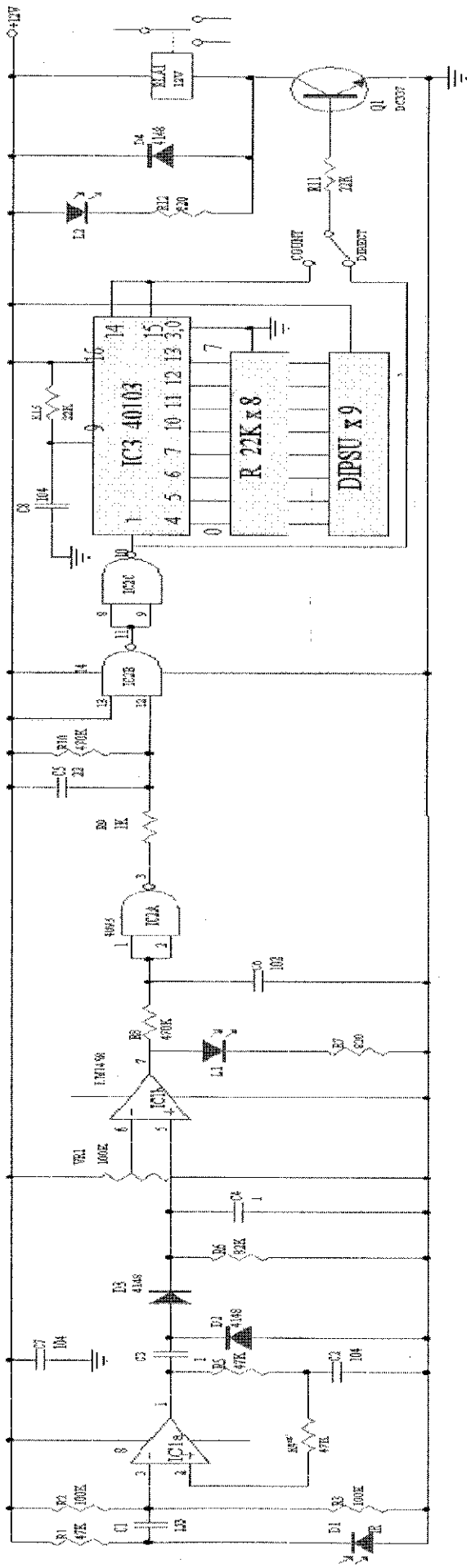
สามารถรู้ได้ว่าวงจรภาคส่งสัญญาณอินฟาเรดได้รับแรงดันไฟฟ้าแล้วจาก R4 และ LED D4 ที่ได้ต่อไว้

2.2.2.2 การสร้างและการทดสอบ

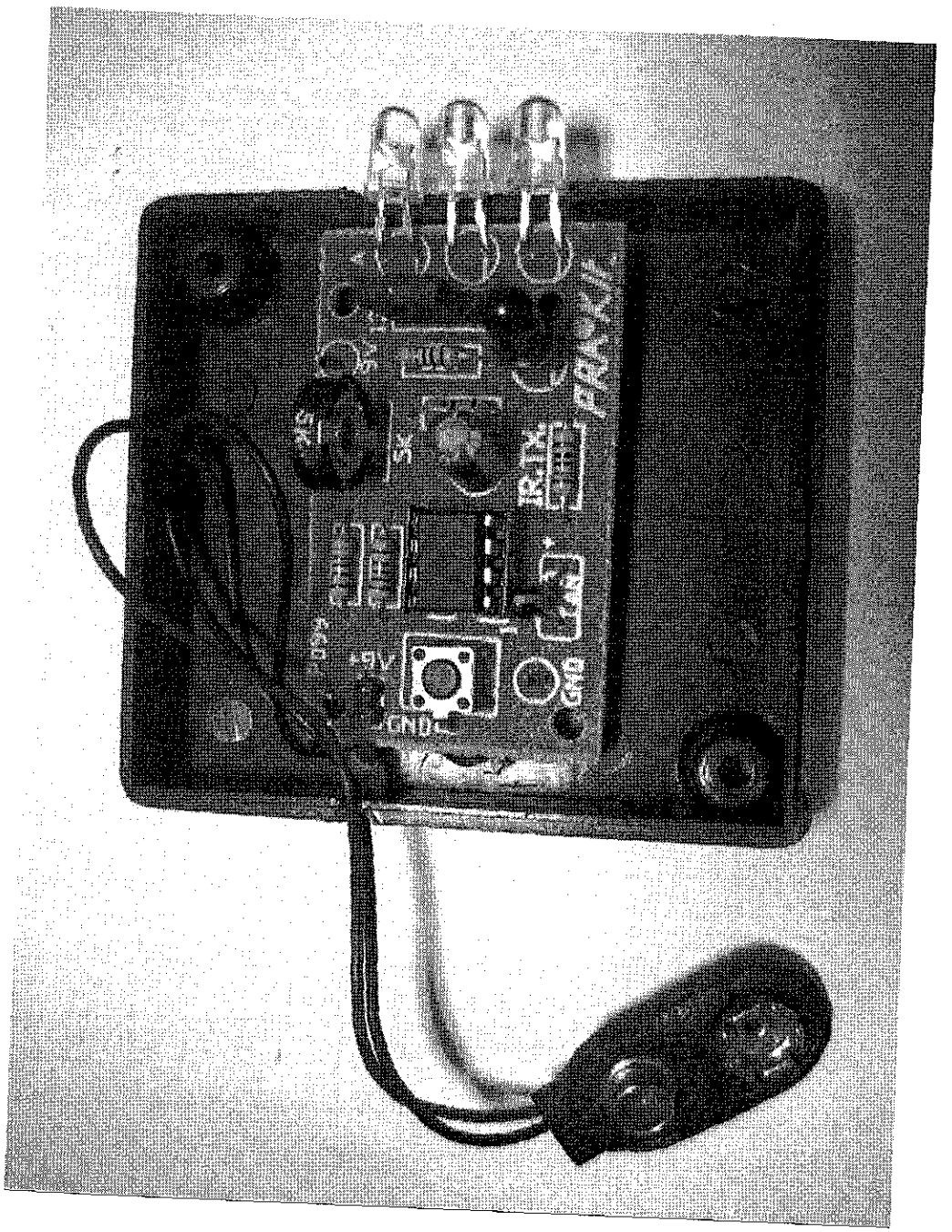
นำวงจรทั้งหมดประกอบลงบนแผ่นปริ้นท์ ป้อนแรงดันไฟฟ้า 9 โวลต์ให้กับชุดส่ง สังเกตไฟLED D4 จะติด จากนั้นทดสอบโดยต่อแรงดันไฟฟ้า +12V ให้กับชุดรับสัญญาณปรับVR1 จนกระทั้ง LED D4 จะติด วางวงจรภาคส่งสัญญาณอินฟาเรดและวงจรภาครับสัญญาณอินฟาเรด หางจากกันในระยะประมาณ 10 เมตรเมื่อเรานำสิ่งกีดขวางผ่านไปผ่านมาระหว่างวงจรภาคส่งสัญญาณและวงจรภาครับสัญญาณ ทางฝั่งวงจรภาครับสัญญาณอินฟาเรดจะสามารถได้ยินเสียงรีเลย์ทำงานก็จะแสดงว่าวงจรภาคส่งสัญญาณอินฟาเรดและวงจรภาครับสัญญาณอินฟาเรด ทำงานที่ความถี่เดียวกันและวางวงจรภาคส่งสัญญาณตรงกันกับวงจรภาครับสัญญาณ



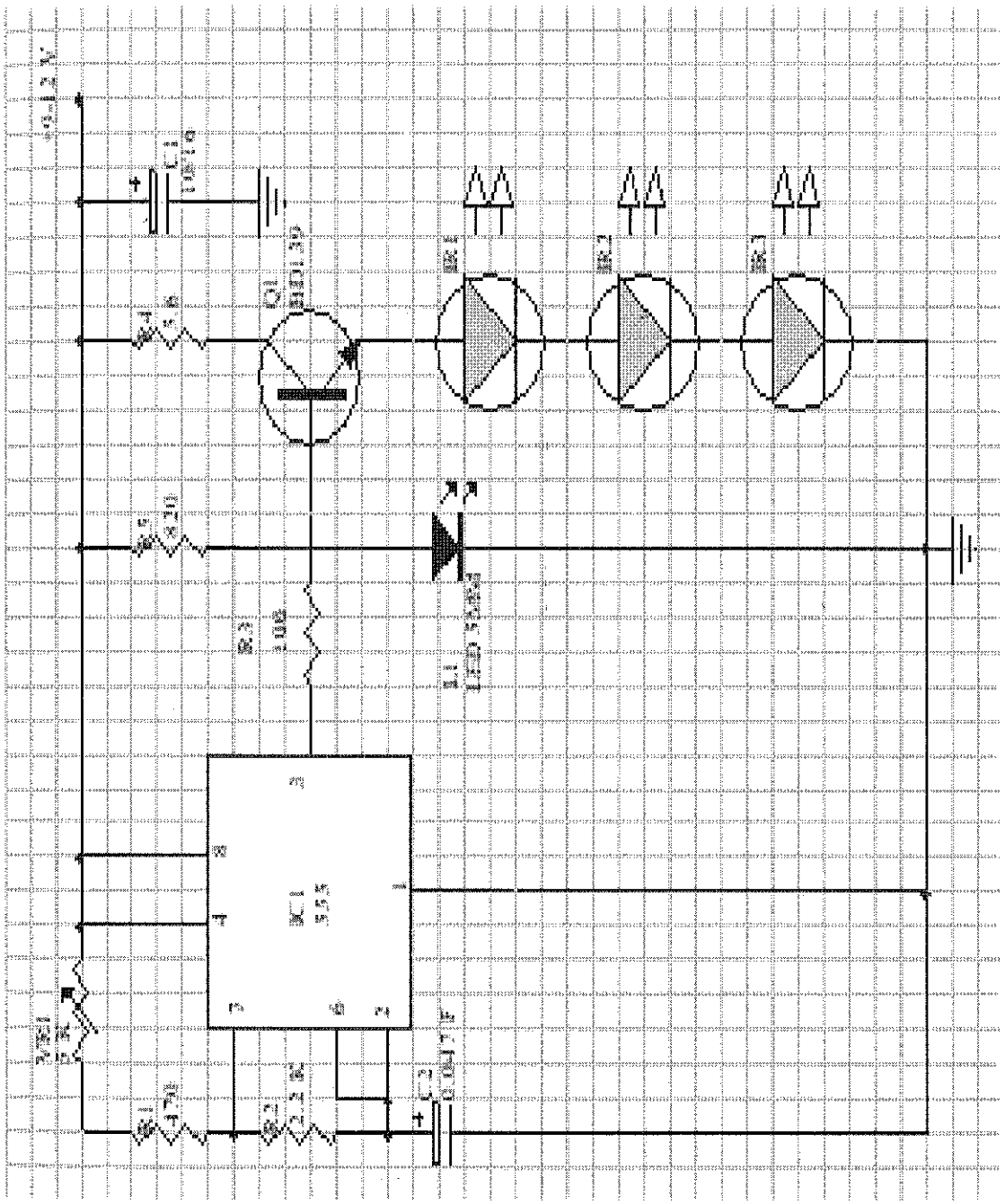
รูปที่ 2.2 ภาพเซ็นเซอร์ อินฟาเรด ภาครับ



รูปที่ 2.3 ภาพวงจรเซ็นเซอร์ อินฟาเรด ภาครับ



รูปที่ 2.4 ภาพเซ็นเซอร์อินฟาเรดภาคส่ง



รูปที่ 2.5 ภาพวงจร เซ็นเซอร์อินฟาเรดภาคส่ง

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

2.2.3 วงจรประมวลผลสัญญาณจราจร

2.2.3.1 วงจรประมวลผลสัญญาณจราจรนี้ใช้

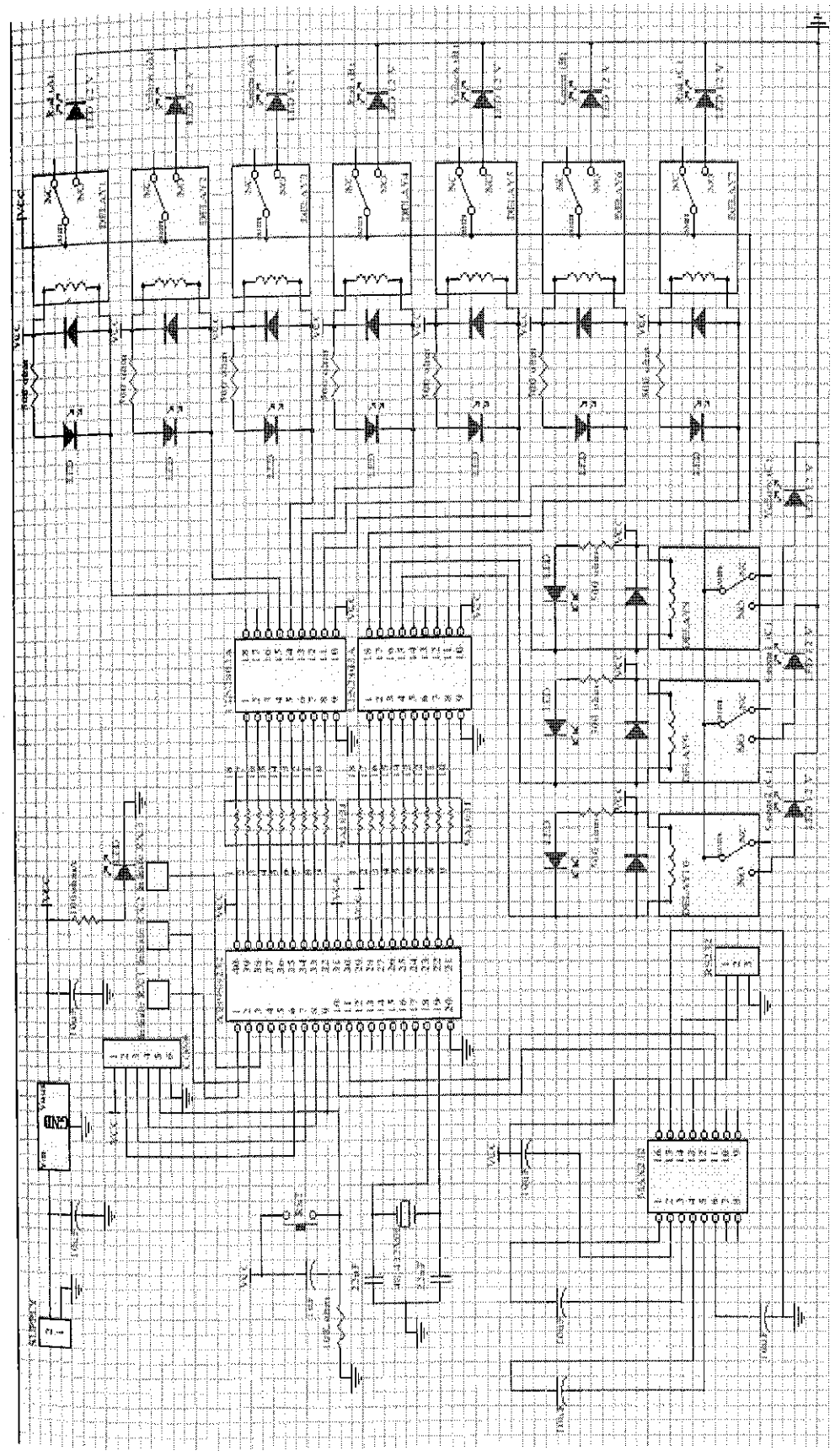
ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 มาทำหน้าที่ทั้งรับสัญญาณจากวงจรภาครับสัญญาณอินฟาเรดมาเพื่อประมวลผลสัญญาณจราจรและส่งสัญญาณไปเพื่อควบคุมไฟจราจรโดยผ่านไอซี เบอร์ ULN2803A

ไอซี เบอร์ ULN2803A ทำหน้าที่รับค่าจาก ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 มาแล้วส่งสัญญาณไฟไปควบคุมรีเลย์ที่ทำหน้าที่เปิดปิดไฟจราจร

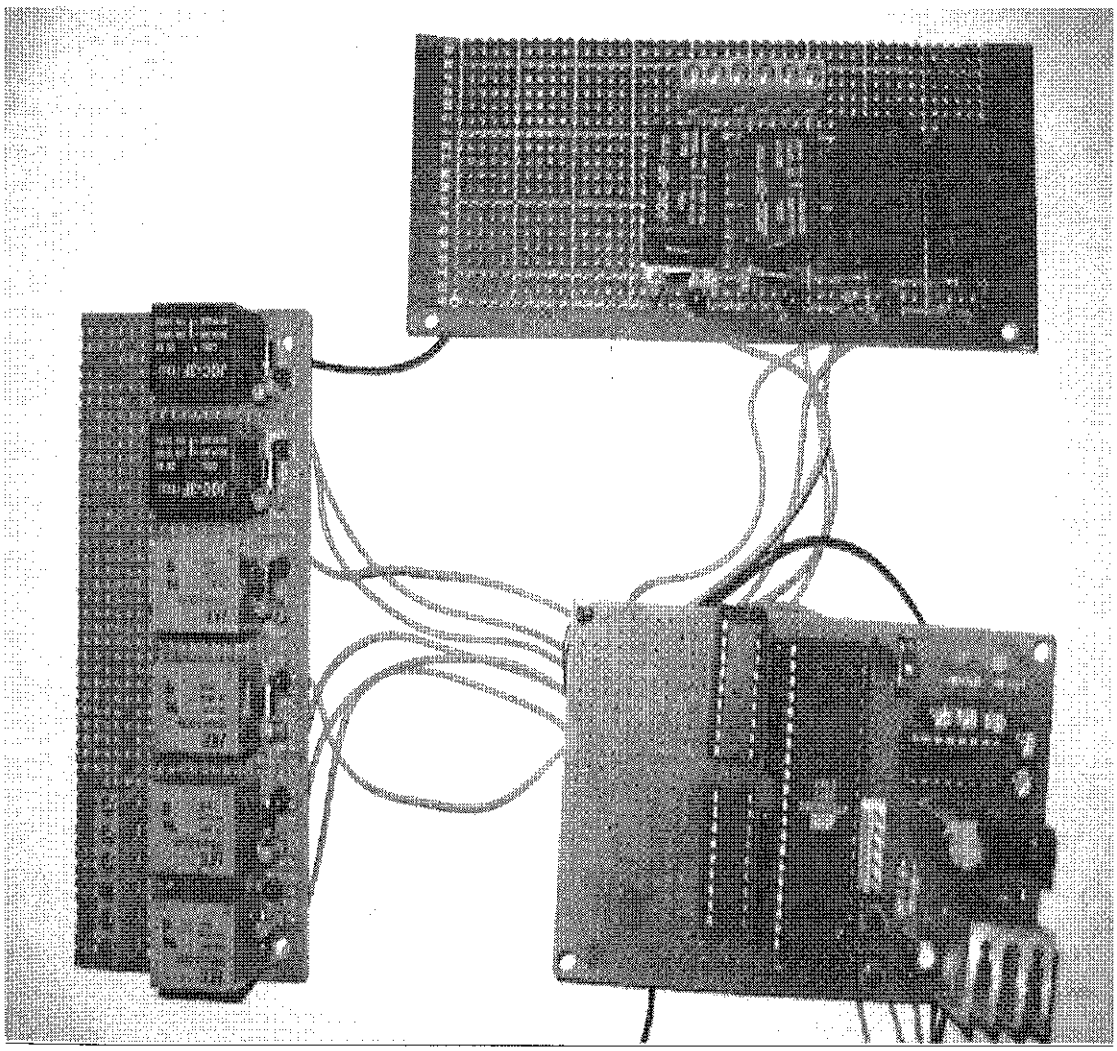
ทรานซิสเตอร์ เบอร์ 7805 เป็นส่วนประกอบของวงจรปรับแรงดัน (Voltage Regulator) เพื่อทำการเปลี่ยนระดับแรงดันจาก $\pm$ 15 V เป็นระดับของสัญญาณ TTL หรือ $\pm$ 5 V

ไอซี เบอร์ Max232 เป็นส่วนประกอบของวงจรที่มีหน้าที่เชื่อมต่อกันระหว่าง วงจรเครื่องรับและคอมพิวเตอร์ โดยวงจรนี้จะทำการส่งข้อมูลจริง 8 บิต ผ่านพอร์ต Serial เพื่อทำการแสดงผลที่จอมอนิเตอร์

Crystal Oscillator ความถี่ 18.432 เมกะเฮิรตซ์ (MHz) ทำหน้าที่สร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์



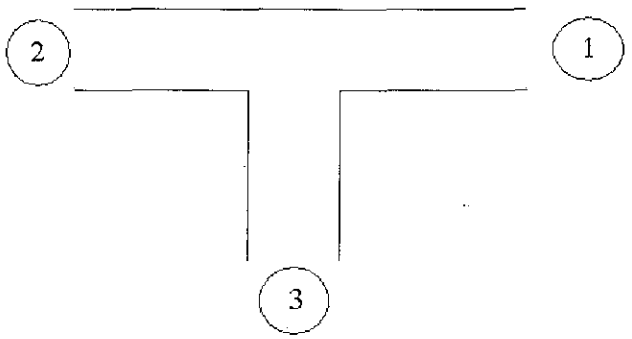
รูปที่ 2.6 ภาพวงจรประมวลผลสัญญาณจราจร



รูปที่ 2.7 ภาพไมโครคอนโทลเลอร์ในการควบไฟจราจร

2.3 การออกแบบ

จากปัญหาในการใช้ยานพาหนะบนท้องถนนนั้น ในที่นี้ได้ทำการเลือกใช้ระบบที่มีถนน 3 เส้นทางในการแก้ปัญหา โดยได้กำหนดรูปแบบของถนนเป็นดังรูป



รูปที่2.8 ภาพแสดงรูปแบบของถนน

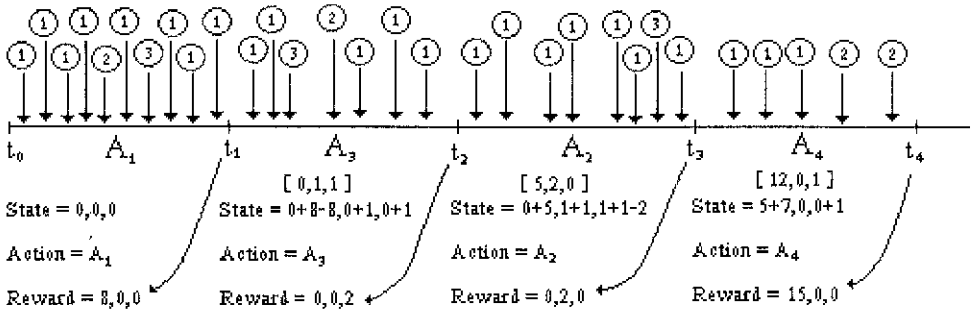
เมื่อได้กำหนดรูปแบบของถนนแล้ว จากนั้นหาสถานที่จริงที่สอดคล้องกับรูปแบบของถนน(ได้เลือกสถานที่ ที่ 3แยก หน้าอาคารส่วนกิจการนักศึกษา ทางเข้าอาคารเรียนรวม ณ มหาวิทยาลัยเทคโนโลยีสุรนารี)เพื่อจะได้สังเกตปัญหาจากสถานที่จริง ซึ่งได้ใช้เวลาในการสำรวจปัญหาเป็นเวลา 2 อาทิตย์และทำการเก็บข้อมูลปริมาณรถจักรยานยนต์ในแต่ละช่วงเวลา

กำหนดเวลาของไฟเขียวในการปล่อยรถทั้ง 3 เส้นทาง โดยได้กำหนดไว้ดังนี้

Action	เส้นทางที่ 1	เส้นทางที่ 2	เส้นทางที่ 3
A_1	15 Sec	-	-
A_2	-	15 Sec	-
A_3	-	-	15 Sec
A_4	30 Sec	-	-
A_5	-	30 Sec	-
A_6	-	-	30 Sec

ตารางที่ 2.2 แสดงกำหนดเวลาของไฟเขียวในการปล่อยรถทั้ง 3 เส้นทาง

2.3.1 แผนการจำลองสถานการณ์โดยใช้ *Q-Learning* พิจารณา



รูปที่2.9 ภาพแผนการจำลองสถานการณ์ *Q-Learning*

จากแผนภาพจะอธิบายได้ดังนี้

ช่วงเวลาที่ t_0 ถึง t_1 พบว่า รถที่มาจากถนนสายที่ 1 มีจำนวน 8 คัน รถที่มาจากถนนสายที่ 2 มีจำนวน 1 คัน และรถที่มาจากถนนสายที่ 3 มีจำนวน 1 คัน ที่เวลาเริ่มต้นระบบจะสุ่ม*Action* ขึ้นมาก่อน 1 *Action* (ระบบเลือก Action A_1) ที่ *Action* A_1 จะทำการปล่อยรถที่สายที่ 1 เป็นเวลา t_1-t_0 โดยที่ระบบจะมีการวิเคราะห์และกำหนดค่าแบบ *Q-Learning* นั้นคือ

ที่เวลา t_0 ระบบจะมี

State = 0, 0, 0

Action = A_1

ที่เวลา t_1 ระบบจะมี

State = 0+8-8, 0+1, 0+1 = 0, 1, 1

Action = A_3

Reward = 8, 0, 0

จากนั้นระบบจะทำการเก็บค่า *Reward* เพื่อที่จะนำค่า *Reward* ไปพัฒนาระบบ

ช่วงเวลาที่ t_1 ถึง t_2 พบว่า รถที่มาจากถนนสายที่ 1 มีจำนวน 5 คัน รถที่มาจากถนนสายที่ 2 มีจำนวนเพิ่มขึ้นอีก 1 คัน และรถที่มาจากถนนสายที่ 3 มีจำนวนเพิ่มขึ้นอีก 1 คัน ระบบอยู่ที่ $Action_3$ จะทำการปล่อยรถที่สายที่ 3 เป็นเวลา t_2-t_1 โดยที่ระบบจะมีการวิเคราะห์และกำหนดค่าแบบ *Q-Learning* นั้นคือ

ที่เวลา t_1 ระบบจะมี

State = 0, 1, 1

Action = A_3

ที่เวลา t_2 ระบบจะมี

State = 0+5, 1+1, 1+1-2 = 5, 2, 0

Action = A_2

Reward = 0, 0, 2

จากนั้นระบบจะทำการเก็บค่า *Reward* เพื่อที่จะนำค่า *Reward* ไปพัฒนาระบบ

ช่วงเวลาที่ t_2 ถึง t_3 พบว่า รถที่มาจากถนนสายที่ 1 มีจำนวนเพิ่มขึ้นอีก 7 คัน ไม่รถที่มาจากถนนสายที่ 2 และรถที่มาจากถนนสายที่ 3 มีจำนวนเพิ่มขึ้นอีก 1 คัน ระบบอยู่ที่ $Action_2$ จะทำการปล่อยรถที่สายที่ 2 เป็นเวลา t_3-t_2 โดยที่ระบบจะมีการวิเคราะห์และกำหนดค่าแบบ *Q-Learning* นั้นคือ

ที่เวลา t_2 ระบบจะมี

State = 5, 2, 0

Action = A_2

ที่เวลา t_3 ระบบจะมี

State = 5+7, 2-2, 0+1 = 12, 0, 1

Action = A_4

Reward = 0, 2, 0

จากนั้นระบบจะทำการเก็บค่า *Reward* เพื่อที่จะนำค่า *Reward* ไปพัฒนาระบบ

ช่วงเวลาที่ t_3 ถึง t_4 พบว่า รถที่มาจากถนนสายที่ 1 มีจำนวนเพิ่มขึ้นอีก 3 คัน รถที่มาจากถนนสายที่ 2 มีจำนวนเพิ่มขึ้นอีก 2 คัน และไม่รถที่มาจากถนนสายที่ 3 ระบบอยู่ที่ $Action_4$ จะทำการปล่อยรถที่สายที่ 1 เป็นเวลา t_4-t_3 โดยที่ระบบจะมีการวิเคราะห์และกำหนดค่าแบบ *Q-Learning* นั้นคือ

ที่เวลา t_3 ระบบจะมี

State = 12, 0, 1

Action = A_4

ที่เวลา t_3 ระบบจะมี

State =12+3-15, 0+2, 1+0 = 0, 2, 1

Action = …

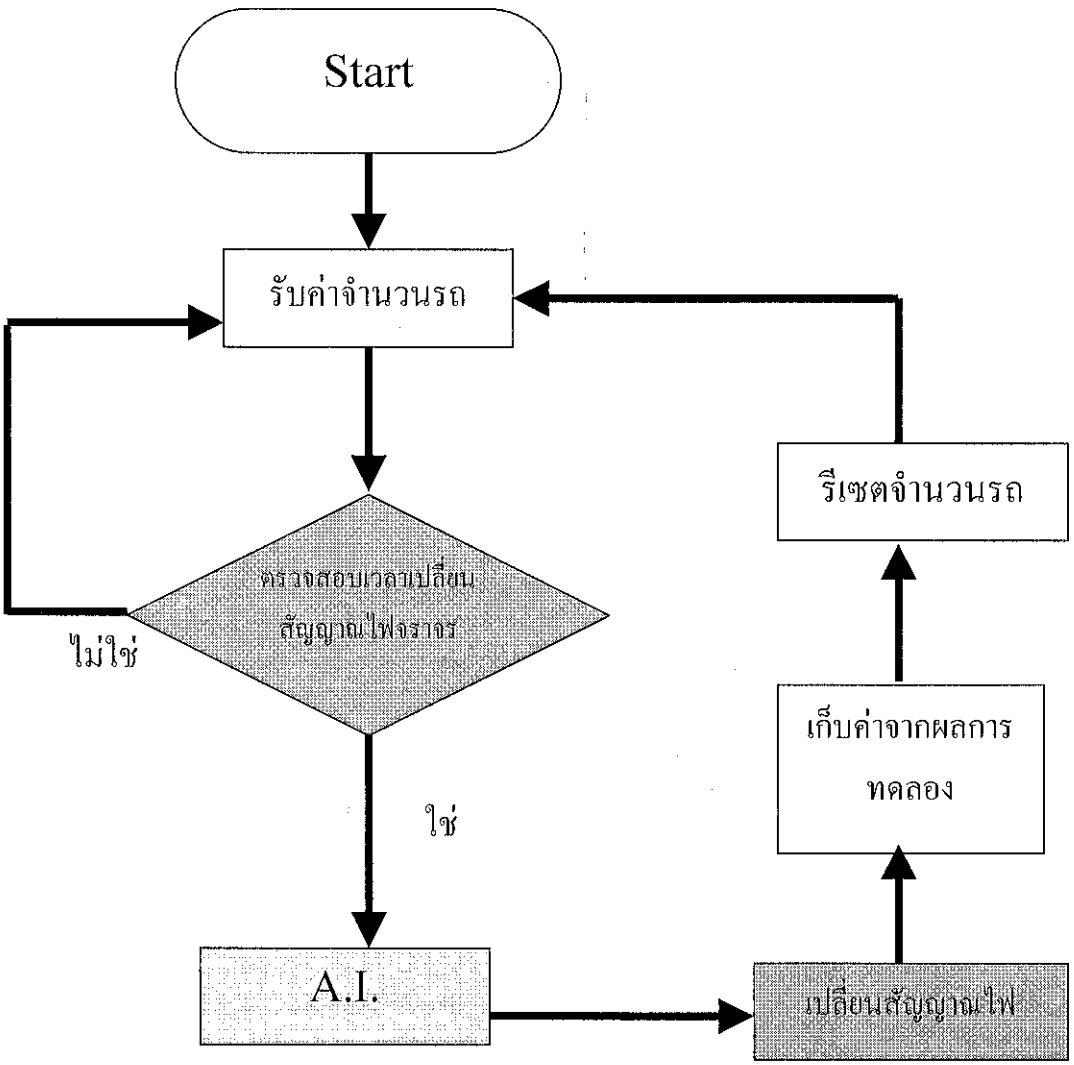
Reward = 15, 0, 0

จากนั้นระบบจะทำการเก็บค่า *Reward* เพื่อที่จะนำค่า *Reward* ไปพัฒนาระบบ

2.4 การเขียนโปรแกรม

2.4.1 การเขียนโปรแกรมจำลองระบบ

การเขียนโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรในโครงงานนี้ คณะผู้จัดทำได้เลือกที่จะใช้ภาษา *C++* ในการเขียนและวิเคราะห์ปัญหา เนื่องจากเป็นภาษาที่ทีการใช้อย่างแพร่หลายสามารถเรียนรู้ เข้าใจได้ง่ายและสามารถประมวลผลในการแก้ไขปัญหาได้อย่างรวดเร็ว ทั้งนี้ในการเขียนตัวโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยโปรแกรมภาษา *C++* เพื่อจำลองระบบดังกล่าวนั้นสามารถเขียนเป็นแผนภูมิเพื่อให้เข้าใจอย่างง่ายๆได้ดังนี้

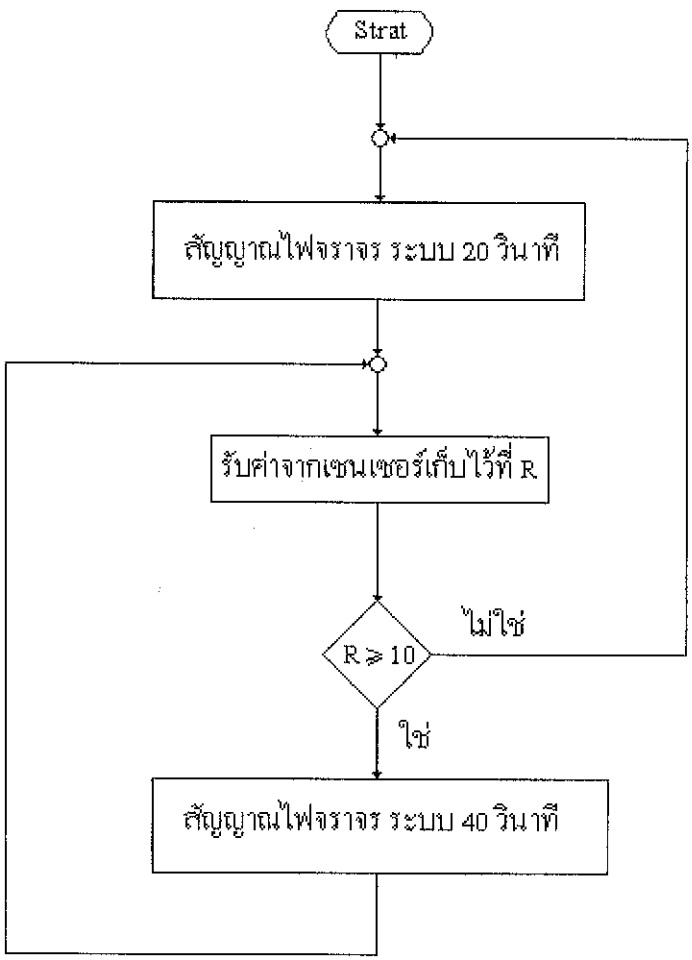


รูปที่2.10 แผนภาพ Flowcharts ในการเขียนโปรแกรมจำลองระบบใน C++

หลังจากนั้นทำการปรับเปลี่ยนในส่วนของโปรแกรมเพิ่มเติม โดยการลบส่วนที่เป็นเอไอและเขียนคำสั่งเพิ่มเติมบางส่วนลงไป ผลก็คือ โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์จะเปลี่ยนกลายเป็น โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ซึ่งมีโครงสร้างพื้นฐานของตัวโปรแกรมแบบเดียวกัน จึงทำให้สามรถเปรียบเทียบผลการรันโปรแกรมทั้งสองได้ง่ายมากยิ่งขึ้น

2.4.2 การเขียนโปรแกรมจำลองระบบในไมโครคอนโทลเลอร์

เนื่องจากการที่เราต้องการนำผลที่ได้จากการรันในโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรไปใช้ทดสอบในสถานการณ์จริงบนท้องถนน เพื่อวัดประสิทธิภาพในการควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ จึงต้องทำการเขียนโปรแกรมลงไมโครคอนโทลเลอร์ด้วยภาษา *Assembly* ซึ่งหลักการทำงานของตัวโปรแกรมที่ทำการเขียนขึ้นมานั้นสามารถอธิบายในรูปแผนภูมิอย่างง่ายดังนี้



รูปที่2.11 แผนภาพ Flowcharts ในการเขียนโปรแกรมจำลองระบบในไมโครคอนโทลเลอร์

บทที่ 3
การทดลอง

3.1 การทดลองในส่วนของโปรแกรมจำลองระบบ

3.1.1 วิธีการทดลอง

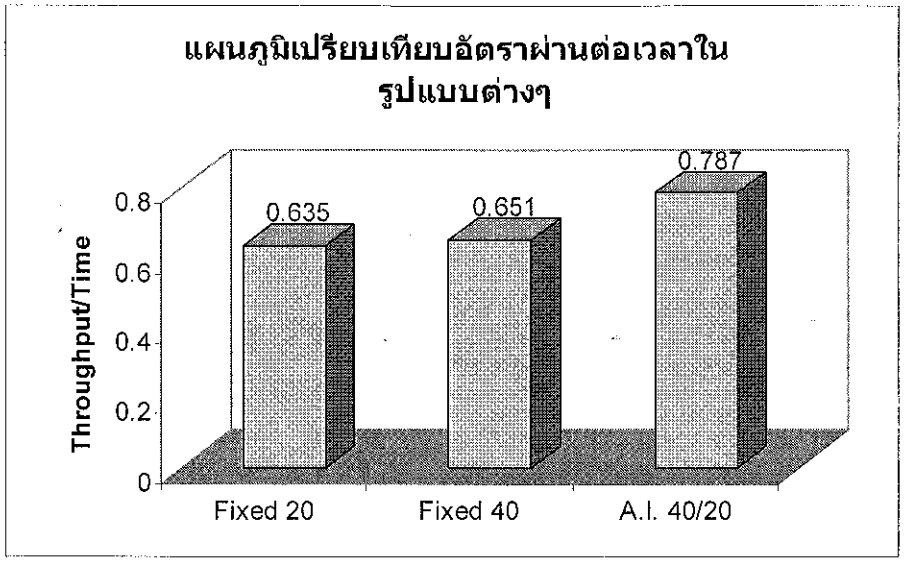
การทดลองโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์และโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาในส่วนของตัวโปรแกรมสามารถทำการศึกษาเพิ่มเติมได้จากภาคผนวก ก. และ ข. ซึ่งจะแสดงการเขียนโปรแกรมภาษา *C++* เพื่อใช้ในการจำลองระบบสัญญาณไฟจราจรทั้งสองรูปแบบเพื่อใช้ในการทดลองโปรแกรมจำลองระบบ โดยในส่วนนี้จะขอกล่าวถึงแต่การตั้งค่าในตัวโปรแกรมเพื่อใช้ในการวัดผลการทดลองโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบปัญญาประดิษฐ์เปรียบเทียบกับโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ซึ่งทั้งสองจะมีโครงสร้างพื้นฐานของโปรแกรมเดียวกันเพื่อความชัดเจนและง่ายต่อการเปรียบเทียบผลการทดลอง โดยที่การทดลองจะแบ่งออกเป็น 2 กรณี ดังนี้ คือ

1. กำหนดให้โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบปัญญาประดิษฐ์ มีค่าเวลา 40 และ 20 วินาที เปรียบเทียบกับโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ซึ่งมีค่าเวลาเป็น 20 และ 40 วินาทีตามลำดับ โดยที่มีอัตรารถเข้าทั้ง 3 เส้นทางเป็น 20, 20 และ 20 คันต่อนาทีตามลำดับ
2. กำหนดให้การตั้งค่าเวลาให้เป็นแบบในกรณีแรก แต่ทำการเปลี่ยนอัตรารถเข้าให้เป็น 20, 10 และ10 คันต่อนาทีตามลำดับทั้ง 3 รูปแบบการทดลองและเปรียบเทียบผลการทดลองที่ได้

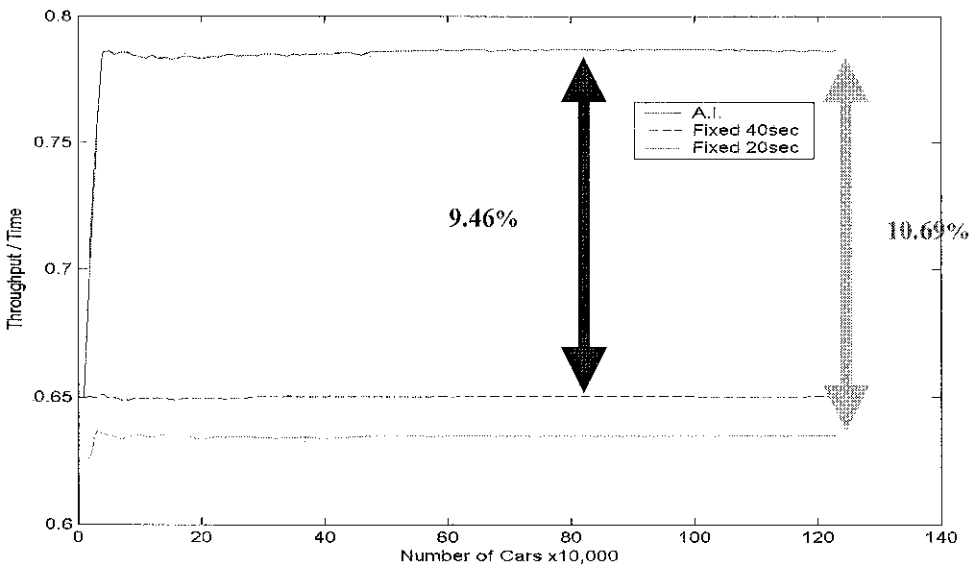
เมื่อทำการตั้งค่าโปรแกรมตามการทดลองที่กำหนดขึ้นมา หลังจากนั้นให้ทำการรันโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจร โดยนำผลการทดลองที่ได้มาเปรียบเทียบกัน เพื่อแสดงให้เห็นถึงประสิทธิภาพของระบบในแบบต่างๆ จากค่าอัตราผ่านรวมต่อเวลาโดยที่ทำการกำหนดค่าตัวแปรอื่นๆในโปรแกรมให้มีค่าคงที่ทุกการทดลอง

3.1.2 ผลการทดลอง

3.1.2.1 กรณีที่ 1

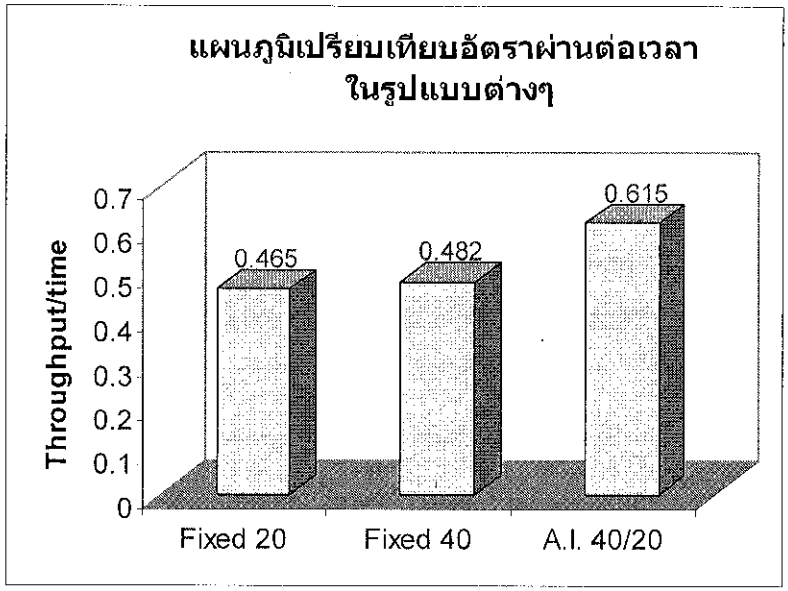


รูปที่ 3.1 ภาพแผนภูมิเปรียบเทียบอัตราผ่านในรูปแบบต่างๆ

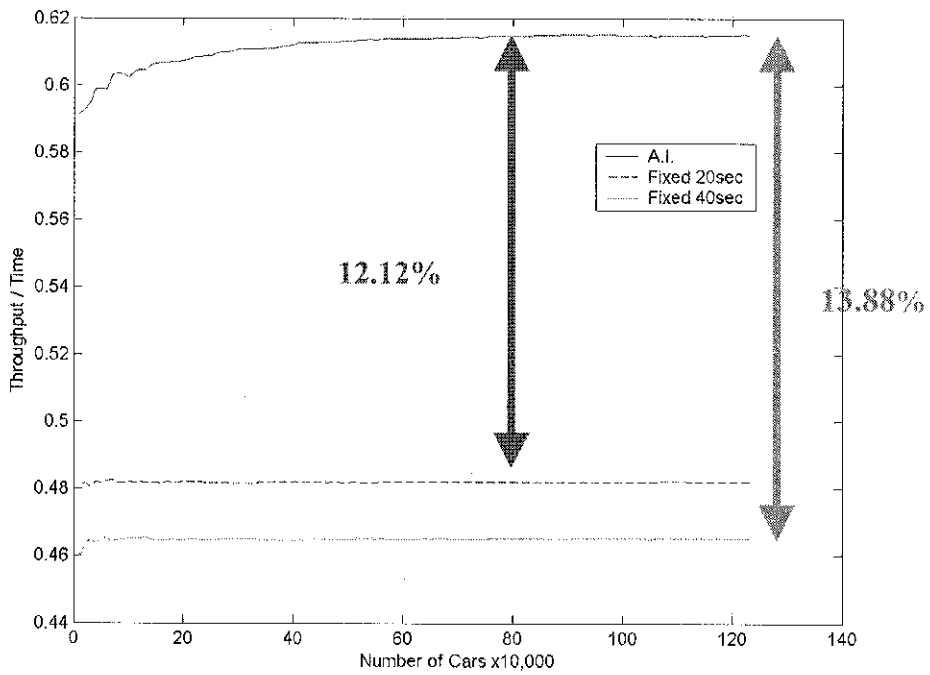


รูปที่ 3.2 ภาพผลการทดลองโปรแกรมจำลองระบบทั้ง 3 รูปแบบ
เมื่อกำหนดอัตรารถเข้า เท่ากับ 20, 20 และ 20 คันต่อนาทีตามลำดับ

3.1.2.2 กรณีที่ 2



รูปที่ 3.3 ภาพแผนภูมิเปรียบเทียบอัตราผ่านในรูปแบบต่างๆ



รูปที่ 3.4 ภาพผลการทดลองโปรแกรมจำลองระบบทั้ง 3 รูปแบบ
เมื่อกำหนดอัตรารถเข้า เท่ากับ 20, 10 และ 10 คันต่อนาทีตามลำดับ

3.1.3 วิเคราะห์ผลการทดลอง

จากผลการทดลองในกรณีที่ 1 เมื่อเราทำการรันโปรแกรมและนำค่าผลการทดลองที่ได้จากการรันโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรทั้งสามมาเพื่อทำการเปรียบเทียบอัตราผ่านรวมต่อเวลา ในรูปที่ 3.2 ของกราฟเชิงเส้นจะพบว่าค่าที่ได้จากโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้นสามารถตอบสนองต่อการทดลองได้ดีกว่า ซึ่งสามารถให้ค่าอัตราผ่านรวมต่อเวลาประมาณ 0.78 โดยที่ค่าที่ได้จากการรันโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาจะมีค่าประมาณ 0.48 และ 0.46 ในส่วนของการตั้งเวลา 40 และ 20 วินาทีตามลำดับ ซึ่งสามารถกล่าวได้ว่าการที่โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ สามารถให้การตอบสนองได้ดีกว่า เนื่องจากการที่ตัวโปรแกรมนั้นสามารถเรียนรู้และทำการเลือกช่วงเวลาที่เหมาะสมต่อการเปลี่ยนแปลงสัญญาณไฟจราจรในระบบด้วยวิธีการของ *Q-Learning* จึงให้ค่าอัตราผ่านรวมต่อเวลา ที่สูงกว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา และเมื่อทำการสังเกตค่าอัตราผ่านรวมต่อเวลา จากจำนวนรถ 2,000,000 คัน โดยการเปรียบเทียบด้วยแผนภูมิแท่งดังรูปที่ 3.1 จะพบว่าค่าที่ได้นั้น ในส่วนของโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ยังคงมีค่ามากกว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาคือ 0.787, 0.651 และ 0.635 ตามลำดับ

จากผลการทดลองในกรณีที่ 2 เมื่อเราทำการเปลี่ยนแปลงอัตรารถเข้าในแต่ละเส้นทางเป็น 20, 10 และ 10 คันต่อนาทีตามลำดับ จะพบว่าค่าอัตราผ่านรวมต่อเวลาจากการนำผลการรันโปรแกรมมาพิจารณาในรูปแบบของกราฟเชิงเส้นดังรูปที่ 3.4 สิ่งที่พบคือ ค่าอัตราผ่านรวมต่อเวลาของโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ ยังคงสามารถตอบสนองต่อระบบได้ดีที่สุดเช่นเดิมเพียงแต่มีค่าที่ได้จะมีค่าลดต่ำลงเมื่อเปรียบเทียบกับการทดลองในกรณีที่ 1 เนื่องจากปริมาณรถที่กำหนดให้เป็นอัตรารถเข้ามีค่าลดลงจากการทดลองในกรณีที่ 1 คือ สามารถให้ค่าอัตราผ่านรวมต่อเวลา ประมาณ 0.61 โดยค่าที่ได้จากโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา นั้นให้ค่าอัตราผ่านรวมต่อเวลา ประมาณ 0.48 และ 0.46 ในส่วนของการตั้งเวลา 40 และ 20 วินาทีตามลำดับและเมื่อทำการดูค่าอัตราผ่านรวมต่อเวลา จากจำนวนรถ 2,000,000 คัน เช่นเดียวกับในกรณีที่ 1 โดยเปรียบด้วยแผนภูมิแท่งดังรูปที่ 3.3 จะพบว่าค่าที่ได้นั้นในส่วนของโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ยังคงมีค่ามากกว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา คือ 0.615, 0.482 และ 0.465 ตามลำดับ จึงสามารถกล่าวได้ว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ด้วยวิธีการ *Q-Learning* นั้นสามารถตอบสนองต่อการทดลองได้ดีกว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ซึ่งเป็นระบบที่ใช้ในการควบคุมสัญญาณไฟจราจรในปัจจุบันมากประมาณ 9.46% - 13.88% ดังที่แสดงให้เห็นในรูปที่ 3.2 และ 3.4 ตามลำดับ

3.1.4 สรุป

โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้นสามารถให้ค่าอัตราผ่านรวมต่อเวลา จากสถานการณ์การทดลองในรูปแบบต่างๆได้ดีกว่าโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา เนื่องจากการที่ตัวโปรแกรมสามารถทำการเรียนรู้จากผลของการกระทำที่เคยได้ทำไป และทำการเลือกช่วงเวลาที่เหมาะสมต่อสถานการณ์ที่เป็นอยู่ในขณะนั้นมากยิ่งขึ้น เพื่อให้ได้ค่าอัตราผ่านรวมต่อเวลาที่ดีที่สุด ตามหลักการของวิธีการ *Q-Learning* โดยที่มีประสิทธิภาพของโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์มีค่าที่มากกว่าประมาณ 9.46% - 13.88% เมื่อเปรียบเทียบกับโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา

3.2 การทดลองในสถานการณ์จริง

3.2.1 วิธีการทดลอง

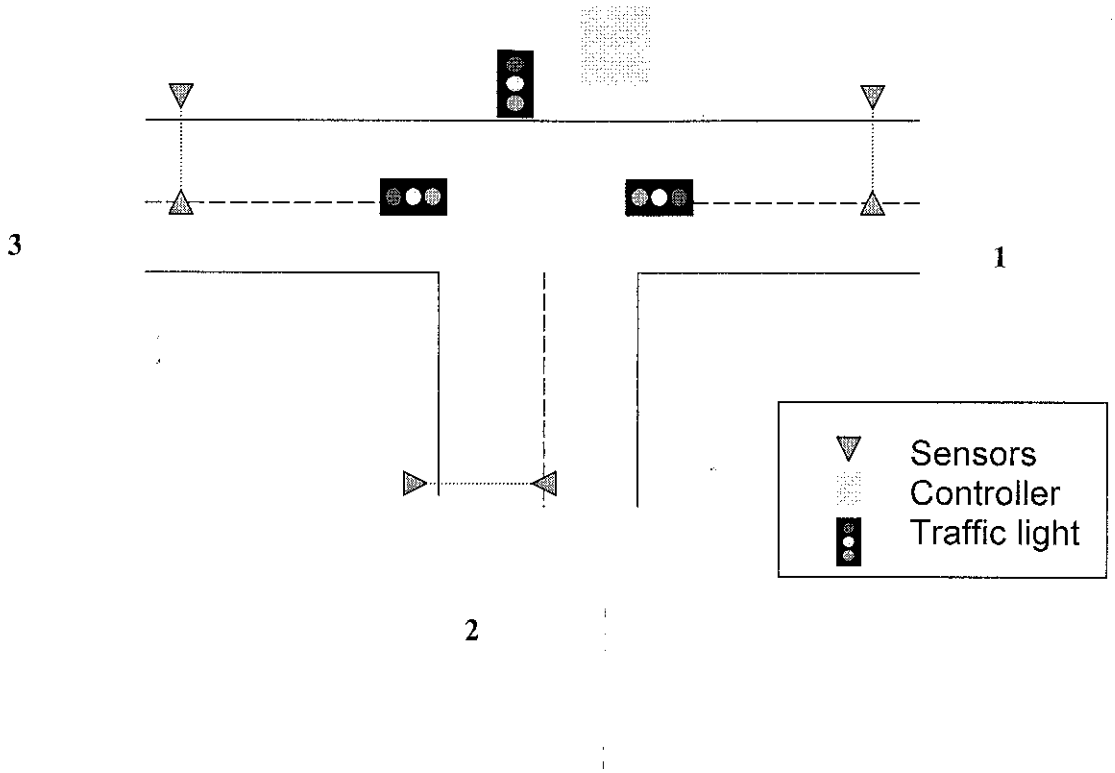
ในการทดลองระบบปัญญาประดิษฐ์ ***(Artificial Intelligence* หรือ *A.I.* หรือ เอไอ)** เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบไฟจราจร เป็นการทดลองจากสถานที่ทำการทดลองจริงในสถานการณ์ที่สมมติขึ้น โดยสถานที่ที่เลือกใช้ในการทดลองคือ บริเวณ 3 แยกประตู 3 ของมหาวิทยาลัยเทคโนโลยีสุรนารี ซึ่งในการทดลองจะแบ่งเป็น 3 การทดลอง เพื่อให้สามารถเปรียบเทียบผลการทดลองได้ ดังนี้

การทดลองที่ 1 ที่ใช้คนในการตัดสินใจในการปล่อยจราจร โดยในการทดลองนี้ได้ทำการขอความเหลือจากพนักงานรักษาความปลอดภัย ให้กระทำการปล่อยจราจรให้

การทดลองที่ 2 การทดลองโดยใช้ระบบไฟจราจรแบบตั้งเวลา(Fixed Time) โดยการควบคุมเวลาในการปล่อยจราจรในแต่ละเส้นทางไว้ที่ 20 วินาที

การทดลองที่ 3 การทดลองใช้ไฟจราจรระบบปัญญาประดิษฐ์ *(Artificial Intelligence* หรือ *A.I.* หรือ เอไอ)

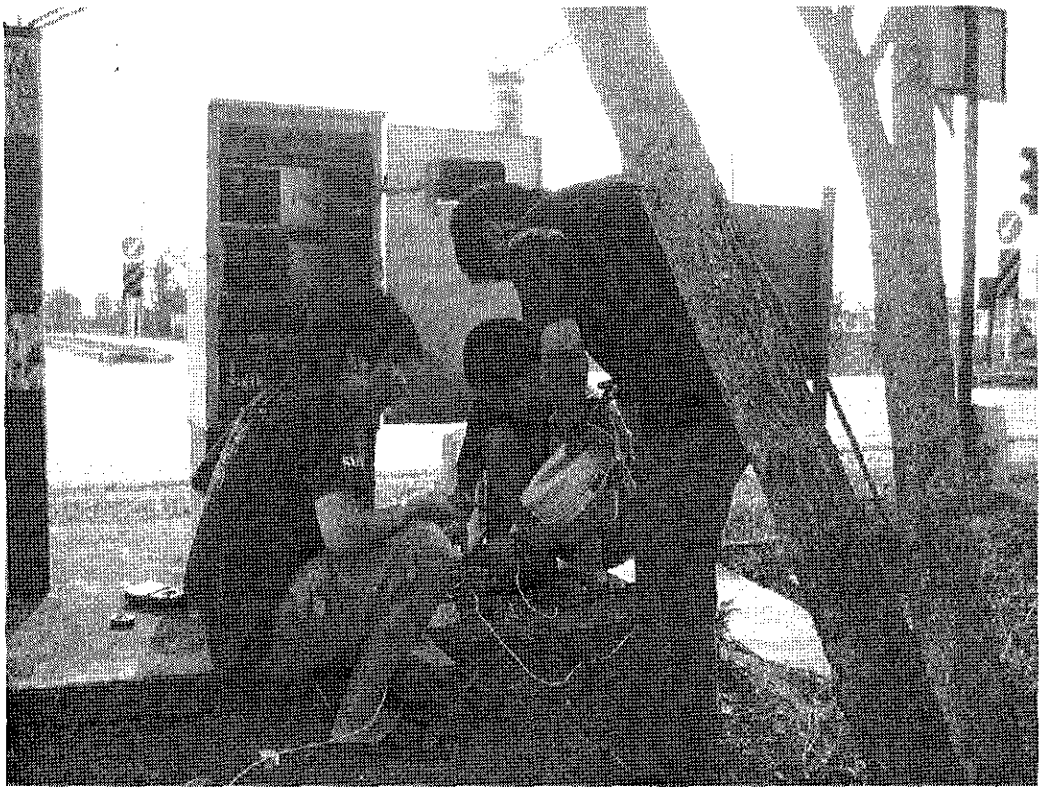
โดยการติดตั้งเซ็นเซอร์ ภาครับ- ภาคส่ง แบบอินฟาเรดทั้ง 3 ชุด ที่ระยะห่างจากบริเวณ 3 แยก ประมาณ 40 เมตรเพื่อใช้ในการตรวจจับปริมาณรถที่ผ่านเข้ามาในแต่ละเส้นทาง และทำการส่งข้อมูลจำนวนรถที่ได้ผ่านสายส่งข้อมูลไปยังตัวไมโครคอนโทลเลอร์เพื่อประมวลผลที่ได้รับทั้ง 3 เส้นทาง แล้วทำการปรับเปลี่ยนไฟสัญญาณจราจรให้เหมาะสมกับสถานการณ์ในขณะนั้น ซึ่งระบบปัญญาประดิษฐ์ที่นำมาใช้นั้นยังไม่สามารถเรียนรู้ได้ตลอดเวลาจากถานการณ์จริง แต่จะเป็นการเขียนโปรแกรมให้ตัวไมโครคอนโทลเลอร์นั้นเลือกช่วงเวลาที่เหมาะสม จากที่ทำการกำหนดในการทดลองแทน โดยในที่นี้เวลาที่ทำการเลือกคือ 40 และ 20 วินาที ในส่วนของโปรแกรมภาษา *Assembly* สำหรับตัวไมโครคอนโทลเลอร์ที่ใช้ในการทดลองนั้น สามารถทำการศึกษาเพิ่มเติมได้ในส่วนของภาคผนวก ข. สำหรับโปรแกรมระบบการทดลองในสถานการณ์จริงโดยใช้ระบบไฟจราจรแบบตั้งเวลา และภาคผนวก ค. สำหรับโปรแกรมระบบการทดลองในสถานการณ์จริงโดยใช้ระบบไฟจราจรด้วยระบบปัญญาประดิษฐ์ ซึ่งการติดตั้งอุปกรณ์ที่ใช้ในการทดลองจากสถานที่จริงสามารถดูได้จากภาพจำลองแสดงการติดตั้งอุปกรณ์ในการทดลองจากสถานที่จริงดังกล่าว



รูปที่ 3.5 ภาพจำลองแสดงการติดตั้งอุปกรณ์ในการทดลองจากสถานที่จริง

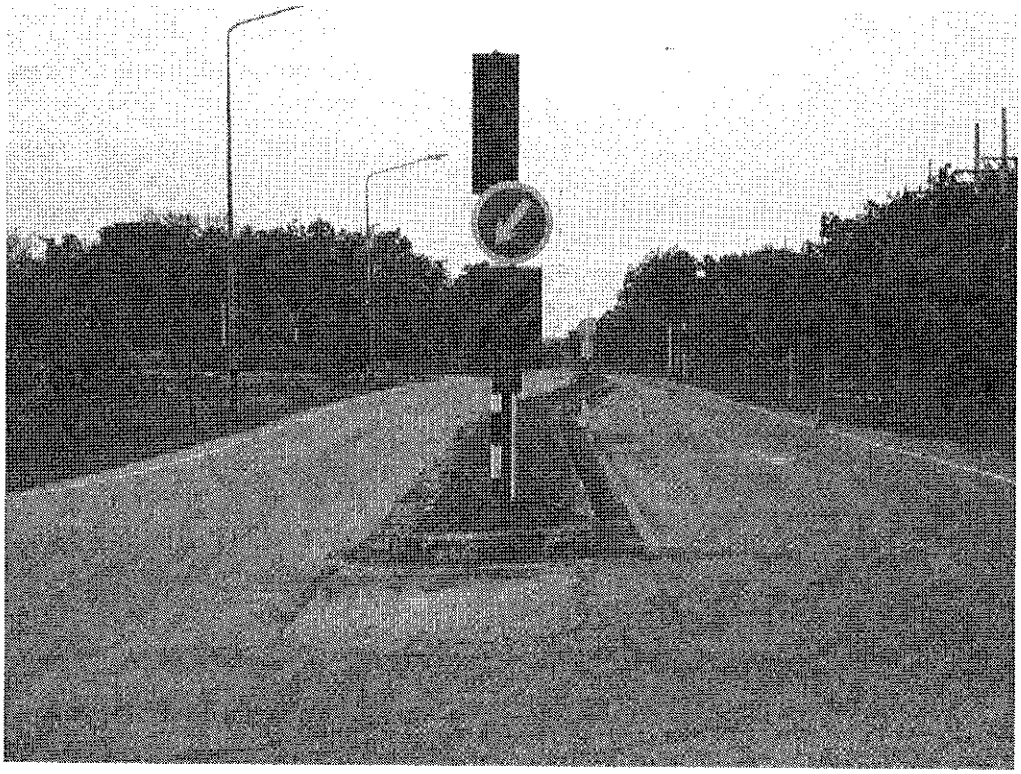
ทั้งนี้เนื่องจากเป็นการทดลองในสถานที่จริงจากสถานการณ์จำลองจากบริเวณแยกเรียนรวม ซึ่งผู้เข้าร่วมการทดลอง มี ประมาณ 60 คน จากวิศวกรรมโทรคมนาคม รุ่นที่ 11 และ 12 จึงทำการเลือกอัตรารถที่เข้ามาสู่บริเวณแยกให้เหมาะสมกับผู้เข้าร่วมการทดลอง จากตารางนับรถซึ่งได้จากการสังเกต ณ บริเวณทางแยกหน้าอาคารเรียนรวม อัตราที่ได้คือ 40 , 15 และ 5 คันต่อนาที โดยจะใช้อัตราดังกล่าวในทุกๆการทดลอง ด้วยการให้มีผู้ควบคุมการปล่อยรถในแต่ละเส้นทางเพื่อรักษาอัตรารถให้คงที่ แต่จะทำการปรับเปลี่ยนความเร็วในการขับขี่ คือ 20 , 30 และ 40 กิโลเมตรต่อชั่วโมงตามลำดับ เพื่อให้สามารถเปรียบเทียบผลการทดลองได้มากยิ่งขึ้น โดยทำการวัดข้อมูลจากปริมาณรถที่สามารถผ่านสัญญาณไฟจราจรหรืออัตราผ่านของรถในแต่ละเส้นทางและทั้งหมดทุกเส้นทางรวมกัน เป็นเวลา 7 นาที ต่อ ความเร็วต่างๆ และทำการบันทึกผลการทดลอง

รูปที่ 3.6 ภาพแยกการทดลองประตู 3

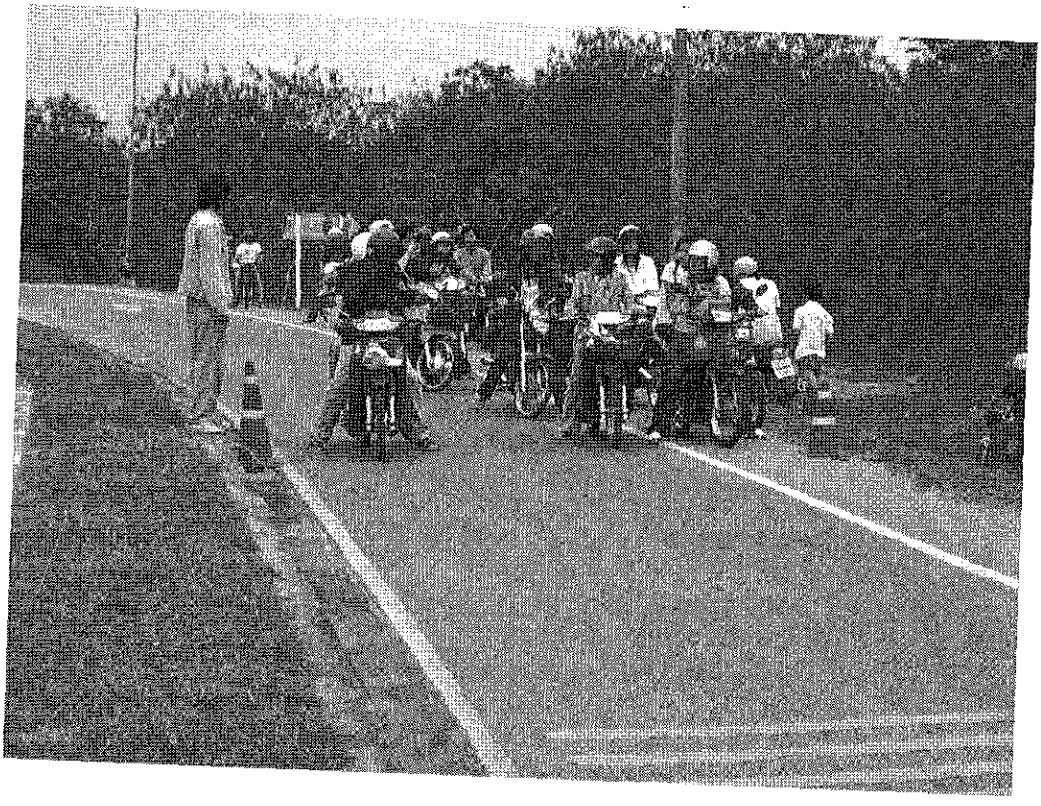


รูปที่3.7 ภาพแสดงการติดตั้งตัวไมโครคอนโทลเลอร์

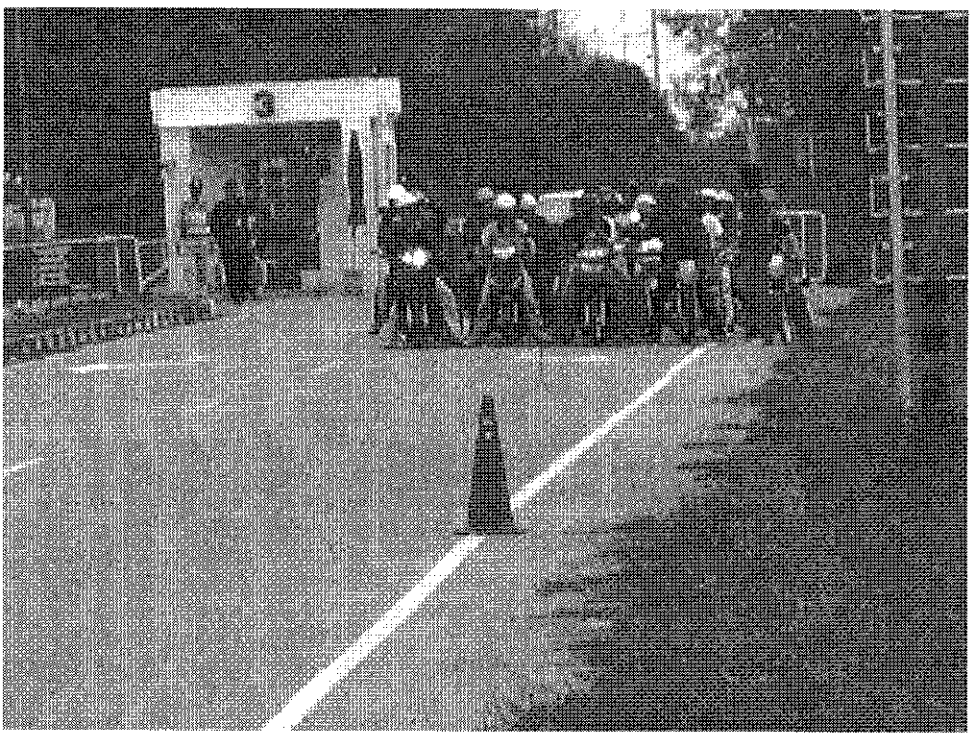
รูปที่3.8 ภาพการติดตั้งสัญญาณไฟจราจร



รูปที่ 3.9 ภาพบริเวณแยกประตู 3 ก่อนการทดลอง

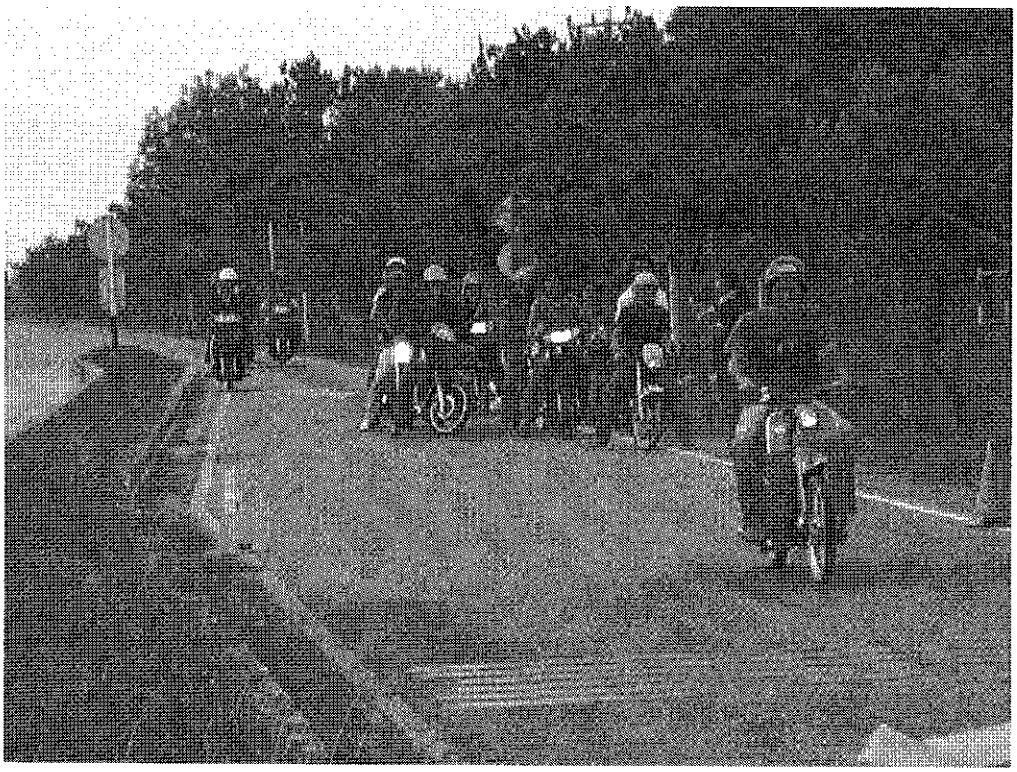


รูปที่ 3.10 ภาพแสดงการควบการปล่อยรถ เส้นทางที่ 1



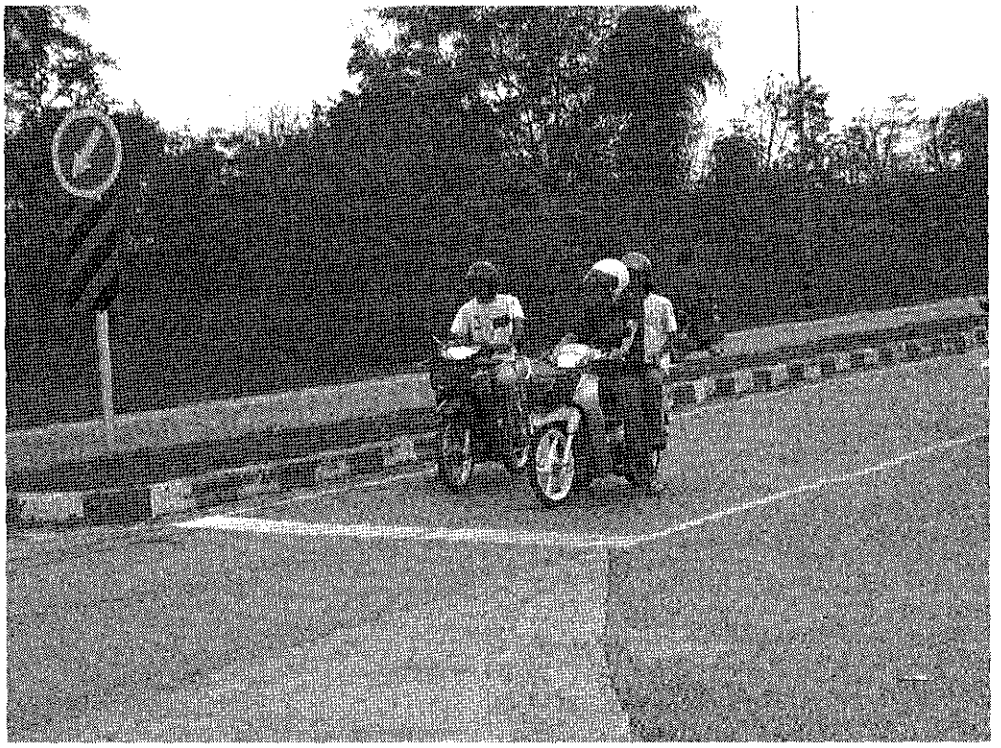
รูปที่ 3.11 ภาพแสดงการควบการปล่อยรถ เส้นทางที่ 2

รูปที่ 3.12 ภาพแสดงการควบการปล่อยรถ เส้นทางที่ 3



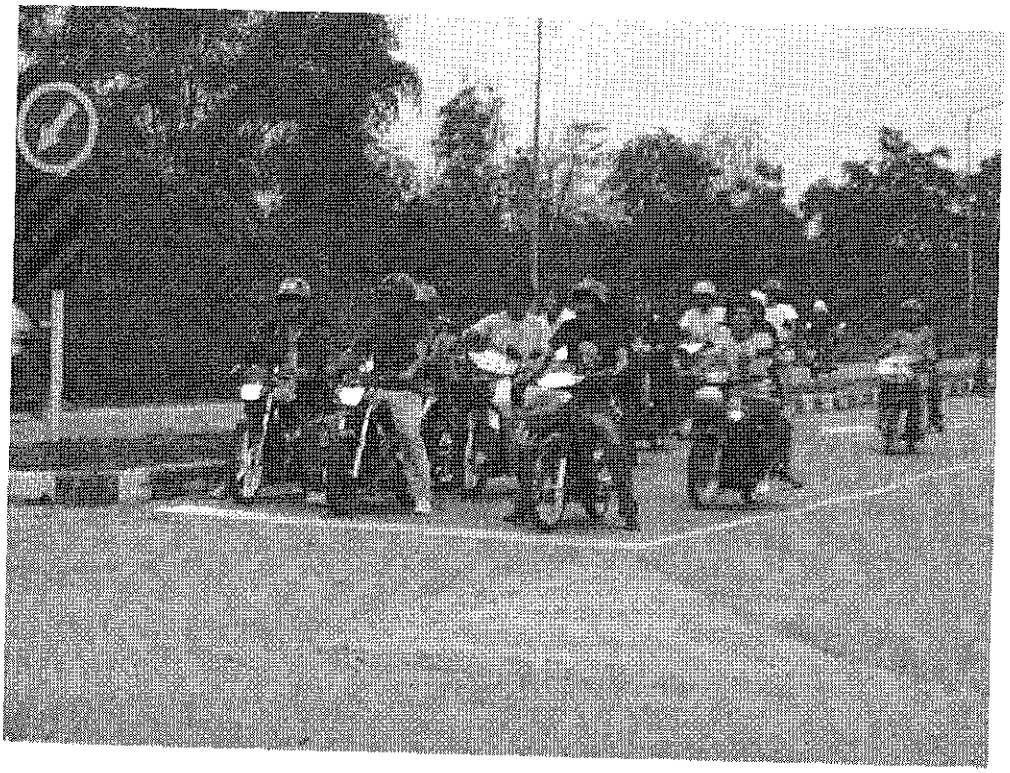
รูปที่3.13 ภาพแสดงการปล่อยรถตามจำนวนที่ระบุในการทดลอง

รูปที่ 3.14 ภาพการทดลองโดยใช้มนุษย์ในการตัดสินใจ



รูปที่ 3.15 ภาพแสดงการปฏิบัติตามกฎจราจรในการทดลอง

รูปที่ 3.16 ภาพแสดงพฤติกรรมในการใช้เส้นทาง



รูปที่ 3.17 ภาพกลุ่มผู้ร่วมทดสอบปฏิบัติตามไฟจราจรที่ปรากฏ

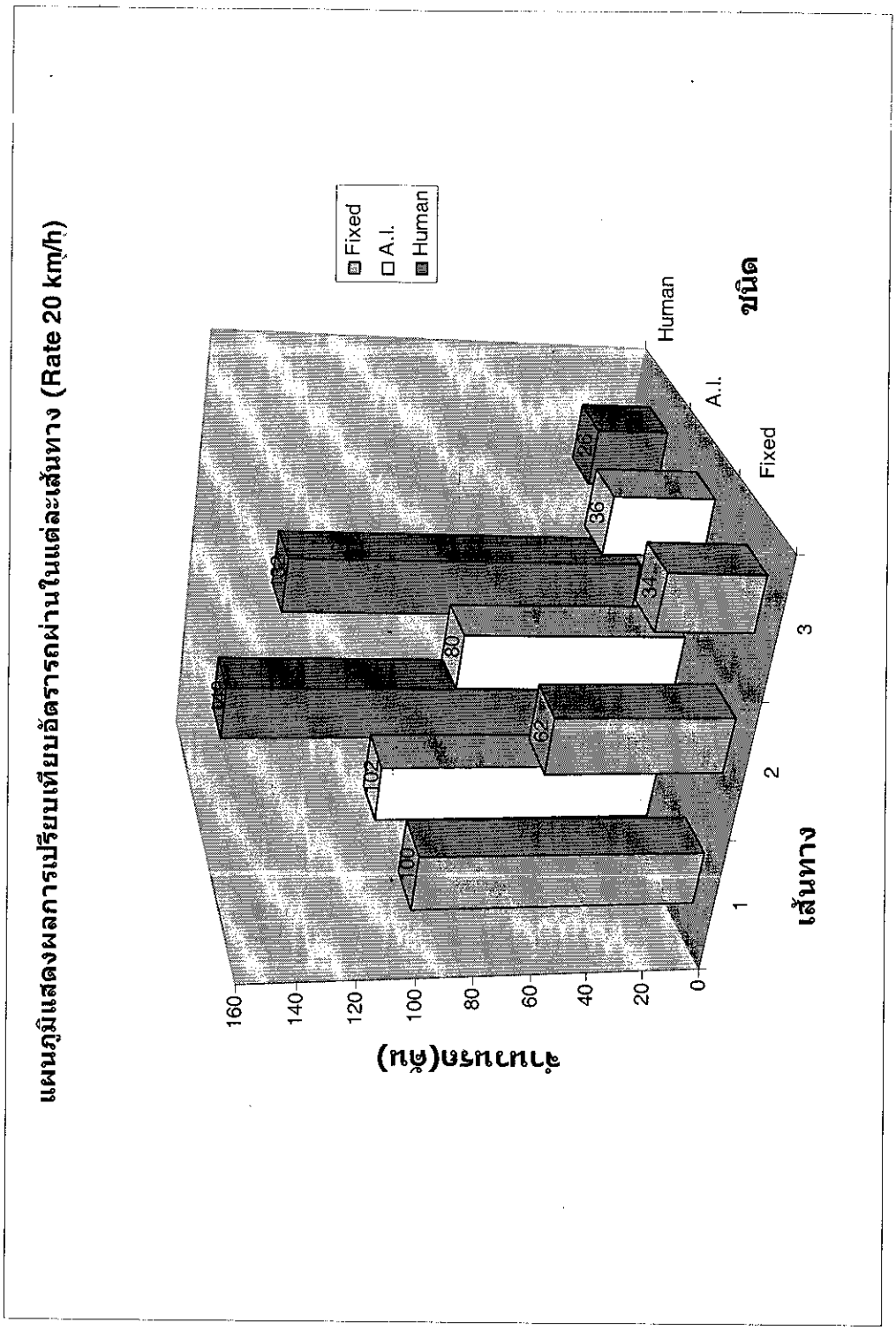
รูปที่ 3.18 ภาพแสดงการเก็บข้อมูลและจับเวลา

3.2.2 ผลการทดลอง

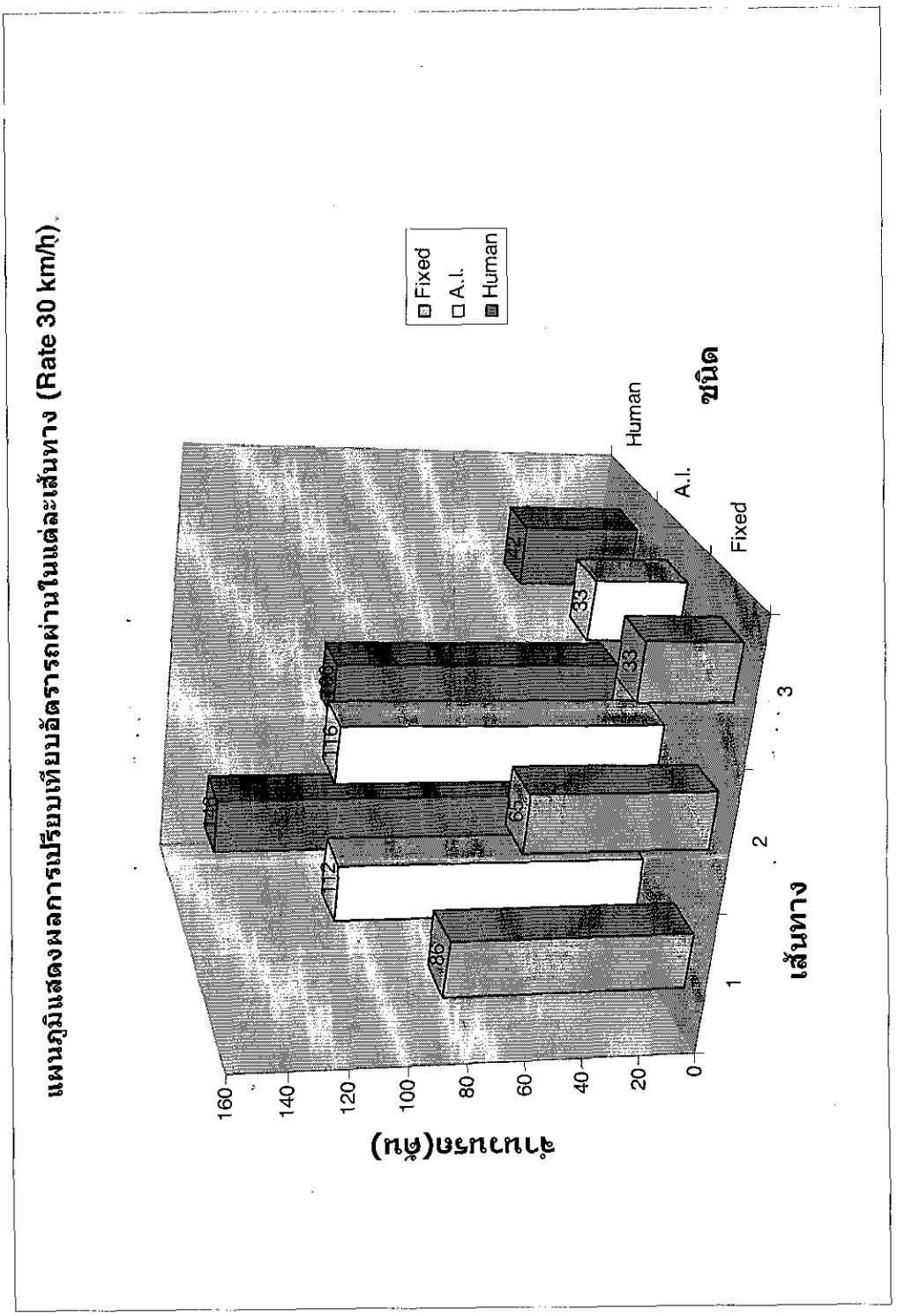
ตารางที่ 3.1 แสดงผลการนับจำนวนรถที่ผ่านสัญญาณไฟจราจรรวม

Type	Rate (km/h)		
	20	**30**	**40**
Fixed	196	184	174
A.I.	218	261	233
Human	307	294	278

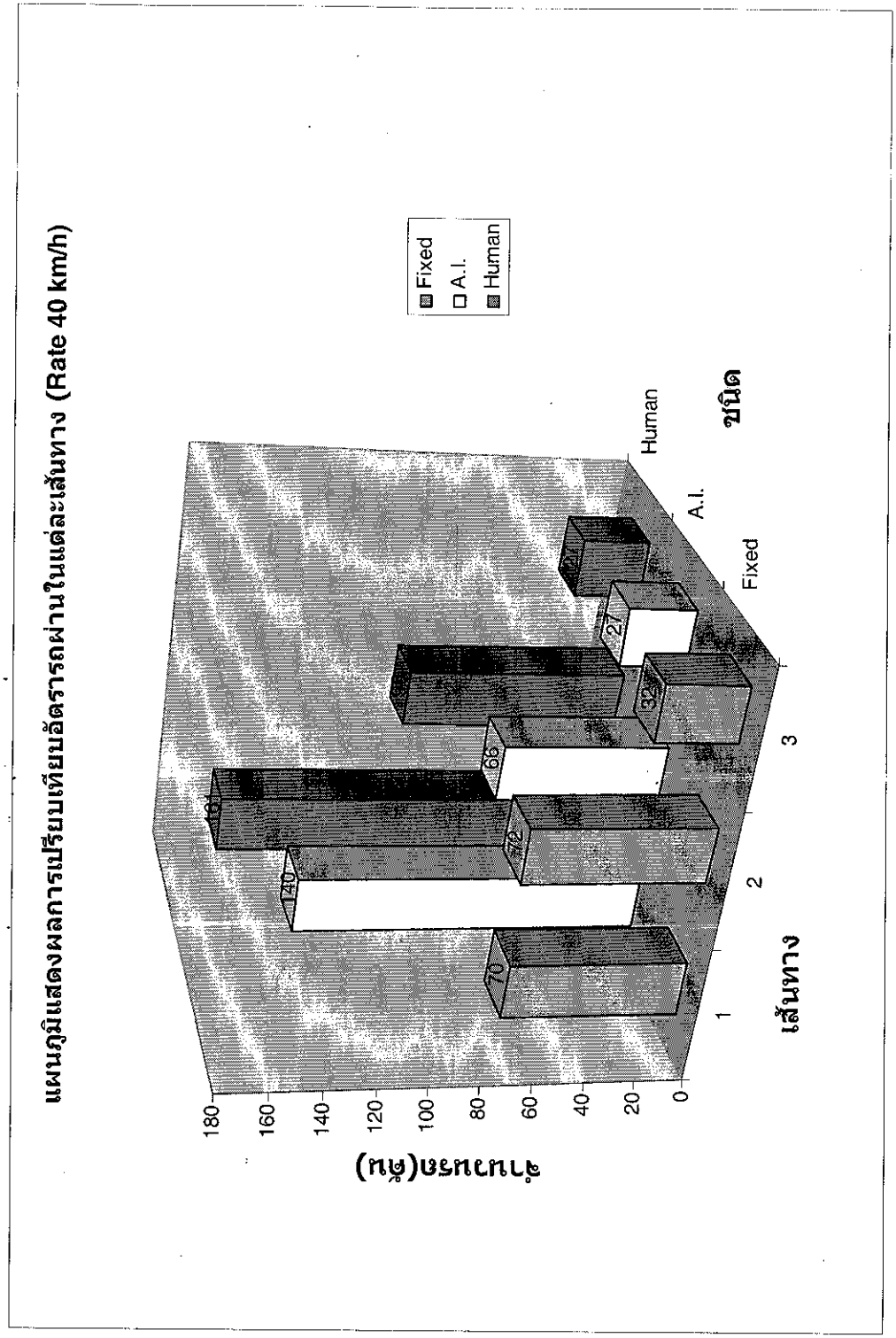
หมายเหตุ สามารถดูบันทึกข้อมูลผลการนับรถทั้งหมดได้จาก ภาคผนวก ในส่วนของตาราง ผลการทดลองจากสถานที่จริง



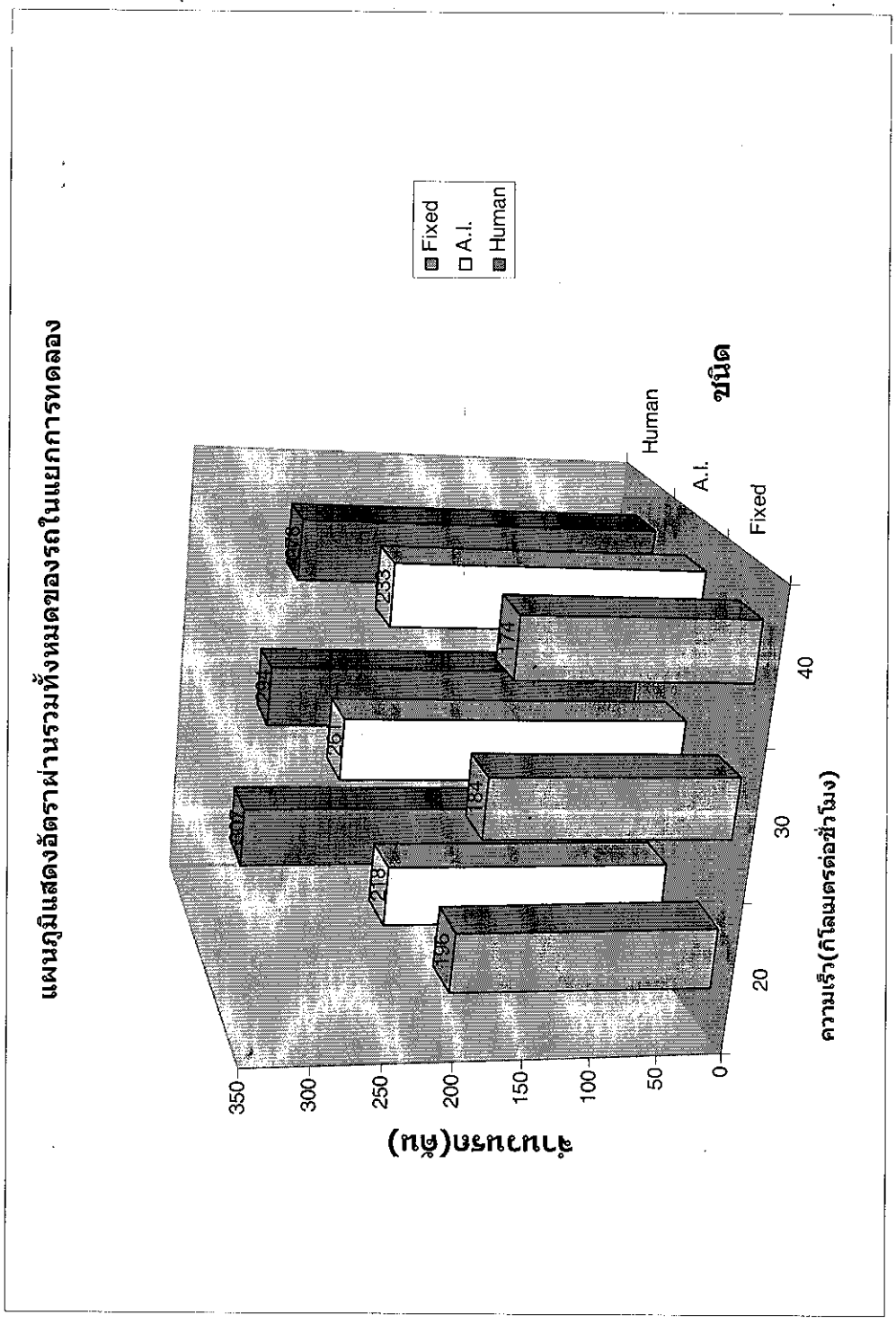
รูปที่ 3.19 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง เมื่อความเร็วในการทดลองเป็น 20 กิโลเมตรต่อชั่วโมง



รูปที่ 3.20 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง
เมื่อความเร็วในการทดลองเป็น 30 กิโลเมตรต่อชั่วโมง



รูปที่ 3.21 ภาพแผนภูมิแสดงผลการเปรียบเทียบอัตรารถผ่านในแต่ละเส้นทาง เมื่อความเร็วในการทดลองเป็น 40 กิโลเมตรต่อชั่วโมง



รูปที่ 3.22 ภาพแผนภูมิแสดงอัตราผ่านรวมทั้งหมดของรถในแยกการทดลองที่ความเร็วต่างๆ

3.2.3 วิเคราะห์ผลการทดลอง

จากตารางที่ 3.1 แสดงผลการนับจำนวนรถที่ผ่านสัญญาณไฟจราจรรวมจากการนำจำนวนรถที่ผ่านในแต่ละช่วงเวลามารวมกัน จะเห็นได้ว่าปริมาณรถจักยานยนต์ที่สามารถผ่านได้ในแต่ละเส้นทางที่ความเร็วต่างๆต่อช่วงเวลาในการทดลอง ในส่วนของการใช้คนตัดสินใจจะมีอัตราผ่านปริมาณมากที่สุด เนื่องจากบริเวณแยกดังกล่าวไม่ซับซ้อนจนเกินไปและปริมาณความหนาแน่นของการจราจรบริเวณแยกที่กำหนดเป็นเพียงกลุ่มทดสอบกลุ่มเล็กๆ จึงสามารถทำการตัดสินใจได้อย่างรวดเร็วในการปล่อยรถจักยานยนต์ในแต่ละเส้นทาง ในส่วนของระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ผลการทดลองที่ได้ จะพบได้ว่าที่ความเร็วต่างๆ ในแต่ละเส้นทางและทุกเส้นทางรวมกัน ดังรูปที่ 3.19 และ 3.20 จะมีปริมาณรถจักยานยนต์ที่สามารถผ่านสัญญาณไฟจราจรต่อช่วงเวลาหรืออัตราผ่านของรถจักยานยนต์มากกว่า เมื่อเปรียบเทียบกับระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาเป็นส่วนใหญ่ ยกเว้นที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง ดังรูปที่ 3.21 ค่าของระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลานั้นจะมีค่าอัตราผ่านของรถมากกว่าในส่วนของระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ อาจเป็นเพราะเป็นการทดลองสุดท้ายที่ทำมาต่อเนื่องกว่า 2 ชั่วโมง และกลุ่มผู้ร่วมทดลองเริ่มเหนื่อยล้ากับการทดลองทำให้ผลที่ได้อาจมีค่าผิดพลาดจากความเป็นจริง แต่เมื่อสังเกตจากปริมาณรถจักยานยนต์ที่สามารถผ่านสัญญาณไฟจราจรต่อช่วงเวลาหรืออัตราผ่านของรถจักยานยนต์ทั้งหมดรวมกันจากรูปที่ 3.22 ในส่วนของระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้นยังคงมีค่าอัตราผ่านของรถมากกว่าระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา แสดงให้เห็นว่าในการควบคุมสัญญาณไฟจราจรนั้น ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์สามารถทำได้ดีกว่าในภาพโดยรวมเนื่องจากสามารถทำการเลือกช่วงเวลาให้เหมาะสมต่อสถานการณ์ในขณะที่เป็นอยู่ให้เหมาะสมกับช่วงเวลาที่ทำการกำหนดให้ โดยมีประสิทธิภาพมากกว่าระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาประมาณ 5.31% - 17.30% และหากเปรียบเทียบระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์กับการใช้มนุษย์ในการตัดสินใจ จากรูปที่ 3.22 ผลที่ได้จะพบว่าระบบที่ใช้มนุษย์ทำการตัดสินใจจะสามารถทำได้ดีกว่าระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ประมาณ 5.94% - 16.95% ในส่วนของภาพรวมและในแต่ละเส้นทาง แต่มีค่าบางค่าที่ระบบปัญญาประดิษฐ์สามารถทำได้ใกล้เคียงกันหรือมีค่ามากกว่า เช่น รูปที่ 3.19 และ รูปที่ 3.21 อาจมองได้ว่าเนื่องจากระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ที่ใช้ในการทดลองครั้งนี้เป็นเพียงต้นแบบที่ใช้ในการทดสอบการทดลองเพื่อใช้ในการแก้ปัญหาสัญญาณไฟจราจร ยังไม่สามารถทำการเรียนรู้ได้ตลอดเวลา เนื่องจากปัญหา อุปสรรค และข้อจำกัดบางประการในการทำโครงงาน

3.2.4 สรุป

จากการทดลองในสถานที่จริง จะพบว่าระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์สามารถทำงานได้อย่างดีและมีประสิทธิภาพ โดยที่พิจารณาจากการให้ค่าอัตราผ่านที่สูงกว่าระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา ประมาณ 5.31% - 17.30% และสามารถทำงานได้มีประสิทธิภาพใกล้เคียงกันกับการใช้มนุษย์ในการตัดสินใจโดยมีประสิทธิภาพที่น้อยกว่าเมื่อพิจารณาจากการค่าอัตราผ่านประมาณ 5.94% - 16.95% ซึ่งเราจะพบได้ว่าการนำระบบปัญญาประดิษฐ์เข้ามาเพื่อใช้ในการแก้ไขปัญหาสัญญาณไฟจราจรที่มีอยู่เดิมนั้น ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์สามารถทำงานได้เป็นอย่างดี และให้ผลการทดลองอยู่ในระดับที่น่าพอใจอย่างยิ่ง

บทที่ 4
บทสรุป และข้อเสนอแนะ

4.1 สรุปผลโครงงาน

จากการทดลองระบบปัญญาประดิษฐ์ ***(Artificial Intelligence* หรือ *A.I.)*** เพื่อทำการวิเคราะห์และแก้ไขปัญหาระบบไฟสัญญาณจราจร ทำให้เราทราบถึงวิธีการทำงานของระบบปัญญาประดิษฐ์โดยการใช้เทคนิคที่เรียกว่า การเรียนรู้แบบรีอินฟอร์สเมนท์ *(Reinforcement Learning* หรือ *RL)* ด้วยวิธีการ *Q-Learning* จากการเขียนโปรแกรมภาษา *C++* โดยการศึกษาปัญหาของระบบสัญญาณไฟจราจรที่มีอยู่ในปัจจุบัน ซึ่งระบบปัญญาประดิษฐ์สามารถแก้ไขปัญหาดังกล่าวในขั้นต้นได้เป็นอย่างดี โดยในส่วนของการรันโปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้น สามารถให้การตอบสนองต่อระบบที่ทำการออกแบบในการทดลองได้เป็นอย่างดี เมื่อนำระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้นมาใช้ในการทดลองผลในสถานที่จริง ผลการทดลองเป็นที่น่าพอใจจากการให้ค่าอัตราผ่านของรถต่อช่วงเวลาอยู่ในเกณฑ์ที่ดี ซึ่งแสดงให้เห็นว่าระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์นั้นสามารถทำงานได้มีประสิทธิภาพใกล้เคียงกันกับการมนุษย์ในการตัดสินใจ และมีประสิทธิภาพสูงกว่าระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลาที่ใช้อยู่ในปัจจุบันเป็นอย่างมาก รวมทั้งเรายังสามารถพัฒนาเพื่อให้ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์มีประสิทธิภาพที่ดียิ่งๆขึ้นไป

4.2 สิ่งที่ได้รับจากโครงงาน

1. ได้รับความรู้เกี่ยวกับเทคนิคที่เรียกว่า Reinforcement Learning
2. หลักการวิเคราะห์ข้อมูลก่อนทำโครงงาน ทำให้ทราบถึงปัญหาของระบบจราจรทั่วไป
3. ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมด้วยภาษา C++
4. ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมด้วยภาษา Assembly
5. ได้รับความรู้ในการใช้งาน ไมโครคอนโทลเลอร์และนำไปใช้ในสถานการณ์จริง
6. ได้รับความรู้ความสามารถในการเลือกใช้วัสดุอุปกรณ์ต่างๆที่เหมาะสมในโครงงาน
7. สามารถแก้ไขปัญหาและอุปสรรคเฉพาะหน้าได้
8. ทำให้รู้จักหลักการทำงานร่วมกับผู้อื่น
9. สามารถนำความรู้ที่ได้รับไปใช้งาน กับโครงงานอื่นๆได้

4.3 ปัญหาและอุปสรรคระหว่างการทดลอง

1. ต้องใช้เวลาในการศึกษาสภาพจราจรรวมถึงการเก็บข้อมูลจากสถานที่จริง
2. ต้องใช้เวลาในการศึกษาภาษา C++ และ Assembly เพิ่มเติม
3. เกิดการลองผิดลองถูกในการเลือกใช้เซ็นเซอร์ ทำให้เกิดความล่าช้าในโครงงาน

4. งบประมาณในการทำโครงงานไม่เพียงพอต่อการใช้ในแก้ปัญหาในโครงงาน จึงสามารถทำโครงงานแก้ไขปัญหาได้เพียงขั้นต้น

5. ตัวควบคุมสัญญาณไฟจราจรแบบ PLC ชำรุดไม่สามารถใช้งานได้

6. ใช้เวลาในปรับปรุงโปรแกรมทั้งในส่วนของภาษา ซี และ Assembly

7. ต้องใช้เวลาในการเตรียมการทดลองให้พร้อมสำหรับการทดลองจากสถานที่จริง เนื่องจากต้องขอความร่วมมือจากผู้เข้าร่วมการทดลองโครงงาน

8. เนื่องจากต่อนทำการทดลองเป็นเวลากลางวันไฟสัญญาณจราจรที่ทำขึ้นมีความสว่างไม่เพียงพอจึงต้องใช้คนช่วยบอกสัญญาณไฟ

รูปที่ 4.1 ภาพการใช้คนช่วยในการบอกสัญญาณไฟจราจร

4.4 ปัญหาหลังการทดลอง

1. ระบบเอไอที่สร้างขึ้นไม่ได้เป็นแบบเรียนรู้ได้ตลอดเวลาจึงไม่สามารถแก้ไขปัญหาจากสถานการณ์จริงได้อย่างสมบูรณ์

2. จำนวนรถที่สามารถตรวจสอบได้จาก เซ็นเซอร์ นั้นไม่สามารถตรวจสอบพฤติกรรมของผู้ใช้ยานพาหนะบนท้องถนนได้อย่างสมบูรณ์

3. ข้อจำกัดเกี่ยวกับสถานที่ใช้ในการทดลอง กลุ่มผู้ทดสอบ ซึ่งเป็นบริเวณแยกที่ไม่มีความซับซ้อนทางจราจรมาก่ทำให้ระบบที่ใช้คนปล่อยรถแต่ละเส้นทางนั้นสามารถทำได้ดีกว่า และเนื่องจากกลุ่มผู้ร่วมทดสอบเป็นเพียงกลุ่มเล็กๆเมื่อเทียบกับการจราจรบนท้องถนนจริง รวมทั้งความเหนื่อยล้าจากการทดลองตลอด 2 ชั่วโมง ทำให้มีบางกลุ่มยุติการร่วมการทดลอง ส่งผลถึงผลการทดลองในส่วนของเอไออาจมีค่าน้อยกว่าความเป็นจริง

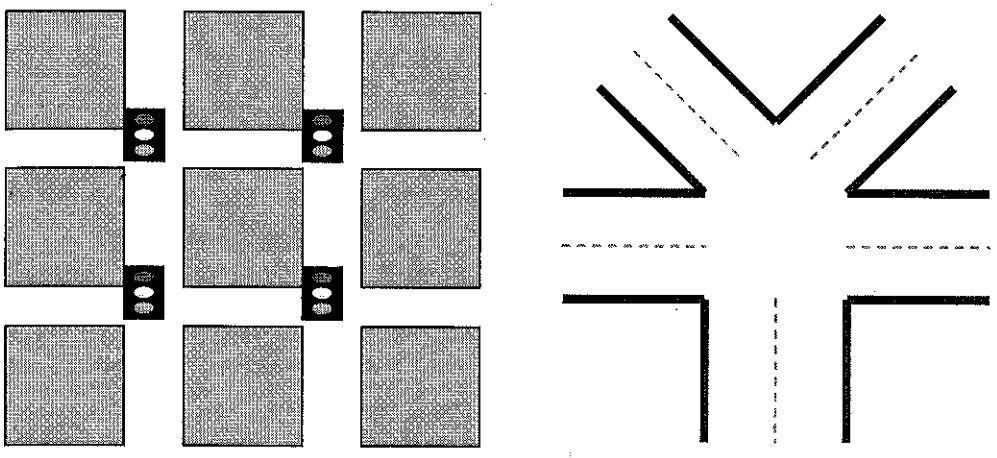
4.5 แนวทางการแก้ไข

1. พัฒนาระบบเอไอให้เป็นแบบที่สามารถเรียนรู้ได้ตลอดเวลาจากสถานการณ์จริงซึ่งจะสามารถแก้ไขปัญหาได้อย่างถูกต้องและแม่นยำมากยิ่งขึ้น

2. ใช้เซ็นเซอร์ แบบ Loop ที่เป็นเซ็นเซอร์ที่ผู้ทำโครงงานพบว่าดีที่สุดในการตรวจสอบจำนวนยานพาหนะ และพัฒนาให้สามารถทำการตรวจจับจำนวนยานพาหนะได้ถูกต้อง แน่นอน จากการศึกษาผลงานของ อ.ดร. รังสรรค์ ทองทา ได้ทำการทดลองและวิจัยไว้ซึ่งแสดงในส่วนของภาคผนวก

4.6 ข้อเสนอแนะ

จากโครงงานนี้จะสามารถพัฒนาโปรแกรมให้สามารถทำการแก้ไขปัญหาจราจรในระบบที่มีขนาดใหญ่มากยิ่งขึ้นหรือมีความซับซ้อนมากยิ่งขึ้นได้ ดังเช่นการที่เรานำระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ไปทำการควบคุมสัญญาณไฟจราจรบริเวณแยกต่างๆ ที่มีอยู่รวมๆกันในบริเวณที่เราต้องการ โดยที่แต่ละแยกมีความสัมพันธ์ซึ่งกันและกัน หากว่าเราทำการควบคุมสัญญาณไฟจราจรในการปล่อยปริมาณรถในแต่ละเส้นทางไม่ดีเพียงพอ จะทำให้เกิดปัญหาการจราจรติดขัดมากยิ่งขึ้น รวมถึงบริเวณที่มีแยกมากกว่าสี่แยกขึ้นไป ซึ่งมีการใช้เส้นทางการเดินรถที่ซับซ้อนมากยิ่งขึ้น ดังรูปที่แสดง



รูปที่ 4.2 ภาพแสดงบริเวณที่แยกต่างๆอยู่รวมกันและแยกที่มีความซับซ้อน

โดยที่การควบคุมสัญญาณไฟจราจรด้วยระบบแบบตั้งเวลาที่ใช้อยู่ในปัจจุบันนี้ ไม่สามารถให้ค่าอัตราผ่านที่สูงที่สุดในแต่ละช่วงเวลาต่างๆในบริเวณหรือแยกดังกล่าวได้ ซึ่งหากมีการพัฒนานำเอาระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์ ให้สามารถทำการเรียนรู้ได้ตลอดเวลาผ่านเครือข่ายต่างๆ ตัวระบบปัญญาประดิษฐ์จะสามารถทำการวิเคราะห์และแก้ไขปัญหาได้อย่างถูกต้องมากยิ่งขึ้นสำหรับการเลือกเปลี่ยนสัญญาณไฟจราจรแต่ละครั้ง ทำให้เราได้อัตราผ่านรวมของรถในระบบของบริเวณหรือแยกดังกล่าวมีค่าที่สูงมากยิ่งขึ้นในแต่ละช่วงเวลา

บรรณานุกรม

Q-learning: Off-Policy TD Control, http://www-anw.cs.umass.edu/rlr/,1997

Reinforcement Learning Repository, http://www.cs.ualberta.ca/~sutton/book/6/node6.html

H. M. Deitel, C++ HOW TO PROGRAM 3rd Edition, Deitel & Associates, INC. 2001

ประวัติผู้เขียน

นาย นพดล ลัยนันท์ เกิดเมื่อวันที่ 26 กันยายน พ.ศ. 2526 ภูมิลำเนาอยู่ที่ 122 หมู่ 2 ต.สันกลาง อ.สันกำแพง จ.เชียงใหม่ 50130 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียน มงฟอร์ต วิทยาลัย แผนกมัธยม เมื่อปีพ.ศ.2544 ปัจจุบันเป็นนักศึกษาปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นาย ธัชกร ไกรธนานันต์ เกิดเมื่อวันที่ 3 ธันวาคม พ.ศ. 2527 ภูมิลำเนาอยู่ที่ 711 หมู่ 5 ต.จอหอ อ.เมือง จ.นครราชสีมา 30310 การศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียน บุญวัฒนา เมื่อปีพ.ศ.2544 ปัจจุบันเป็นนักศึกษาปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นาย พงศกร คงสมบูรณ์ เกิดเมื่อวันที่ 1 กันยายน พ.ศ. 2526 ภูมิลำเนาอยู่ที่ 1/1 หมู่ 1 ต.ห้วยขมิ้น อ.หนองแค จ.สระบุรี 18230 การศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียน นครนายกวิทยาคม เมื่อปีพ.ศ.2544 ปัจจุบันเป็นนักศึกษาปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก

ภาคผนวก ก.

โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์

```
# include <iostream.h>      // Precompiled header
# include <stdlib.h>
# include <math.h>
# include <iomanip.h>
# include <cstdlib>
# include <fstream.h>
//# include <graphics.h>

int max_episode = 3000000;

const int CLASS_SIZE = 3 ;
const int num_action  = 6 ;
const int num_state   = 8 ;
const int level = 2;

int i,j, k, a ;
int CLASS_arrival ;

double randunif01 ;
double alpha ;
//double min_X ;
double Time = 0 ;
double shortgreen = 20 ;//5;//10 ;// 15 ;
double longgreen  = 40 ;//10; //20 ;//30 ;
double   Q[ level ][ level ][ level ][ num_action ]  ;
int      Numvisit[ level ][ level ][ level ][ num_action ]  ;
double   arrivalrate[ CLASS_SIZE ] = { 20, 20, 20 } ; //{ 1, 20 , 1 } ; //{ 20, 20, 20 } ; //{ 50, 2,
10 } ; //  //number of vehicles arrival per min
int   X[ CLASS_SIZE ] ;
double t[ CLASS_SIZE ] ;
```

```
double  ActionSet[ num_action ] = { longgreen, longgreen, longgreen, shortgreen, shortgreen,
shortgreen } ;   //{ AC   , B,   C,   AC   , B,  C  } // 0, 1, 2, 3, 4, 5
int STATE[ CLASS_SIZE ] = { 0 , 0 , 0 } ;
int action = 0 ;
int next_i, next_j, next_k, next_a ;
double stepsize    = 0.0005 ;
double discountrate = 0.9 ;
double e_greedy = 0.5;
double REWARD ;
double Accum_throughput = 0,  Av_throughput ;
int step ;
int A = 0 , B = 1 , C = 2 ;
double min_t ;
/*****************************************************************
            Declarations
 *****************************************************************/
double Randunif01();
void  Initialization() ;
double Randpois_RV(double randunif01 , double  alpha);
void  Find_CLASS_arrival();
void  Findnext_a() ;
void  FindAction_egreedy();
void  Writeresults() ;
void  WriteQresults() ;
/*****************************************************************
            Main Program
*****************************************************************/
void main()
{
        Initialization() ;
        for ( i = 0; i< CLASS_SIZE; i++ )
                arrivalrate[ i ] = arrivalrate[ i ]/60 ; //number of vehicles arrival per sec
```

```
double Time_changelights = Time + ActionSet[ action ] ;  // seconds
for ( step = 0; step<max_episode ; step++ )
{
        Find_CLASS_arrival() ;
        //cout<< Time <<  setw(20) << Av_throughput<< setw(20) <<
CLASS_arrival<< setw(10) << STATE[ 0 ] << setw(10) << STATE[ 1 ] << setw(10) <<
STATE[ 2 ] <<endl;
        Time += min_t ;          //check for time to change lights
        if ( Time_changelights > Time )
        {
                X[ CLASS_arrival ]++ ;
                if ( X[ CLASS_arrival ] >=10 )
                        STATE[ CLASS_arrival ] = 1 ;
                else
                        STATE[ CLASS_arrival ] = 0 ;
        }
        else // its time to change lights
        {
                double delay = 2 ; // seconds
                int num_car = 0 ;
                i = STATE[ 0 ] ;
                j = STATE[ 1 ] ;
                k = STATE[ 2 ] ;
                a = action ;

                if ( action == 0 || action == 3 )  // A and C
                {
                        int maxcar = int(ActionSet[ action ]/delay) ;

                        if ( maxcar < X[ A ] )
                                num_car = maxcar ;
                        else
```

```
                num_car = X[ A ] ;

        X[ A ] = X[ A ] - num_car ;
        REWARD += double(num_car) ;

        if ( maxcar < X[ C ] )
                num_car = maxcar ;
        else
                num_car = X[ C ] ;

        X[ C ] = X[ C ] - num_car ;

        STATE[ A ] = 0 ;
        if ( X[ A ] >= 10 )
                STATE[ A ] = 1 ;

        STATE[ C ] = 0 ;
        if ( X[ C ] >= 10 )
                STATE[ C ] = 1 ;
}

if ( action == 1 || action == 4 ) // B
{       int maxcar = int(ActionSet[ action ]/delay) ;

        if ( maxcar < X[ B ] )
                num_car = maxcar ;
        else
                num_car = X[ B ] ;
        X[ B ] = X[ B ] - num_car ;

        STATE[ B ] = 0;
        if ( X[ B ] >= 10 )
```

```
                STATE[ B ] = 1 ;
}

if ( action == 2 || action == 5 )  // C
{       int maxcar = int(ActionSet[ action ]/delay) ;

        if ( maxcar < X[ C ] )
                num_car = maxcar ;
        else
                num_car = X[ C ] ;
        X[ C ] = X[ C ] - num_car ;
        STATE[ C ] = 0 ;
        if ( X[ C ]  >= 10 )
                STATE[ C ] = 1 ;
}

REWARD += double(num_car) ;  //throughput
cout<<setw( 20 )<< REWARD ;

// update Q function

next_i = STATE[ 0 ] ;
next_j = STATE[ 1 ] ;
next_k = STATE[ 2 ] ;

// get next_a = argmax Q[ next_i ][ next_j ][ next_k ][ a_ ]

Findnext_a();
double stepsized = stepsize/( 1 + double(step)/10000000) ;

//cout<<setw( 20 )<< stepsized << setw( 20 ) << discountrate  ;
```

```
                    Q[ i ][ j ][ k ][ a ] = Q[ i ][ j ][ k ][ a ] +
                                stepsized * ( REWARD +
discountrate*Q[ next_i ][ next_j ][ next_k ][ next_a ] - Q[ i ][ j ][ k ][ a ]) ;

                    Numvisit[ i ][ j ][ k ][ a ]++ ;
                    /// get new action from e greedy
                    FindAction_egreedy();

                    // update time to change lights
                    Time_changelights = Time + ActionSet[ action ] ;

                    // Average throughput
                    Accum_throughput += REWARD ;
                    Av_throughput = Accum_throughput/Time ;

                    Writeresults() ;
                }// else
        }// for step

        WriteQresults() ;
        system("pause");
}// end main

/*********************************************************************
            Subfunctions
 *********************************************************************/
void Initialization()
{
        int level = 2;
        X[ 0 ] = 0 ;
        X[ 1 ] = 0 ;
        X[ 2 ] = 0 ;
```

```
        action = 0 ;
        for ( i = 0; i< level;  i++ )
        {
                for ( j = 0; j< level; j++ )
                {
                        for ( k = 0; k<level; k++ )
                        {
                                for ( a = 0; a<num_action; a++ )
                                {
                                        Q[ i ][ j ][ k ][ a ] = 0;//Randunif01() ;
                                        Numvisit[ i ][ j ][ k ][ a ] = 0 ;
                                }// for a
                        }// for k
                }// for j
        }// for i
}
/*****************************************************************
                Random number generators
 *****************************************************************/

double Randunif01()
{
        randunif01 = ((double) rand())/RAND_MAX;
        return (randunif01);
}
double Randpois_RV(double randunif01 , double  alpha)
{
   double  x;
   x  = -(1/alpha)*log(1-randunif01);
   return (x);
}
```

```
/*****************************************************************
                        Find Next Event
*****************************************************************/
void Find_CLASS_arrival()
{
        min_t  = 100000000000;
        for ( i = 0; i < CLASS_SIZE ; i++ )
        {
                alpha = arrivalrate[ i ];
                randunif01    = Randunif01();
                t[ i ]        = Randpois_RV(randunif01,alpha);
                if ( min_t > t[ i ] )
                {
                        min_t = t[ i ];
                        CLASS_arrival  = i;
                }

        }
}

/*****************************************************************
                        Findnext_a
*****************************************************************/
void Findnext_a()
{
        double max_Q = -1000000000 ;
        for ( int a_ = 0 ; a_< num_action ; a_++ )
        {
                if ( max_Q < Q[ next_i ][ next_j ][ next_k ][ a_ ])
                {
                        max_Q *= Q[ next_i ][ next_j ][ next_k ][ a_ ] ;
                        next_a = a_ ;
```

```
                    }
                }
}
/*************************************************************************
                        FindAction_egreedy
*************************************************************************/
void FindAction_egreedy()
{
    double proba, max_prob = 0 ;

    if ( Randunif01() < e_greedy ) // ( Randunif01() < double(step)/double(max_episode) )
        action = next_a ;
    else
    {
        for ( int a_ = 0 ; a_ < num_action ; a_++ )
        {
            proba =  Randunif01() ;
            if ( max_prob < proba  )
            {
                max_prob  = proba ;
                action = a_ ;
            }
        }
    }
    //cout<< setw( 20) << action<< endl;
}

/*************************************************************************
                        Write to files
*************************************************************************/
void Writeresults()
{
```

```
        if ( step%1000 == 1 )
        {
                ofstream outClientFile("Throughput.txt", ios::app );
                if ( !outClientFile ) { // overload ! operator
                                cerr<<" File could not be opened"<<endl;
                                exit(1);
                        }

                outClientFile<< step <<  setw(20) << Time << setw(20) << Av_throughput <<
endl;

                cout<< step << setw( 20) << Time <<  setw(20) << Av_throughput<<endl ;
        }

}

/*************************************************************************
                        Write to files
*************************************************************************/
void WriteQresults()
{

                ofstream outClientFile1("Q.txt", ios::app );
                if ( !outClientFile1 ) { // overload ! operator
                                cerr<<" File could not be opened"<<endl;
                                exit(1);
                        }

                for ( i = 0; i<level; i++ )
                {
                        for ( j = 0; j< level; j++ )
                        {
```

```
                    for ( k = 0; k< level; k++ )
                    {
                        for ( a = 0; a< num_action; a++ )
                        {
                            outClientFile1<< i<< setw(20) << j <<
setw(20) << k<<  setw(20)<< a<<  setw(20) << Q[ i ][ j ][ k ][ a ] << setw(20) <<
Numvisit[ i ][ j ][ k ][ a ] <<endl;
                        }// for a
                    }// for k
                }// for j
            }// for i


}
```

ภาคผนวก ข.

โปรแกรมจำลองระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา

```
# include <iostream.h>      // Precompiled header
# include <stdlib.h>
# include <math.h>
# include <iomanip.h>
# include <cstdlib>
# include <fstream.h>
int max_episode = 1000000;

const int CLASS_SIZE = 3 ;
const int num_action  = 3 ;
const int num_state   = 8 ;
const int level = 2;

int i,j, k, a ;
int CLASS_arrival ;

double randunif01 ;
double alpha ;
double Time = 0 ;
double shortgreen = 40 ;//5;//10 ;// 15 ;

double   Q[ level ][ level ][ level ][ num_action ]  ;
int      Numvisit[ level ][ level ][ level ][ num_action ]  ;
double   arrivalrate[ CLASS_SIZE ] = { 40, 20, 20 } ;  //number of vehicles arrival per min
int   X[ CLASS_SIZE ] ;
double t[ CLASS_SIZE ] ;
double   ActionSet[ num_action ] = { shortgreen, shortgreen, shortgreen } ;
                                        //{ AC   , B,    C  }
        // 0                    1                   2
```

```
int STATE[ CLASS_SIZE ] = { 0 , 0 , 0 } ;
int action = 0 ;
int next_i, next_j, next_k, next_a ;
double stepsize     = 0.0005 ;
double discountrate = 0.9 ;
double e_greedy = 0.5;
double REWARD ;
double Accum_throughput = 0,  Av_throughput ;
int step ;
int A = 0 , B = 1 , C = 2 ;
double min_t ;
/*************************************************************************
                Declarations
 *************************************************************************/
double Randunif01();
void   Initialization() ;
double Randpois_RV(double randunif01 , double  alpha);
void   Find_CLASS_arrival();
void   Findnext_a() ;
void   FindAction_egreedy();
void   Writeresults() ;
void   WriteQresults() ;
/*************************************************************************
                Main Program
*************************************************************************/
void main()
{
        Initialization() ;
        for ( i = 0; i< CLASS_SIZE; i++ )
                arrivalrate[ i ] = arrivalrate[ i ]/60 ; //number of vehicles arrival per sec
        double Time_changelights = Time + ActionSet[ action ] ;  // seconds
```

```
for ( step = 0; step<max_episode ; step++ )
{
        Find_CLASS_arrival() ;
        //cout<< Time <<  setw(20) << Av_throughput<< setw(20) <<
CLASS_arrival<< setw(10) << STATE[ 0 ] << setw(10) << STATE[ 1 ] << setw(10) <<
STATE[ 2 ] <<endl;
        Time += min_t ;
        //check for time to change lights

        if ( Time_changelights > Time )
        {
                X[ CLASS_arrival ]++ ;

                if ( X[ CLASS_arrival ] >=10 )
                        STATE[ CLASS_arrival ] = 1 ;
                else
                        STATE[ CLASS_arrival ] = 0 ;
        }
        else // its time to change lights
        {
                double delay = 2 ; // seconds
                int num_car = 0 ;

                i = STATE[ 0 ] ;
                j = STATE[ 1 ] ;
                k = STATE[ 2 ] ;
                a = action ;
                REWARD = 0 ;

                if ( action == 0 )  // A and C
                {
                        int maxcar = int(ActionSet[ action ]/delay) ;
```

```
            if ( maxcar < X[ A ] )
                    num_car = maxcar ;
            else
                    num_car = X[ A ] ;

            X[ A ] = X[ A ] - num_car ;
            REWARD += double(num_car) ;

            if ( maxcar < X[ C ] )
                    num_car = maxcar ;
            else
                    num_car = X[ C ] ;

            X[ C ] = X[ C ] - num_car ;

            STATE[ A ] = 0 ;
            if ( X[ A ] >= 10 )
                    STATE[ A ] = 1 ;

            STATE[ C ] = 0 ;
            if ( X[ C ] >= 10 )
                    STATE[ C ] = 1 ;
    }

    if ( action == 1 ) // B
    {       int maxcar = int(ActionSet[ action ]/delay) ;

            if ( maxcar < X[ B ] )
                    num_car = maxcar ;
            else
                    num_car = X[ B ] ;
            X[ B ] = X[ B ] - num_car ;
```

```
                STATE[ B ] = 0 ;
                if ( X[ B ] >= 10 )
                        STATE[ B ] = 1 ;
        }

        if ( action == 2 ) // C
        {       int maxcar = int(ActionSet[ action ]/delay) ;

                if ( maxcar < X[ C ] )
                        num_car = maxcar ;
                else
                        num_car = X[ C ] ;
                X[ C ] = X[ C ] - num_car ;
                STATE[ C ] = 0 ;
                if ( X[ C ] >= 10 )
                        STATE[ C ] = 1 ;
        }

        REWARD += double(num_car) ;  //throughput

        if ( REWARD == 0 )
                int check = 0;

        cout<<Time<< setw(20) << REWARD << setw(20) <<
Accum_throughput<< setw(20) <<Av_throughput<< endl;

        // update time to change lights
        Time_changelights = Time + ActionSet[ action ] ;

        // Average throughput
        Accum_throughput += REWARD ;
        Av_throughput = Accum_throughput/Time ;
```

```
                Writeresults() ;
            }// else

        }// for step  */

        //WriteQresults() ;

        system("pause");

}// end main

/*********************************************************************
            Subfunctions
 *********************************************************************/

void Initialization()
{
        int level = 2;

        X[ 0 ] = 0 ;
        X[ 1 ] = 0 ;
        X[ 2 ] = 0 ;

        action = 0 ;

        for ( i = 0; i< level;  i++ )
        {
            for ( j = 0; j< level; j++ )
            {

                for ( k = 0; k<level; k++ )
                {
```

```
                    for ( a = 0; a<num_action; a++ )
                    {
                        Q[ i ][ j ][ k ][ a ] = 0;//Randunif01() ;

                        Numvisit[ i ][ j ][ k ][ a ] = 0 ;
                    }// for a
                }// for k

            }// for j

        }// for i

}
/*********************************************************************
        Random number generators
 *********************************************************************/

double Randunif01()
{
    randunif01 = ((double) rand())/RAND_MAX;
    return (randunif01);
}

double Randpois_RV(double randunif01 , double  alpha)
{
  double  x;
  x  = -(1/alpha)*log(1-randunif01);
  return (x);
}
```

```
/*************************************************************************
                                Find Next Event
*************************************************************************/
void Find_CLASS_arrival()
{
                        min_t  = 100000000000;
                        for ( i = 0; i < CLASS_SIZE ; i++ )
                        {
                                alpha = arrivalrate[ i ];
                                randunif01    = Randunif01();
                                t[ i ]        = Randpois_RV(randunif01,alpha);
                                if ( min_t > t[ i ] )
                                {
                                                min_t = t[ i ];
                                                CLASS_arrival  = i;
                                }

                        }
}

/*************************************************************************
                                Write to files
*************************************************************************/
void Writeresults()
{

        if ( step%1000 == 1 )
        {
                ofstream outClientFile("Throughput-FIXED40new.txt", ios::app );
                if ( !outClientFile ) { // overload ! operator
```

```
                    cerr<<" File could not be opened"<<endl;
                    exit(1);
                }

        outClientFile<< step <<  setw(20) << Time << setw(20) << Av_throughput <<
endl;

        cout<< step << setw( 20) << Time <<  setw(20) << Av_throughput<<endl ;
    }

}

/*************************************************************************
                    Write to files
*************************************************************************/
void WriteQresults()
{

        ofstream outClientFile1("Fix.txt", ios::app );
        if ( !outClientFile1 ) { // overload ! operator
                    cerr<<" File could not be opened"<<endl;
                    exit(1);
                }

        for ( i = 0; i<level; i++ )
        {
            for ( j = 0; j< level; j++ )
            {
                    for ( k = 0; k< level; k++ )
                    {
                        for ( a = 0; a< num_action; a++ )
                        {
```

```
                                        outClientFile1<< i<< setw(20) << j <<
setw(20) << k<<  setw(20)<< a<<  setw(20) << Q[ i ][ j ][ k ][ a ] << setw(20) <<
Numvisit[ i ][ j ][ k ][ a ] <<endl;
                                            }// for a
                                        }// for k
                                    }// for j
                                }// for i

}
```

ภาคผนวก ค.

โปรแกรมระบบการทดลองในสถานการณ์จริงโดยใช้ระบบควบคุมสัญญาณไฟจราจรแบบตั้งเวลา

```
        ORG   0000H                 ;เริ่มต้นโปรแกรม

        MOV   P0,#00H               ;เคลียร์ค่าในพอร์ต P0
        MOV   P1,#00H               ;เคลียร์ค่าในพอร์ต P1
        MOV   P2,#00H               ;เคลียร์ค่าในพอร์ต P2
        MOV   P3,#00H               ;เคลียร์ค่าในพอร์ต P3
        CLR   A                     ;เคลียร์ค่าในรีจิสเตอร์ A

LOOP:   MOV   P0,#00110000B         ;กำหนดให้แยก A ไฟเขียว และแยก B ไฟแดง
        MOV   P2,#00100000B         ;กำหนดให้แยก C ไฟเขียว
        CALL  DELAY_10SEC           ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P0,#00101000B         ;กำหนดให้แยก A ไฟเหลือง และแยก B ไฟ
แดง
        CALL  DELAY_2SEC            ;หน่วงเวลาไฟเหลือง
        CALL  DELAY_1SEC
        MOV   P0,#00100100B         ;กำหนดให้แยก A ไฟแดง และแยก B ไฟแดง
        MOV   P2,#00110000B         ;กำหนดให้แยก C ไฟเขียว และเลี้ยวขวาได้
        CALL  DELAY_10SEC           ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P2,#01000000B         ;กำหนดให้แยก C ไฟเหลือง
        CALL  DELAY_2SEC            ;หน่วงเวลาไฟเหลือง
        CALL  DELAY_1SEC
        MOV   P2,#10000000B         ;กำหนดให้แยก C ไฟแดง
        MOV   P0,#10000100B         ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเขียว
        CALL  DELAY_10SEC           ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P0,#01000100B         ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเหลือง
```

```
        CALL   DELAY_2SEC                ;หน่วงเวลาไฟเหลือง
        CALL   DELAY_1SEC
        JMP    LOOP                      ;กระโดดไปที่ loop

DELAY_10SEC:    CALL   DELAY_2SEC        ;หน่วงเวลา 10 วินาที
                CALL   DELAY_2SEC
                CALL   DELAY_2SEC
                CALL   DELAY_2SEC
                CALL   DELAY_2SEC
                RET

DELAY_2SEC: CALL   DELAY_1SEC            ;หน่วงเวลา 2 วินาที
                CALL   DELAY_1SEC
                RET

DELAY_1SEC: CALL   DELAY_500M            ;หน่วงเวลา 1 วินาที
                CALL   DELAY_500M
                RET

DELAY_500M: CALL   DELAY_250M            ;หน่วงเวลา 0.5 วินาที
                CALL   DELAY_250M
                RET

DELAY_250M: MOV    R7,#5                 ;R7=5
DELAY_50M:      MOV    B,#200            ;B=200
DELAY_250U: MOV    A,#128                ;A=128
                DJNZ   Acc,$             ;ทำซ้ำจนกว่า A=0
                DJNZ   B,DELAY_250U      ; ทำซ้ำจนกว่า B=0
                DJNZ   R7,DELAY_50M      ; ทำซ้ำจนกว่า R7=0
                RET
```

ภาคผนวก ง.

โปรแกรมระบบการทดลองในสถานการณ์จริงที่ใช้ระบบควบคุมสัญญาณไฟจราจรด้วยระบบปัญญาประดิษฐ์

```
        ORG   0000H             ;เริ่มต้นโปรแกรม

        MOV   P0,#00H           ;เคลียร์ค่าในพอร์ต P0
        MOV   P1,#0FFH          ;กำหนดพอร์ต P0=0FFH
        MOV   P2,#00H           ;เคลียร์ค่าในพอร์ต P2
        MOV   P3,#00H           ;เคลียร์ค่าในพอร์ต P3
        MOV   R0,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R0
        MOV   R1,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R1
        MOV   R2,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R2
        MOV   R4,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R4
        MOV   R5,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R5
        MOV   R6,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R6
        CLR   A                 ;เคลียร์ค่าในรีจิสเตอร์ A
        CLR   C                 ;เคลียร์ค่าตัวทด

;ลูปที่กำหนดให้แยก A,C ไฟเขียว และแยก B ไฟแดง เป็นเวลา 20 วินาที

loop1:  MOV   P0,#00110000B     ;กำหนดให้แยก A ไฟเขียว และแยก B ไฟแดง
        MOV   P2,#00100000B     ;กำหนดให้แยก C ไฟเขียว
        CALL  DELAY_10SEC       ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P0,#00101000B     ;กำหนดให้แยก A ไฟเหลือง และแยก B ไฟแดง
        CALL  DELAY_2SEC        ;หน่วงเวลาไฟเหลือง
        CALL  DELAY_1SEC
        MOV   R0,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R0
        MOV   R4,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R4
        CLR   C
```

```
        MOV   A,R5              ;นำค่าใน R5 เก็บในA
        CJNE  A,#00H,rr1        ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   bb1               ;A=00H กระโดดไป bb1
rr1:    CLR   C
        CJNE  A,#01H,uu1        ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop2             ;A=01H กระโดดไป loop2
uu1:    CLR   C
        CJNE  A,#02H,bb1        ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop5             ;A=02H กระโดดไป loop5

bb1:    CLR   C
        MOV   A,R6              ;นำค่าใน R6 เก็บในA
        CJNE  A,#00H,qq1        ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   ll1               ;A=00H กระโดดไป ll1
qq1:    CLR   C
        cjne  a,#01H,ww1        ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop3             ;A=01H กระโดดไป loop3
ww1:    CLR   C
        CJNE  a,#02H,ll1        ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop6             ;A=02H กระโดดไป loop6

ll1:

;ลูปที่กำหนดให้แยก C ไฟเขียวและเลี้ยวขวาได้ และแยก A,B ไฟแดง เป็นเวลา 20 วินาที

loop2:  MOV   P0,#00100100B     ;กำหนดให้แยก A ไฟแดง และแยก B ไฟแดง
        MOV   P2,#00110000B     ;กำหนดให้แยก C ไฟเขียว และเลี้ยวขวาได้
        CALL  DELAY_10SEC       ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P2,#01000000B     ;กำหนดให้แยก C ไฟเหลือง
        CALL  DELAY_2SEC        ;หน่วงเวลาไฟเหลือง
```

```
        CALL   DELAY_1SEC
        MOV    R5,#00H          ;เคลียร์ค่าในรีจิสเตอร์ R5
        MOV    R1,#00H          ;เคลียร์ค่าในรีจิสเตอร์ R1
        CLR    C

        MOV,   A,R4             ;นำค่าใน R4 เก็บในA
        CJNE   A,#00H,rr2       ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP    bb2              ;A=00H กระโดดไป bb2
rr2:    CLR    C
        CJNE   A,#01H,uu2       ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP    loop1            ;A=01H กระโดดไป loop1
uu2:    CLR    C
        CJNE   A,#02H,bb2       ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP    loop4            ;A=02H กระโดดไป loop4

bb2:    CLR    C
        MOV    A,R6             ;นำค่าใน R6 เก็บในA
        cjne   A,#00H,qq2       ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP    ll2              ;A=00H กระโดดไป ll2
qq2:    CLR    C
        CJNE   A,#01H,ww2       ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP    loop3            ;A=01H กระโดดไป loop3
ww2:    CLR    C
        CJNE   A,#02H,ll2       ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP    loop6            ;A=02H กระโดดไป loop6

ll2:
```

```
;ลูปที่กำหนดให้แยก B ไฟเขียว และแยก A,C ไฟแดง เป็นเวลา 20 วินาที

loop3:  MOV   P2,#10000000B      ;กำหนดให้แยก C ไฟแดง
        MOV   P0,#10000100B      ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเขียว
        CALL  DELAY_10SEC        ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        MOV   P0,#01000100B      ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเหลือง
        CALL  DELAY_2SEC         ;หน่วงเวลาไฟเหลือง
        CALL  DELAY_1SEC
        MOV   R2,#00H            ;เคลียร์ค่าในรีจิสเตอร์ R2
        MOV   R6,#00H            ;เคลียร์ค่าในรีจิสเตอร์ R6
        CLR   C

        MOV   A,R4               ;นำค่าใน R4 เก็บในA
        CJNE  A,#00H,rr3         ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   bb3                ;A=00H กระโดดไป bb3
rr3:    CLR   C
        CJNE  A,#01H,uu3         ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop1              ;A=01H กระโดดไป loop1
uu3:    CLR   C
        CJNE  A,#02H,bb3         ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop4              ;A=02H กระโดดไป loop4
bb3:    CLR   C
        MOV   A,R5               ;นำค่าใน R5 เก็บในA
        CJNE  A,#00H,qq3         ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   ll3                ;A=00H กระโดดไป ll3
qq3:    CLR   C
        CJNE  A,#01H,ww3         ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop2              ;A=01H กระโดดไป loop2
ww3:    CLR   C
        CJNE  A,#02H,ll3         ;ตรวจว่าA เท่ากับ 2 หรือไม่
```

```
        JMP     loop5                   ;A=02H กระโดดไป loop5
ppp:    JMP     loop2
com3:   JMP     loop1
com4:   JMP     loop2

ll3:

;ลูปที่กำหนดให้แยก A,C ไฟเขียว และแยก B ไฟแดง เป็นเวลา 40 วินาที

loop4:  MOV     P0,#00110000B           ;กำหนดให้แยก A ไฟเขียว และแยก B ไฟแดง
        MOV     P2,#00100000B           ;กำหนดให้แยก C ไฟเขียว
        CALL    DELAY_10SEC             ;หน่วงเวลาไฟเขียว
        CALL    DELAY_10SEC
        CALL    DELAY_10SEC
        CALL    DELAY_10SEC
        MOV     P0,#00101000B           ;กำหนดให้แยก A ไฟเหลือง และแยก B ไฟแดง
        CALL    DELAY_2SEC              ;หน่วงเวลาไฟเหลือง
        CALL    DELAY_1SEC
        MOV     R0,#00H                 ;เคลียร์ค่าในรีจิสเตอร์ R0
        MOV     R4,#00H                 ;เคลียร์ค่าในรีจิสเตอร์ R4
        CLR     C

        MOV     A,R5                    ;นำค่าใน R5 เก็บในA
        CJNE    A,#00H,rr4              ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP     bb4                     ;A=00H กระโดดไป bb1
rr4:    CLR     C
        CJNE    A,#01H,uu4              ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP     loop2                   ;A=01H กระโดดไป loop2
uu4:    CLR     C
        CJNE    A,#02H,bb4              ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP     loop5                   ;A=02H กระโดดไป loop5
```

```
bb4:    CLR   C
        MOV   A,R6              ;นำค่าใน R6 เก็บในA
        CJNE  A,#00H,qq4        ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   ll4               ;A=00H กระโดดไป ll1
qq4:    CLR   C
        cjne  a,#01H,ww4        ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop3             ;A=01H กระโดดไป loop3
ww4:    CLR   C
        CJNE  a,#02H,ll4        ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop6             ;A=02H กระโดดไป loop6

com1:   JMP   com3
com2:   JMP   com4
com5:   JMP   loop3
sss:    JMP   loop1

ll4:

;ลูปที่กำหนดให้แยก C ไฟเขียวและเลี้ยวขวาได้ และแยก A,B ไฟแดง เป็นเวลา 40 วินาที

loop5:  MOV   P0,#00100100B     ;กำหนดให้แยก A ไฟแดง และแยก B ไฟแดง
        MOV   P2,#00110000B     ;กำหนดให้แยก C ไฟเขียว และเลี้ยวขวาได้
        CALL  DELAY_10SEC       ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        CALL  DELAY_10SEC
        CALL  DELAY_10SEC
        MOV   P2,#01000000B     ;กำหนดให้แยก C ไฟเหลือง
        CALL  DELAY_2SEC        ;หน่วงเวลาไฟเหลือง
        CALL  DELAY_1SEC
        MOV   R5,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R5
        MOV   R1,#00H           ;เคลียร์ค่าในรีจิสเตอร์ R1
```

```
        CLR   C
        MOV   A,R4              ;นำค่าใน R4 เก็บในA
        CJNE  A,#00H,rr5        ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   bb5               ;A=00H กระโดดไป bb2
rr5:    CLR   C
        CJNE  A,#01H,uu5        ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop1             ;A=01H กระโดดไป loop1
uu5:    CLR   C
        CJNE  A,#02H,bb5        ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop4             ;A=02H กระโดดไป loop4

bb5:    CLR   C
        MOV   A,R6              ;นำค่าใน R6 เก็บในA
        cjne  A,#00H,qq5        ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP   ll5               ;A=00 กระโดดไป ll2
qq5:    CLR   C
        CJNE  A,#01H,ww5        ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP   loop3             ;A=01H กระโดดไป loop3
ww5:    CLR   C
        CJNE  A,#02H,ll5        ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP   loop6             ;A=02H กระโดดไป loop6

ll5:


;ลูปที่กำหนดให้แยก B ไฟเขียว และแยก A,C ไฟแดง เป็นเวลา 40 วินาที

loop6:  MOV   P2,#10000000B     ;กำหนดให้แยก C ไฟแดง
        MOV   P0,#10000100B     ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเขียว
        CALL  DELAY_10SEC       ;หน่วงเวลาไฟเขียว
        CALL  DELAY_10SEC
        CALL  DELAY_10SEC
```

```
        CALL   DELAY_10SEC
        MOV    P0,#01000100B      ;กำหนดให้แยก A ไฟแดง และแยก B ไฟเหลือง
        CALL   DELAY_2SEC         ;หน่วงเวลาไฟเหลือง
        CALL   DELAY_1SEC
        MOV    R2,#00H            ;เคลียร์ค่าในรีจิสเตอร์ R2
        MOV    R6,#00H            ;เคลียร์ค่าในรีจิสเตอร์ R6
        CLR    C

        MOV    A,R4               ;นำค่าใน R4 เก็บในA
        CJNE   A,#00H,rr6         ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP    bb6                ;A=00H กระโดดไป bb3
rr6:    CLR    C
        CJNE   A,#01H,uu6         ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP    loop1              ;A=01H กระโดดไป loop1
uu6:    CLR    C
        CJNE   A,#02H,bb6         ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP    loop4              ;A=02H กระโดดไป loop4

bb6:    CLR    C
        MOV    A,R5               ;นำค่าใน R5 เก็บในA
        CJNE   A,#00H,qq6         ;ตรวจว่า A เท่ากับ 0 หรือไม่
        JMP    ll6                ;A=00H กระโดดไป ll3
qq6:    CLR    C
        CJNE   A,#01H,ww6         ;ตรวจว่าA เท่ากับ 1 หรือไม่
        JMP    loop2              ;A=01H กระโดดไป loop2
ww6:    CLR    C
        CJNE   A,#02H,ll6         ;ตรวจว่าA เท่ากับ 2 หรือไม่
        JMP    loop5              ;A=02H กระโดดไป loop5

ll6:    JMP    sss
```

```
DELAY_10SEC:    CALL  DELAY_2SEC  ;หน่วงเวลา 10 วินาที
                CALL  DELAY_2SEC
                CALL  DELAY_2SEC
                CALL  DELAY_2SEC
                CALL  DELAY_2SEC
                RET

DELAY_2SEC: CALL  DELAY_1SEC      ;หน่วงเวลา 2 วินาที
                CALL  DELAY_1SEC
                RET

DELAY_1SEC: CALL  DELAY_500M      ;หน่วงเวลา 1 วินาที
                CALL  DELAY_500M
                RET

DELAY_500M: CALL  DELAY_250M      ;หน่วงเวลา 0.5 วินาที
                CALL  DELAY_250M
                RET
DELAY_250M: MOV   R7,#5           ;R7=5
DELAY_50M:  MOV   B,#200          ;B=200
DELAY_250U: MOV   A,#128          ;A=128
            DJNZ  Acc,$           ;ทำซ้ำจนกว่า A=0
            DJNZ  B,DELAY_250U    ; ทำซ้ำจนกว่า B=0
            DJNZ  R7,DELAY_50M    ; ทำซ้ำจนกว่า R7=0
            JNB   P1.0,set1       ;เฝ้ามองบิตที่พอร์ต P1.0
      aa1:
            JNB   P1.1,set2       ;เฝ้ามองบิตที่พอร์ต P1.1
      aa2:
            JNB   P1.2,set3       ;เฝ้ามองบิตที่พอร์ต P1.2
      aa3:
            RET
```

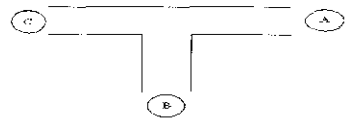
```
set1:   INC   R0          ;เพิ่มค่า R0
        CLR   C
        MOV   A,R0        ;นำค่าใน R0 เก็บในA
        CJNE  A,#0AH,xx   ;ตรวจว่าA เท่ากับ 0AH หรือไม่
        MOV   R4,#01H     ;A = 0AH ให้ R4 = 01H
xx:     CLR   C
        CJNE  A,#14H,xx2  ;ตรวจว่าA เท่ากับ 14H หรือไม่
        MOV   R4,#02H     ;A = 0AH ให้ R4 = 02H
xx2:    JMP   aa1

set2:   INC   R1          ;เพิ่มค่า R1
        CLR   C
        MOV   A,R1        ;นำค่าใน R1 เก็บในA
        CJNE  A,#0AH,xx3  ;ตรวจว่าA เท่ากับ 0AH หรือไม่
        MOV   R5,#01H     ;A = 0AH ให้ R5 = 01H
xx3:    CLR   C
        CJNE  A,#14H,xx4  ;ตรวจว่าA เท่ากับ 14H หรือไม่
        MOV   R5,#02H     ;A = 0AH ให้ R5 = 02H
xx4:    JMP   aa2

set3:   INC   R2          ;เพิ่มค่า R2
        CLR   C
        MOV   A,R0        ;นำค่าใน R2 เก็บในA
        CJNE  A,#0AH,xx5  ;ตรวจว่าA เท่ากับ 0AH หรือไม่
        MOV   R6,#01H     ;A = 0AH ให้ R6 = 01H
xx5:    CLR   C
        CJNE  A,#14H,xx6  ;ตรวจว่าA เท่ากับ 14H หรือไม่
        MOV   R6,#02H     ;A = 0AH ให้ R6 = 02H
xx6:    JMP   aa3
```

ภาคผนวก จ.

ตารางข้อมูลในการนับจำนวนรถ (จักรยานยนต์, รถยนต์) ที่สามแยกหน้าส่วนกิจการนักศึกษาทางเข้าเรียนรวม



วัน	เวลา / เส้นทาง	7.51 น.	7.52 น.	7.53 น.	7.54 น.	7.55 น.	7.56 น.	7.57 น.	7.58 น.	7.59 น.	8:00 น.
20/6/2005	A	51,2	28,0	21,2	36,4	34,3	41,1	14,4	54,2	44,6	76,4
	B	0,1	2,0	4,1	1,3	2,4	0,0	2,0	2,1	1,0	0,1
	C	1,0	0,0	3,1	2,0	1,0	0,0	2,0	0,1	0,1	0,0
21/6/2005	A	0,1	0,1	0,0	0,0	0,0	0,0	0,1	1,1	0,0	11,1
	B	10,2	6,3	7,0	19,2	27,0	37,4	16,3	35,5	20,1	0,0
	C	0,0	0,0	0,0	0,0	0,0	0,0	0,0	2,1	1,1	0,0
22/6/2005	A	37,1	32,2	30,2	40,6	42,3	40,3	32,6	39,3	44,6	31,1
	B	4,0	1,0	1,1	3,0	2,0	2,1	1,1	2,1	0,1	2,0
	C	3,1	1,0	0,0	1,0	1,1	2,1	2,1	2,0	1,0	3,1
23/6/2005	A	24,2	16,2	27,1	24,1	37,0	26,4	40,0	35,2	29,1	33,2
	B	0,0	0,0	3,0	0,0	0,0	1,0	1,0	1,0	0,0	1,0
	C	0,0	2,0	0,0	0,0	0,2	1,0	0,0	0,0	0,0	0,1

Services

วัน	เวลา / เส้นทาง	7.51 น.	7.52 น.	7.53 น.	7.54 น.	7.55 น.	7.56 น.	7.57 น.	7.58 น.	7.59 น.	8:00 น.
24/6/2005	A	28,4	18,0	20,5	26,1	18,0	19,3	25,0	32,3	24,1	21,5
	B	3,0	1,0	2,0	2,0	1,0	0,0	3,1	0,0	0,0	2,0
	C	0,1	0,1	0,1	0,1	0,0	0,0	0,0	2,0	0,0	0,1
27/6/2005	A	26,0	30,2	35,1	28,1	40,3	35,0	33,1	46,2	50,2	58,2
	B	0,0	1,0	2,0	1,1	1,1	0,0	1,1	2,2	1,1	0,0
	C	0,0	0,0	1,0	0,1	0,0	2,0	1,1	0,0	0,2	1,0
28/6/2005	A	26,1	19,3	15,0	16,0	24,0	23,2	19,1	26,1	22,2	28,4
	B	0,0	0,0	0,1	0,0	0,1	1,0	1,0	1,1	1,0	0,0
	C	0,0	0,0	0,0	0,0	1,0	1,0	0,1	0,2	2,0	1,0
29/6/2005	A	28,1	26,0	33,2	40,2	44,4	37,3	29,0	45,3	35,2	48,1
	B	0,0	2,0	2,1	0,0	1,0	0,1	0,0	0,0	0,0	1,0
	C	0,0	0,1	0,0	1,0	0,0	0,0	2,1	2,1	2,0	1,0
30/6/2005	A	6,0	6,1	11,0	15,2	18,3	15,2	20,5	24,2	23,1	26,4
	B	0,0	0,0	0,0	1,0	1,0	1,0	1,1	1,0	0,0	0,0
	C	0,0	0,0	0,0	0,0	1,0	0,0	0,0	1,0	2,0	0,0

วัน	เวลา / เส้นทาง	8.01 น.	8.02 น.	8.03 น.	8.04 น.	8.05 น.	8.06 น.	8.07 น.	8.08 น.	8.09 น.	8:10 น.	รวม
20/6/2005	A	54,5	38,1	81,4	96,2	73,1	63,2	24,4	68,2	37,4	55,1	933,54
	B	2,0	1,0	3,0	1,1	0,0	2,0	1,0	0,1	3,1	0,1	27,15
	C	1,1	0,1	2,0	3,0	1,0	1,3	1,0	1,0	1,0	1,1	21,9
21/6/2005	A	27,3	29,2	33,1	32,0	28,2	27,4	18,1	22,3	19,0	20,3	443,40
	B	1,0	0,0	1,0	2,0	6,1	4,0	1,1	1,0	0,0	1,0	20,4
	C	0,0	0,1	1,0	0,0	1,0	1,0	0,0	1,0	3,0	0,1	8,6
22/6/2005	A	34,0	43,1	43,1	40,6	40,5	33,0	21,1	29,4	27,1	23,2	700,52
	B	2,1	1,0	2,0	0,0	3,0	1,0	3,1	1,0	1,0	2,1	34,8
	C	2,1	1,0	1,1	1,0	0,0	2,1	2,1	1,1	3,1	1,0	30,11
23/6/2005	A	23,3	31,2	27,2	32,1	19,3	26,0	25,2	20,0	25,2	26,2	545,32
	B	1,0	2,0	0,0	1,0	0,0	1,0	0,1	1,0	1,0	0,0	14,1
	C	2,0	0,0	0,1	1,1	1,1	0,0	0,1	0,0	1,1	0,0	8,9
24/6/2005	A	27,6	19,4	15,2	11,1	18,3	12,4	12,1	14,4	16,1	18,2	393,50
	B	2,0	2,0	1,0	1,0	1,0	1,0	1,1	1,0	0,0	1,0	25,2
	C	1,0	0,0	2,0	0,1	1,0	0,0	0,0	0,0	1,0	0,1	7,7

วัน	เวลา / เส้นทาง	8.01 น.	8.02 น.	8.03 น.	8.04 น.	8.05 น.	8.06 น.	8.07 น.	8.08 น.	8.09 น.	8:10 น.	รวม
27/6/2005	A	76,1	51,2	64,2	84,3	76,2	55,4	61,0	93,1	43,3	78,4	1062,36
	B	3,0	1,1	0,0	1,1	0,0	0,0	1,1	0,0	2,1	2,2	19,12
	C	2,0	1,1	0,0	2,0	2,0	2,1	1,1	0,0	1,0	1,1	17,8
28/6/2005	A	23,1	30,1	35,1	36,3	30,0	26,4	21,4	25,1	31,2	23,1	498,32
	B	0,0	1,0	2,1	2,0	2,0	1,0	0,0	1,0	0,0	1,1	14,5
	C	0,0	1,0	3,0	0,0	0,0	3,0	2,2	1,2	0,0	0,1	15,8
29/6/2005	A	51,3	37,0	31,3	38,4	43,1	48,1	36,0	44,5	31,6	17,1	741,42
	B	1,0	0,0	2,0	3,0	1,1	2,0	1,1	0,0	2,0	1,0	19,4
	C	1,0	1,1	2,1	0,0	1,0	1,0	1,1	0,1	0,1	3,1	18,9
30/6/2005	A	30,3	15,2	16,2	30,0	17,1	13,1	20,2	16,1	22,1	24,3	367,36
	B	2,0	2,1	1,0	0,0	1,0	2,0	2,1	1,1	0,0	3,0	19,4
	C	1,0	2,0	2,1	1,1	0,1	1,0	1,1	1,1	1,2	1,1	15,8

วัน	เวลา / เส้นทาง	9.51	9.52	9.53	9.54	9.55	9.56	9.57	9.58	9.59	10:00
20/6/2005	A	20,0	27,1	24,5	37,5	50,6	32,1	26,0	30,0	32,1	30,5
	B	1,0	0,1	4,0	3,0	10,0	13,1	27,2	31,1	46,1	25,1
	C	1,0	0,2	1,0	2,0	2,0	0,1	0,0	0,0	1,1	2,0
21/6/2005	A	40,2	44,0	43,3	22,3	30,0	38,1	30,3	43,2	31,0	41,0
	B	6,1	0,0	2,0	3,0	5,1	9,0	6,0	13,0	15,1	16,1
	C	2,1	2,0	2,0	2,0	1,1	0,0	2,1	0,0	1,1	2,0
22/6/2005	A	28,2	20,1	32,3	35,	25,0	41,4	32,1	40,4	38,2	25,4
	B	4,0	1,1	1,1	3,0	1,0	7,1	3,0	11,0	16,1	30,0
	C	2,0	3,0	2,0	1,0	0,1	2,0	2,0	0,2	2,0	3,0
23/6/2005	A	16,1	41,1	12,1	21,1	18,1	16,1	19,0	20,2	12,2	11,1
	B	20,0	24,0	10,1	15,0	9,0	7,0	4,0	5,0	6,1	10,0
	C	0,0	0,1	0,0	0,0	1,0	1,1	1,0	0,0	0,0	1,0
24/6/2005	A	7,1	15,2	24,4	12,1	35,1	26,2	12,2	27,1	26,1	30,2
	B	1,0	3,0	1,0	2,0	11,0	11,0	10,0	10,1	13,2	11,2
	C	0,0	2,0	0,0	0,3	2,1	2,1	4,1	2,0	1,1	2,0

Service

วัน	เวลา / เส้นทาง	9.51	9.52	9.53	9.54	9.55	9.56	9.57	9.58	9.59	10:00
27/6/2005	A	22,0	18,3	19,1	23,1	17,0	32,3	23,1	30,0	39,3	26,1
	B	22,1	18,1	16,1	23,1	25,0	26,1	16,3	20,1	29,3	34,1
	C	1,0	1,1	1,0	0,0	2,0	0,1	0,0	0,0	1,1	2,1
28/6/2005	A	40,0	44,1	25,1	33,6	21,1	38,1	45,0	36,0	39,1	36,1
	B	13,1	20,0	15,0	13,1	8,0	16,1	7,0	6,2	14,1	26,0
	C	1,0	0,0	2,0	2,1	3,0	1,0	0,0	2,0	2,1	3,1
29/6/2005	A	27,1	26,0	30,1	21,3	35,2	26,4	30,1	43,1	26,3	32,1
	B	3,1	4,0	0,0	4,0	6,0	8,2	10,0	16,1	8,0	6,0
	C	2,0	1,1	2,0	1,0	1,1	2,1	2,2	0,2	1,1	0,0
30/6/2005	A	15,1	25,3	20,1	19,0	29,1	20,1	17,0	25,2	17,0	14,3
	B	7,0	5,0	11,0	2,1	13,1	10,0	14,0	8,0	10,0	13,1
	C	0,0	0,1	0,0	1,0	0,0	1,1	0,1	1,1	1,1	2,1

Source57

วัน	เวลา / เส้นทาง	10.01	10.02	10.03	10.04	10.05	10.06	10.07	10.08	10.09	10:10	รวม
20/6/2005	A	36,1	22,2	23,2	42,2	20,11	13,2	14,1	10,0	11,2	14,1	513,48
	B	33,1	42,3	35,2	15,1	28,2	18,2	24,1	22,0	8,1	11,1	396,21
	C	1,0	2,2	5,0	3,0	1,0	1,0	1,0	1,1	2,0	3,0	29,7
21/6/2005	A	47,5	23,4	36,2	25,1	16,0	22,2	31,1	28,2	18,1	13,0	621,32
	B	10,1	17,1	17,1	17,0	18,2	19,0	14,1	14,0	13,1	13,2	227,31
	C	4,0	0,0	0,0	0,0	1,0	1,2	0,0	2,1	1,0	1,1	24,8
22/6/2005	A	34,2	32,4	34,0	47,4	23,4	22,0	24,2	7,1	28,1	18,0	585,40
	B	16,0	28,2	26,1	41,1	15,1	15,1	17,0	39,2	23,1	10,2	307,16
	C	0,0	5,0	2,0	1,0	1,0	1,0	1,1	1,2	1,0	0,0	30,6
23/6/2005	A	5,0	12,0	12,0	6,1	1,1	8,2	10,1	8,0	13,0	15,2	286,18
	B	12,0	10,0	12,0	18,0	20,0	24,0	18,0	15,0	9,0	8,0	256,2
	C	0,0	0,0	1,0	0,0	3,0	1,1	0,0	2,0	1,1	2,0	14,4
24/6/2005	A	24,3	13,1	29,0	14,1	10,0	14,0	12,1	6,2	10,2	2,0	348,27
	B	13,1	13,1	13,1	20,0	8,0	6,1	7,0	8,1	4,0	7,1	172,11
	C	2,0	3,0	2,2	4,0	3,1	2,1	2,0	2,1	3,0	1,0	39,12

วัน	เวลา / เส้นทาง	10.01	10.02	10.03	10.04	10.05	10.06	10.07	10.08	10.09	10:10	[illegible]
27/6/2005	A	33,3	30,4	29,1	40,3	25,2	16,1	32,1	43,2	34,0	14,2	545,31
	B	44,1	46,0	31,1	36,2	26,1	23,4	37,2	22,0	29,1	23,1	546,26
	C	0,1	0,1	3,0	1,0	3,1	1,0	1,0	2,0	2,0	1,0	22,7
28/6/2005	A	39,1	46,1	36,2	25,1	16,2	41,3	27,1	31,0	23,0	40,2	681,25
	B	10,0	17,0	3,1	20,0	11,0	8,0	5,0	3,0	14,1	11,0	240,8
	C	3,2	1,0	1,1	0,0	0,0	2,0	1,0	1,1	0,2	1,1	26,10
29/6/2005	A	40,3	34,0	32,2	19,4	26,3	22,1	17,1	31,3	18,0	36,2	571,36
	B	10,0	12,2	13,1	8,0	4,1	13,2	5,2	8,0	9,1	5,0	152,13
	C	1,0	1,0	2,0	3,1	1,1	0,1	1,1	1,1	0,1	1,1	23,14
30/6/2005	A	16,4	25,0	24,2	19,2	11,0	30,2	16,1	22,1	26,2	18,2	408,28
	B	5,2	9,2	11,1	6,0	4,1	13,1	15,0	20,0	12,2	11,0	199,12
	C	2,1	1,1	1,0	0,0	0,0	1,0	1,1	0,1	0,1	2,0	14,10

วัน	เวลา / เส้นทาง	12.51	12.52	12.53	12.54	12.55	12.56	12.57	12.58	12.59	13:00
20/6/2005	A	9,2	17,2	16,2	20,3	21,1	27,0	21,4	26,0	24,1	21,1
	B	14,1	7,1	5,0	10,1	9,1	4,0	8,0	8,0	5,0	7,0
	C	8,2	3,1	4,0	3,5	3,0	5,0	5,1	4,1	6,0	3,0
21/6/2005	A	ฝนตก									
	B	ฝนตก									
	C	ฝนตก									
22/6/2005	A	10,1	14,0	18,1	18,2	24,0	19,2	18,1	17,2	21,3	17,3
	B	7,1	8,0	4,0	8,1	3,0	1,0	5,0	3,0	9,0	4,1
	C	7,0	2,3	5,0	7,	4,0	3,1	9,0	2,0	8,0	5,0
23/6/2005	A	15,2	20,0	17,0	22,2	21,1	30,3	15,1	16,1	23,4	18,0
	B	16,0	12,1	20,0	11,2	16,0	11,1	8,0	5,0	15,3	6,1
	C	1,2	1,1	0,1	2,0	1,0	2,0	0,0	3,1	1,0	2,0
24/6/2005	A	20,0	22,1	18,2	18,0	26,3	24,1	20,0	16,0	25,0	19,2
	B	5,0	6,1	2,0	14,1	11,2	8,0	13,0	15,1	9,2	15,0
	C	1,1	2,0	0,0	0,1	0,0	3,0	1,0	2,1	4,0	0,2

SOUVENIRS

วัน	เวลา / เส้นทาง	12.51	12.52	12.53	12.54	12.55	12.56	12.57	12.58	12.59	13:00
27/6/2005	A	16,0	16,0	15,4	20,1	23,1	14,3	15,0	11,2	24,1	26,1
	B	11,1	9,2	9,0	11,1	12,2	6,0	13,0	8,1	5,0	7,0
	C	5,0	3,0	1,0	4,1	2,2	6,1	4,0	1,1	6,2	1,3
28/6/2005	A	26,1	18,0	16,0	21,3	26,0	31,0	22,0	25,1	35,1	22,1
	B	22,3	16,1	25,1	23,0	22,0	26,0	14,1	16,1	10,1	15,0
	C	2,0	1,0	0,0	3,0	3,2	1,1	2,0	1,2	4,0	5,2
29/6/2005	A	8,0	6,0	12,1	16,1	16,2	19,2	11,3	21,1	23,2	16,0
	B	0,0	5,0	10,1	8,1	2,1	11,2	6,0	14,0	9,0	13,0
	C	2,1	1,1	3,1	4,0	3,0	2,1	1,1	0,2	3,1	1,1
30/6/2005	A	10,0	12,1	17,2	20,1	22,1	19,1	24,2	16,1	27,0	15,2
	B	10,1	15,2	7,0	15,1	13,0	13,1	13,0	16,1	10,1	7,0
	C	1,1	1,1	1,2	0,1	0,0	2,0	1,0	1,1	0,1	2,1

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

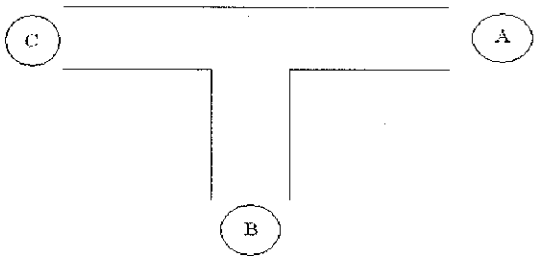
วัน	เวลา / เส้นทาง	13.01	13.02	13.03	13.04	13.05	13.06	13.07	13.08	13.09	13:10	รวม
20/6/2005	A	24,2	19,1	17,5	18,2	9,1	9,1	16,3	9,1	4,0	8,0	335,32
	B	6,0	12,1	5,4	7,0	1,0	3,0	7,0	18,0	8,1	7,1	151,11
	C	2,0	4,1	5,0	0,1	2,1	1,0	2,1	2,1	3,0	2,0	67,15
21/6/2005	A	ฝนตก										
	B	ฝนตก										
	C	ฝนตก										
22/6/2005	A	8,0	11,1	16,2	10,2	12,3	11,1	10,1	7,0	8,2	7,1	276,28
	B	4,2	11,0	9,0	3,1	5,0	10,0	15,0	10,0	9,0	8,1	131,7
	C	1,0	2,1	3,0	2,0	2,1	1,0	1,1	0,1	1,1	3,1	68,10
23/6/2005	A	20,0	25,1	22,3	9,5	14,0	6,0	15,4	12,2	22,0	14,0	457,29
	B	10,0	12,1	7,0	6,0	0,2	1,0	5,0	6,3	15,1	11,0	193,15
	C	0,1	1,1	2,0	1,1	0,1	3,0	1,0	1,2	2,2	1,1	25,14
24/6/2005	A	28,0	22,0	26,1	18,2	15,1	24,0	21,1	22,0	22,1	25,3	431,18
	B	4,0	20,2	16,0	16,3	11,0	14,0	23,3	15,0	19,1	16,2	252,18
	C	3,1	0,0	1,0	1,0	0,0	0,0	3,1	5,2	1,1	1,1	28,11

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

วัน	เวลา / เส้นทาง	13.01	13.02	13.03	13.04	13.05	13.06	13.07	13.08	13.09	13:10	[illegible]
27/6/2005	A	30,2	22,1	26,0	18,0	14,0	23,2	13,4	17,0	21,1	10,0	374,23
	B	4,2	9,1	6,1	7,0	6,0	13,1	15,1	16,0	9,1	6,0	182,13
	C	2,0	4,0	6,2	1,1	0,3	1,0	1,0	2,1	3,0	2,1	55,18
28/6/2005	A	26,1	19,1	16,3	16,1	22,2	26,1	30,0	32,0	26,1	21,3	476,20
	B	24,1	23,0	13,2	13,0	16,0	22,1	15,2	26,1	18,0	26,0	385,15
	C	1,1	0,0	2,0	3,0	1,1	2,1	3,0	2,2	1,1	0,1	37,14
29/6/2005	A	22,1	11,3	16,2	7,0	13,1	11,1	10,1	8,0	13,0	22,1	281,22
	B	9,0	11,0	6,0	12,1	15,1	13,1	5,0	16,1	20,2	16,0	201,11
	C	2,1	0,2	3,2	1,1	1,1	2,0	3,2	1,1	4,0	2,1	39,20
30/6/2005	A	16,1	22,0	9,1	15,1	14,0	20,0	26,1	19,3	17,1	16,0	356,19
	B	14,1	0,3	18,5	10,1	14,0	7,0	8,0	12,1	15,1	13,0	230,19
	C	1,1	1,1	0,0	2,1	2,2	1,1	0,0	1,0	1,1	1,0	19,15

ภาคผนวก ฉ.

ตารางข้อมูลดิบที่ได้จากการทดลองในสถานการณ์จริง



การทดลองโดยใช้ระบบไฟจราจรแบบตั้งเวลา(Fixed Time)

ความเร็วในการขับขี่ 20 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	16	8	16	14	20	12	14	100
B	4	13	9	14	8	4	10	62
C	4	4	2	6	7	8	3	34
รวม	24	25	27	34	35	24	27	196

ความเร็วในการขับขี่ 30 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	3	16	4	20	11	28	4	86
B	9	7	7	12	6	13	11	65
C	5	4	5	6	4	6	3	33
รวม	17	27	16	38	21	47	18	184

ความเร็วในการขับขี่ 40 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	5	4	18	14	5	6	18	70
B	10	9	19	6	16	2	10	72
C	1	5	8	7	2	2	7	32
รวม	16	18	45	27	23	10	35	174

การทดลองโดยใช้คนในการตัดสินใจในการปล่อยจราจร

ความเร็วในการขับขี่ 20 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	37	31	4	13	10	26	28	149
B	28	33	13	0	22	25	11	132
C	3	4	5	4	3	2	5	26
รวม	68	68	22	17	35	53	44	307

ความเร็วในการขับขี่ 30 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	20	31	16	6	34	26	13	146
B	14	26	12	24	24	5	1	106
C	3	7	5	6	4	5	12	42
รวม	37	64	33	36	62	36	26	294

ความเร็วในการขับขี่ 40 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	41	5	16	29	15	13	42	161
B	16	13	13	14	4	11	19	90
C	5	10	5	2	0	2	3	27
รวม	62	28	34	45	19	26	64	278

การทดลองใช้ไฟจราจรระบบปัญญาประดิษฐ์ *(Artificial Intelligence* หรือ *A.I.* หรือ เอไอ)

ความเร็วในการขับขี่ 20 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	20	27	4	7	15	9	20	102
B	18	19	15	13	10	5	0	80
C	5	14	2	0	4	5	6	36
รวม	43	60	21	20	29	19	26	218

ความเร็วในการขับขี่ 30 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	6	0	22	26	22	24	12	112
B	28	20	18	6	12	17	15	116
C	4	2	5	6	9	7	0	33
รวม	38	22	45	38	43	48	27	261

ความเร็วในการขับขี่40 กิโลเมตรต่อชั่วโมง

เส้นทาง	เวลา (นาที)							รวม
	1	2	3	4	5	6	7	
A	19	15	21	20	18	27	20	140
B	9	10	20	8	1	10	8	66
C	1	6	6	6	5	3	0	27
รวม	29	31	47	34	24	40	28	233

tures

- npatible with MCS®51 Products
- 8ytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 8ytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- to 6V Operating Range
- ly Static Operation: 0 Hz to 24 MHz
- ee-level Program Memory Lock
- x 8-bit Internal RAM
- Programmable I/O Lines
- ee 16-bit Timer/Counters
- e Interrupt Sources
- grammable UART Serial Channel
- Serial Interface
- w-power Idle and Power-down Modes
- errupt Recovery from Power-down
- grammable Watchdog Timer
- al Data Pointer
- wer-off Flag

scription

AT89S8252 is a low-power, high-performance CMOS 8-bit microcontroller with 8K s of downloadable Flash programmable and erasable read-only memory and 2K s of EEPROM. The device is manufactured using Atmel's high-density nonvolatile ory technology and is compatible with the industry-standard 80C51 instruction nd pinout. The on-chip downloadable Flash allows the program memory to be grammed In-System through an SPI serial interface or by a conventional nonvol- memory programmer. By combining a versatile 8-bit CPU with downloadable h on a monolithic chip, the Atmel AT89S8252 is a powerful microcontroller, which des a highly-flexible and cost-effective solution to many embedded control cations.

AT89S8252 provides the following standard features: 8K bytes of downloadable h, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog , two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt ecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, T89S8252 is designed with static logic for operation down to zero frequency and orts two software selectable power saving modes. The Idle Mode stops the CPU allowing the RAM, timer/counters, serial port, and interrupt system to continue oning. The Power-down mode saves the RAM contents but freezes the oscillator, ling all other chip functions until the next external interrupt or hardware reset.

downloadable Flash can be changed a single byte at a time and is accessible gh the SPI serial interface. Holding RESET active forces the SPI bus into a serial amming interface and allows the program memory to be written to or read from s lock bits have been activated.

ATMEL®

8-bit Microcontroller with 8K Bytes Flash

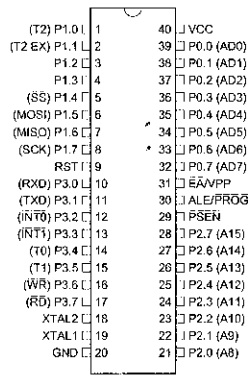
AT89S8252

0401F–MICRO–11/03

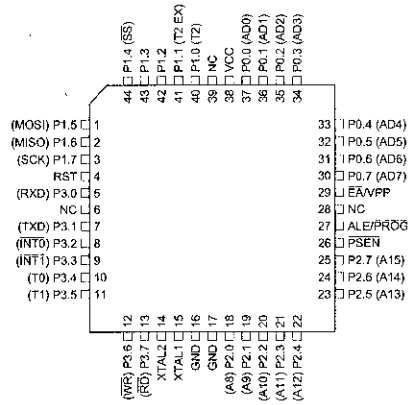
ATMEL®

Configurations

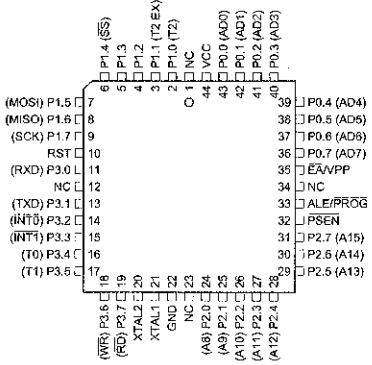
PDIP



TQFP



PLCC



Description

Supply voltage.

Ground.

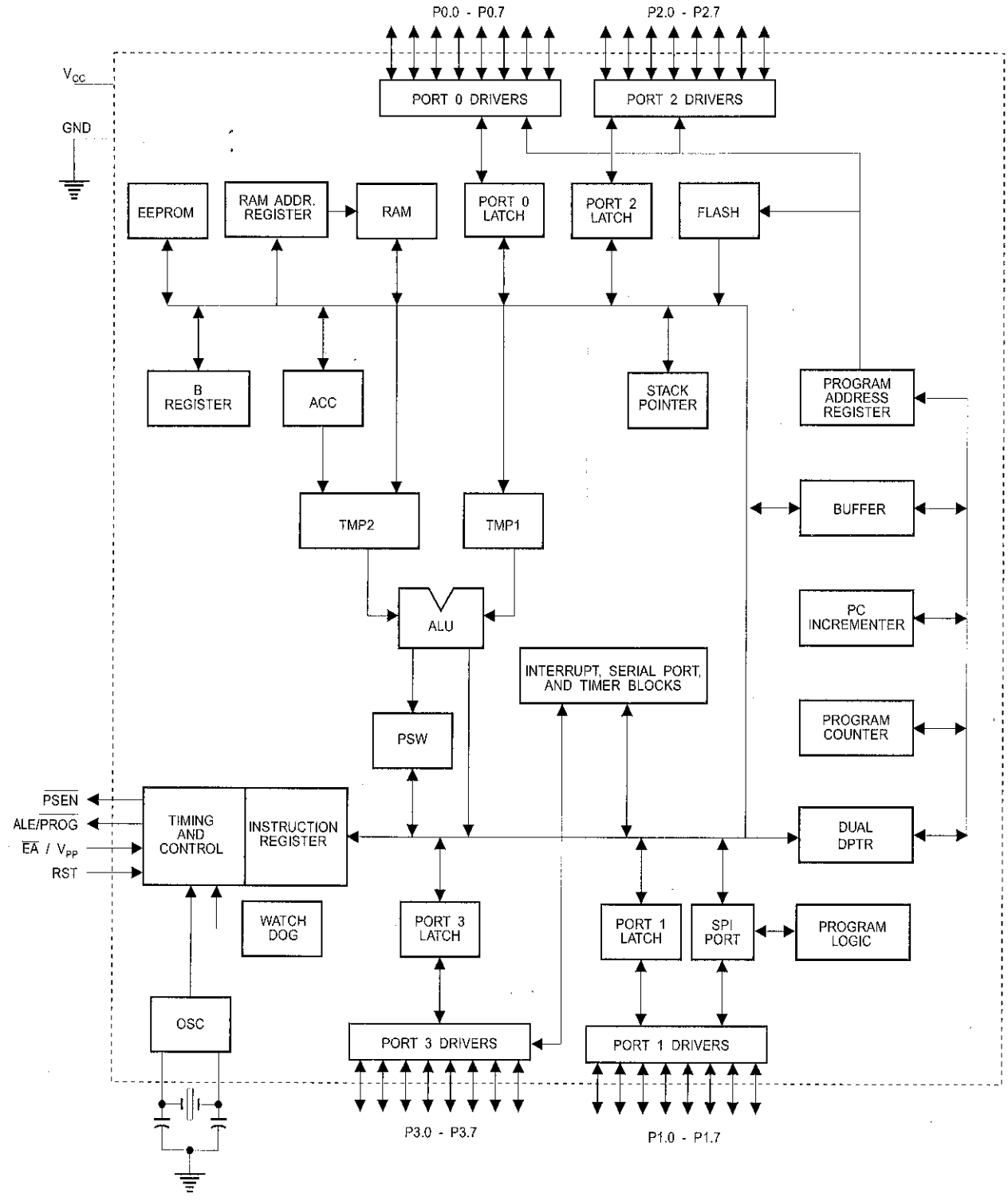
Port 0 is an 8-bit open drain bi-didirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

ck Diagram


P0.0 - P0.7
P2.0 - P2.7
V_{CC}
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
EEPROM
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
BUFFER
TMP2
TMP1
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
PSEN
ALE/PROG
EA / V_{PP}
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DUAL DPTR
WATCH DOG
PORT 3 LATCH
PORT 1 LATCH
SPI PORT
PROGRAM LOGIC
OSC
PORT 3 DRIVERS
PORT 1 DRIVERS
P3.0 - P3.7
P1.0 - P1.7

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

t 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

t 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Γ

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

Ε/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

ΕN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

ΧL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

ΧL2

Output from the inverting oscillator amplifier.

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี