

ระบบติดตามรถตู้โดยสารสาธารณะโดยใช้โปรโตคอล MQTT
PUBLIC VAN MONITORING SYSTEM USING MQTT

ภวพล ภวปัญญากุล
ธีรรัตน์ จิงชนาเจริญเลิศ

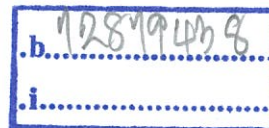
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

ระบบติดตามรถตู้โดยสารสาธารณะโดยใช้โปรโตคอล MQTT
PUBLIC VAN MONITORING SYSTEM USING MQTT



ภวพล ภวปัญญากุล
ธีรัตน์ จิงธนาเจริญเลิศ

เลขหมู่.....
เลขทะเบียน 144370
วันเดือนปี 24 พ.ย. 2559



ปฏิญญานិพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

ปริญญาานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบติดตามรถตู้โดยสารสาธารณะโดยใช้โปรโตคอล MQTT

PUBLIC VAN MONITORING SYSTEM USING MQTT

ผู้จัดทำ

1. นายภพล ภาวปัญญากุล รหัสนักศึกษา 55010911
2. นายธีรรัตน์ จิงชนาเจริญเลิศ รหัสนักศึกษา 55011416



อาจารย์ที่ปรึกษา

(อาจารย์สรยุทธ กลมกล่อม)

ระบบติดตามรถตู้โดยสารสาธารณะโดยใช้โปรโตคอล MQTT

นายภวพล ภาวปัญญากุล 55010911

นายธีรรัตน์ จิงชนาเจริญเลิศ 55011416

อาจารย์สรยุทธ กลมกล่อม อาจารย์ที่ปรึกษา
ปีการศึกษา 2558

บทคัดย่อ

โครงการงานชิ้นนี้จัดทำขึ้นเพื่อศึกษา ออกแบบ และสร้างระบบเพื่อติดตามรถตู้โดยสารสาธารณะโดยมีการแสดงผลบนหน้าเว็บแอปพลิเคชัน โครงการงานจะแบ่งเป็น 2 ส่วนหลัก โดยในส่วนแรกคือ ส่วนของอุปกรณ์ฮาร์ดแวร์ ที่เป็นอุปกรณ์ที่นำไปติดตั้งไว้ที่รถตู้โดยสารสาธารณะ ซึ่งอุปกรณ์ฮาร์ดแวร์ จะประกอบไปด้วย Arduino Board และ GPRS/GPS Module เป็นอุปกรณ์ที่ใช้ในการติดตามรถตู้โดยสารสาธารณะด้วยเทคโนโลยี GPS และทำการเชื่อมต่อกับอินเทอร์เน็ตด้วยเทคโนโลยี GPRS และส่วนที่สองคือส่วนของเว็บเซิร์ฟเวอร์ ที่จะทำการรับข้อมูลจากอุปกรณ์ฮาร์ดแวร์ เพื่อนำข้อมูล และตำแหน่งของรถตู้โดยสารสาธารณะมาแสดงผลบนหน้าเว็บแอปพลิเคชันในรูปแบบของแผนที่ออนไลน์ โดยการรับส่งข้อมูลระหว่างอุปกรณ์ฮาร์ดแวร์ กับเว็บเซิร์ฟเวอร์ จะใช้โปรโตคอล MQTT ผลลัพธ์ที่ได้คือ ระบบที่สามารถให้แสดงตำแหน่งของรถตู้โดยสารสาธารณะ และข้อมูลต่าง ๆ เช่น เลขทะเบียนรถ ชื่อคนขับ บนแผนที่ออนไลน์เพื่อให้ผู้ใช้งานสามารถค้นหา และติดตามรถตู้โดยสารสาธารณะที่ต้องการได้ อีกทั้งมีส่วนติดต่อผู้ใช้งานที่สวยงาม ใช้งานง่าย ซึ่งประโยชน์ที่ได้รับจากโครงการนี้คือ ช่วยลดจำนวนการเกิดอุบัติเหตุของรถตู้โดยสารสาธารณะซึ่งมีสาเหตุที่สำคัญจากการขับรถเร็วเกินกว่าที่กฎหมายกำหนด ด้วยการติดตามตำแหน่ง และแสดงความเร็วปัจจุบันของรถตู้โดยสารสาธารณะ

Public Van Monitoring System Using MQTT

Mr. Pavapol Pavapanyakul 55010911

Mr. Theerat Chungthanacharoenlert 55011416

Mr. Sorayut Glomglome Advisor

Academic Year 2558

ABSTRACT

Our Project aims to study, design and implement public van monitoring system that shows on web application. Project is divided into two main parts. The first part is about hardware installed on public van. The hardware combines with microcontroller and GPRS/GPS module. The hardware uses GPS technology to track a public van and use GPRS technology to connect to Internet. Second part involves webserver receiving data from hardware and shows public van location and relevance data on the online map in web application. We use MQTT protocol to send and receive data between hardware and webserver. The result is the system that user can search and monitoring public vans and see public vans details such as van license plate number, driver name. User interface is beautiful and ease of use. Nowadays traffic accidents caused by public vans driven too fast and it is hard to dissuade drivers. The system can be developed for further use and can be decreased traffic accidents cause by driving speed over the limit of traffic laws with monitoring and shows public vans speed.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีด้วยความช่วยเหลือจากหลายฝ่ายทั้งในทางตรงและทางอ้อม ปริญญานิพนธ์ฉบับนี้จะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้ ขอขอบคุณ อาจารย์ที่ปรึกษา คือ อาจารย์สรยุทธ กลมกล่อม เป็นผู้คำแนะนำ คำปรึกษา และให้ความช่วยเหลือตลอดการทำโครงการ ซึ่งทำให้การทำงานต่าง ๆ เป็นไปได้อย่างราบรื่น และสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ อาจารย์ และบุคลากรต่าง ๆ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำ และสั่งสอนความรู้ต่าง ๆ มาโดยตลอด รวมทั้งห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (Information Security Advisory Group) และ ห้องวิจัยฮาร์ดแวร์ (Hardware Laboratory) ที่เอื้อเฟื้อสถานที่ในการทำวิจัย และพัฒนาโครงการ

ขอขอบคุณรุ่นพี่หลาย ๆ คน ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำ ให้คำปรึกษาและ แบ่งปันความรู้ในทุก ๆ ด้าน

ขอบคุณพี่สราวุธ ชุนประเสริฐ ที่ได้ให้คำแนะนำ และให้คำปรึกษา แบ่งปันความรู้ในด้านอุปกรณ์ฮาร์ดแวร์ ทำให้โครงการชิ้นนี้สำเร็จได้ด้วยดี

ในท้ายที่สุดนี้ ขอขอบคุณ บิดา มารดา และครอบครัว ที่ได้เลี้ยงดูดูแล สั่งสอน พร้อมทั้งให้โอกาสในการศึกษา และให้กำลังใจเสมอมา

ภวพล ภวปัญญากุล

ธีรัตน์ จิงธนาเจริญเลิศ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ	3
1.3 ประโยชน์ที่คาดว่าจะได้รับ	3
1.4 ขอบเขตของโครงการ	3
1.5 วิธีการดำเนินงาน	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	9
2.1 ทฤษฎี MQTT Protocol	9
2.2 ทฤษฎีพัฒนาอุปกรณ์ฮาร์ดแวร์ Vehicle Tracker	21
2.3 ทฤษฎีพัฒนาระบบ แสดงผลข้อมูลบนแผนที่	33
2.4 เทคโนโลยี Internet of Things	38
บทที่ 3 การวิเคราะห์ออกแบบและพัฒนาระบบ	39
3.1 ความต้องการของระบบ	39
3.2 โครงสร้างในการพัฒนาระบบ	40
บทที่ 4 การทดลองและผลการทดลอง	53
4.1 การรับข้อมูลจากอุปกรณ์ฮาร์ดแวร์และการแสดงผลของรถบนแผนที่ โดยการนำข้อมูลจากฐานข้อมูลพร้อมแสดงรายละเอียดของรถ	53
4.2 การเพิ่มและแก้ไขข้อมูลของรถคู่โดยสารสาธารณะผ่านระบบ Admin	60
4.3 การค้นหารถคู่โดยสารสาธารณะบนแผนที่ด้วยหมายเลขทะเบียนรถ	72

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทสรุปและข้อเสนอแนะ	75
5.1 บทสรุปของโครงการ	75
5.2 ปัญหาและแนวทางแก้ไข	76
5.3 แนวทางการพัฒนาต่อ	76
บรรณานุกรม	88

สารบัญตาราง

ตาราง	หน้า
1.1 สถิติอุบัติเหตุรถโดยสารสาธารณะ ตั้งแต่เดือน ต.ค.54 - มี.ค.56	1
1.2 Gantt Chart ภาคเรียนที่ 1.....	5
1.3 Gantt Chart ภาคเรียนที่ 2.....	7
2.1 เปรียบเทียบการใช้พลังงานของการใช้พลังงานระหว่าง โพรโตคอล HTTPS กับ MQTT ในการรับข้อมูลเข้ามาในมือถือ.....	9
2.2 เปรียบเทียบการใช้พลังงานของการใช้พลังงานระหว่าง โพรโตคอล HTTPS กับ MQTT ในการส่งข้อมูลออกจากมือถือ	10
2.3 รูปแบบของ Fixed header ของ MQTT Frame	11
2.4 ชนิดของข้อความ.....	12
2.5 ระดับของ QoS	13
2.6 ค่าของ Remaining Length	14
2.7 รูปแบบ Protocol Version 3 ของ MQTT Frame.....	14
2.8 รูปแบบ Connect flags ของ MQTT Frame.....	15
2.9 ค่า และ ความหมายของ Connect return code	16
2.10 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 0.....	19
2.11 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 1.....	19
2.12 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 2.....	20
3.1 ความสามารถของระบบในการจัดการรายละเอียดรถตู้โดยสารสาธารณะ	41
3.2 ความสามารถของระบบในการติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ.....	42
3.3 ความสามารถของระบบในการค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน.....	43
3.4 ความสามารถของระบบในการเข้าดูรายละเอียดรถตู้โดยสารสาธารณะ.....	44
3.5 รูปแบบของตาราง Vans ในฐานข้อมูล	45
3.6 รูปแบบของตาราง Drivers ในฐานข้อมูล	45
3.7 รูปแบบของตาราง Van_routes ในฐานข้อมูล.....	46
3.8 รูปแบบของตาราง Van_locations ในฐานข้อมูล	46
3.9 รูปแบบของตาราง Users ในฐานข้อมูล	46

สารบัญรูป

รูป	หน้า
1.1 สถิติอุบัติเหตุรถโดยสารสาธารณะ ตั้งแต่เดือน ต.ค. 54 - มี.ค. 56.....	2
2.1 การสื่อสารในรูปแบบ 1 ต่อ 1	10
2.2 การสื่อสารในรูปแบบ 1 ต่อ หลาย.....	11
2.3 การสื่อสารในรูปแบบ หลาย ต่อ หลาย.....	11
2.4 MQTT Frame ของ Connect message	17
2.5 MQTT Frame ของ Subscribe message	17
2.6 MQTT Frame ของ Suback message	18
2.7 MQTT Frame ของ Publish message.....	18
2.8 Arduino Board.....	21
2.9 ตัวอย่างการเชื่อมต่อ Arduino กับ Computer และ โปรแกรม ArduinoIDE	22
2.10 เลือกบอร์ด Arduino ที่ต้องการ upload	23
2.11 เลือกหมายเลข Comport ของบอร์ด	23
2.12 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง, การ Compile และการ Upload โปรแกรม	24
2.13 Layout และ Pin out ของ Arduino UNO R3 Board.....	24
2.14 หลักการทำงานของ GPS	26
2.15 ภาพรวมการทำงาน GPRS.....	27
2.16 SIM908 Module	28
2.17 GPS กลุ่ม Pin J6 ของ SIM908 Module.....	30
2.18 GSM กลุ่ม Pin J3 ของ SIM908 Module	30
2.19 ขั้วต่อแบตเตอรี่ กลุ่ม Pin J1 ของ SIM908 Module	31
2.20 UART กลุ่ม Pin J8 ของ SIM908 Module	31
2.21 การทำงานของ AJAX.....	37
3.1 ภาพรวมของระบบ	40
3.2 ความสามารถของระบบ	41
3.3 Entity-Relationship	44
3.4 Activity Diagram การจัดการรายละเอียดรถตู้โดยสารสาธารณะ	47
3.5 Activity Diagram การติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ	48
3.6 Activity Diagram การค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน.....	48

สารบัญรูป (ต่อ)

รูป	หน้า
3.7 Activity Diagram การเข้าดูรายละเอียดรถตู้โดยสารสาธารณะ.....	49
3.8 การเชื่อมต่อระหว่าง Arduino กับ SIM908 Module	50
3.9 Flow Chart การทำงานของอุปกรณ์ฮาร์ดแวร์.....	51
4.1 อุปกรณ์ฮาร์ดแวร์ที่ทำการติดตั้งบนรถ	53
4.2 ผลการรัน Laravel Server	54
4.3 หน้าต่าง โปรแกรม MAMP.....	54
4.4 หน้าต่างการทำงานของอุปกรณ์ฮาร์ดแวร์.....	57
4.5 หน้าต่างการรัน submqtt_with_db.php	57
4.6 ข้อมูลในฐานข้อมูลตาราง van_locations.....	58
4.7 หน้าหลักของเว็บไซต์.....	58
4.8 หน้าต่างแสดงข้อมูลของรถตู้โดยสารสาธารณะ.....	60
4.9 ผลการรัน Laravel Server	60
4.10 หน้าต่าง โปรแกรม MAMP.....	61
4.11 หน้า login ระบบ Admin ของเว็บไซต์	61
4.12 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์.....	62
4.13 หน้าต่างเพิ่มข้อมูลรถตู้โดยสารสาธารณะในระบบ	62
4.14 ข้อมูลของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปในระบบ.....	63
4.15 ข้อมูลในฐานข้อมูลตาราง vans	63
4.16 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการเพิ่มข้อมูล	64
4.17 หน้าต่างแก้ไขข้อมูลของรถตู้โดยสารสาธารณะในระบบ	64
4.18 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์ก่อนการแก้ไขข้อมูล	65
4.19 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการแก้ไขข้อมูล	65
4.20 หน้าต่างข้อมูลของรถตู้โดยสารสาธารณะ.....	66
4.21 หน้าต่าง Tab Menu ของระบบ Admin	66
4.22 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์.....	67
4.23 หน้าต่างเพิ่มข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ	68
4.24 ข้อมูลของสายการเดินทางของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปในระบบ.....	68
4.25 ข้อมูลในฐานข้อมูลตาราง van_routes	69

สารบัญรูป (ต่อ)

รูป	หน้า
4.26 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการเพิ่มข้อมูล.....	69
4.27 หน้าต่างแก้ไขข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ	70
4.28 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์ก่อนการแก้ไขข้อมูล	70
4.29 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการแก้ไขข้อมูล	71
4.30 หน้าต่างข้อมูลของรถตู้โดยสารสาธารณะ	71
4.31 หน้าหลักของเว็บไซต์ที่แสดงข้อมูลรถตู้โดยสารสาธารณะทั้งหมด	72
4.32 หน้าหลักของเว็บไซต์ที่มีการใส่หมายเลขทะเบียนของรถตู้โดยสารสาธารณะที่ต้องการค้นหา.....	73
4.33 หน้าหลักของเว็บไซต์ที่ทำการค้นหาสำเร็จ	73
4.34 หน้าหลักของเว็บไซต์ที่ทำการค้นหาไม่สำเร็จ	74

บทที่ 1

บทนำ

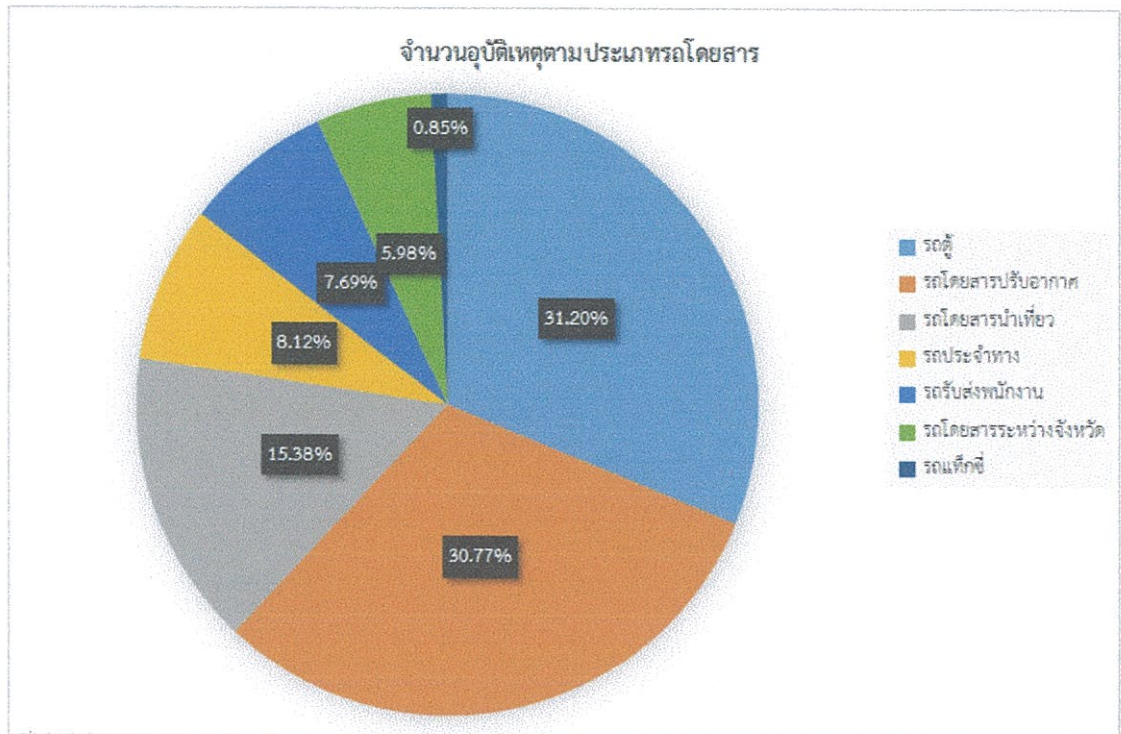
1.1 ที่มาและความสำคัญ

สืบเนื่องจากข้อมูลการจดทะเบียนรถตู้โดยสารสาธารณะของประเทศไทยในปัจจุบันจากกรมการขนส่งทางบก สะสมถึงวันที่ 29 พฤศจิกายน 2555 พบว่า มีจำนวนทั้งหมด 10,623 คัน แบ่งเป็นรถร่วม บขส. จำนวน 5,370 คัน และรถร่วม ขสมก. จำนวน 5,253 คัน ซึ่งจากสถิติ เดือน ตุลาคม 2554 ถึง มีนาคม 2556 ผลปรากฏว่า รถตู้โดยสารสาธารณะ และรถโดยสารปรับอากาศ เกิดอุบัติเหตุบนท้องถนนซึ่งก่อให้เกิดผู้บาดเจ็บ และเสียชีวิตสูงสุด แสดงให้เห็นว่าอุบัติเหตุที่เกิดจากรถตู้โดยสารสาธารณะทำให้เกิดการสูญเสียมากที่สุดเมื่อเทียบกับรถโดยสารชนิดอื่น ๆ โดยอุบัติเหตุส่วนใหญ่่นั้นเกิดจากการขับรถเร็วเกินกว่าที่กฎหมายกำหนด โดยจากการรวบรวมข้อมูลของศูนย์พิทักษ์สิทธิผู้บริโภค มูลนิธิเพื่อผู้บริโภค ในเดือนตุลาคม 2554 ถึงมีนาคม 2556 พบว่า “รถตู้โดยสารสาธารณะ” ซึ่งเป็นยานพาหนะยอดนิยมของคนยุคปัจจุบัน มีสถิติการเกิดอุบัติเหตุสูงสุดเฉลี่ย 6 ศพต่อเดือน ดังตาราง 1.1

ตาราง 1.1 สถิติอุบัติเหตุรถโดยสารสาธารณะ ตั้งแต่เดือน ต.ค. 54 - มี.ค. 56

ประเภทรถโดยสาร	จำนวนอุบัติเหตุ (ครั้ง)	บาดเจ็บ	เสียชีวิต	ร้อยละ
รถตู้โดยสารสาธารณะ	73	594	105	31.20
รถโดยสารปรับอากาศ	72	1672	118	30.77
รถโดยสารนำเที่ยว	36	846	29	15.38
รถประจำทาง	19	61	8	8.12
รถรับส่งพนักงาน	18	248	18	7.69
รถโดยสารระหว่างจังหวัด	14	266	12	5.98
รถแท็กซี่	2	5	2	0.85
รวม	234	3692	292	100

จากตารางข้างต้นสามารถสรุปออกมาเป็นกราฟ ดังรูป 1.1



รูป 1.1 สถิติอุบัติเหตุรถโดยสารสาธารณะ ตั้งแต่เดือน ต.ค. 54 - มี.ค. 56

ซึ่งการตรวจจับ หรือห้ามปรามการขับเร็วเกินกว่าที่กฎหมายกำหนดนั้นเป็นไปได้ยาก แม้จะมีมาตรการตรวจจับด้วยกล้องตรวจจับความเร็ว หรือตั้งด่านตรวจตามจุดต่าง ๆ แต่ยังไม่สามารถตรวจจับความเร็วของรถที่ขับในเส้นทางไกลได้ตลอดทาง จึงทำให้ผู้ขับรถตู้โดยสารสาธารณะส่วนมากใช้จุดนี้ในการหารายได้เพิ่มขึ้นให้ตนเอง โดยไม่ได้คำนึงถึงความเสี่ยงที่อาจเกิดขึ้นจากอุบัติเหตุ ทำให้เกิดอุบัติเหตุอยู่บ่อยครั้ง และมีความสูญเสียมากมายตามมา

ปัจจุบันเทคโนโลยีมีความก้าวหน้า และมีการทำอุปกรณ์ GPS นำทางติดรถยนต์ หรือโทรศัพท์มือถือในการใช้ GPS นำทาง ประกอบกับการศึกษาโปรโตคอล Message Queue Telemetry Transport (MQTT) ที่เป็นโปรโตคอลในการรับ-ส่งข้อมูลที่ใช้ Bandwidth และใช้พลังงานน้อย ทำให้สามารถรับ-ส่งข้อมูลพิกัด GPS และข้อมูลรถตู้โดยสารสาธารณะ โดยใช้อุปกรณ์ฮาร์ดแวร์ที่มีการติดตั้งไว้ที่รถตู้โดยสารสาธารณะ เพื่อระบุตำแหน่งของรถตู้โดยสารสาธารณะบนแผนที่ พร้อมกับสามารถตรวจจับความเร็วของรถตู้โดยสารสาธารณะนั้นได้ตลอดทาง นอกจากนี้ยังสามารถแสดงข้อมูลคนขับ และทะเบียนของรถตู้โดยสารสาธารณะนั้นบนแผนที่ได้

1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อเตรียมตำแหน่งของรถตู้สำหรับส่งต่อให้กับแอปพลิเคชันที่ใช้ป้องกันการที่รถตู้โดยสารสาธารณะขับรถเร็วเกินกฎหมาย
- 2) เพื่อให้ทราบตำแหน่งของรถตู้โดยสารสาธารณะ
- 3) เพื่อเป็นต้นแบบในการนำ MQTT Protocol มาประยุกต์ใช้
- 4) เพื่อศึกษาการทำงานของบอร์ดไมโครคอนโทรลเลอร์ ร่วมกับ GPRS/GPS Module

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เข้าใจการทำงานของโปรโตคอล MQTT ด้วยบอร์ดไมโครคอนโทรลเลอร์ ซึ่งเป็นโปรโตคอลที่นิยมนำมาใช้กับ IoT ในปัจจุบันเนื่องจากเป็นโปรโตคอลที่ใช้ Bandwidth น้อย และประหยัดพลังงาน
- 2) ลดโอกาสการเกิดความสูญเสียทั้งทรัพย์สิน และชีวิตจากการโดยสารรถตู้โดยสารสาธารณะอาศัยการ Monitoring ความเร็ว
- 3) สามารถทราบตำแหน่ง ณ ปัจจุบันของรถตู้โดยสารสาธารณะ
- 4) เข้าใจการทำงานของ IoT

1.4 ขอบเขตของโครงการ

สร้างระบบติดตามรถตู้โดยสารสาธารณะด้วยการแสดงผลผ่านแผนที่ออนไลน์ ซึ่งเป็นระบบที่ออกแบบมาเพื่อเพิ่มความปลอดภัยแก่ผู้ที่ต้องการเดินทางด้วยรถตู้โดยสารสาธารณะ โดยเป็นระบบที่นำเทคโนโลยี GPRS/GPS มาใช้ในการติดตาม ซึ่งระบบจะมีการทำงานดังนี้

- 1) นำ SIM908 Module กับ Arduino Board มาทำการเชื่อมต่อกัน เพื่อให้อุปกรณ์สามารถสื่อสารกันได้ด้วย AT Command
- 2) สร้าง MQTT Packet ที่มีข้อมูล GPS ซึ่งได้รับข้อมูลผ่านส่วน GPS ของ SIM908 Module และข้อมูลรถตู้โดยสารสาธารณะ โดยใช้ AT Command
- 3) ส่งข้อมูลผ่านโปรโตคอล MQTT โดยสร้างอุปกรณ์ฮาร์ดแวร์ ด้วยการใช้ Arduino Board ประกอบกับ SIM908 Module เพื่อ Publish Packet ไปยัง MQTT Broker
- 4) ให้ Apache เซิร์ฟเวอร์ทำการ Subscribe MQTT Packet จาก MQTT Broker เก็บข้อมูลลงฐานข้อมูล
- 5) นำข้อมูลจากฐานข้อมูลมาใช้แสดงผลเส้นทางการเดินทาง รวมถึงความเร็วของรถตู้โดยสารสาธารณะ และข้อมูลต่าง ๆ เช่น เลขทะเบียนรถตู้โดยสารสาธารณะ ข้อมูลคนขับของรถตู้โดยสารสาธารณะแต่ละคันที่ทำการติดอุปกรณ์ฮาร์ดแวร์ นี้บนแผนที่ได้

- 6) ผู้ใช้งานสามารถดูข้อมูลรถตู้โดยสารสาธารณะทุกสายที่อยู่ในขอบเขตที่กำหนด
 - 7) สามารถค้นหารถตู้โดยสารสาธารณะที่อยู่ในแผนที่จากหมายเลขทะเบียนรถ และแสดงข้อมูลคนขับรถตู้โดยสารสาธารณะ และความเร็วของตู้โดยสารสาธารณะที่ทำการค้นหาได้
- โดยอุปกรณ์ฮาร์ดแวร์ ที่ทำการพัฒนาขึ้นมา นั้นเป็นแค่อุปกรณ์ต้นแบบ เพื่อการศึกษาการทำงานของโปรโตคอล MQTT มีข้อจำกัดในการส่งข้อมูล โดยไม่สามารถส่งข้อมูลออกไปได้หากไม่มีสัญญาณมือถือ หรือทำการรับค่า GPS จากดาวเทียมไม่ได้ ซึ่งมีผลกระทบให้การนำข้อมูล GPS ที่มาใช้ในการแสดงผล หรือการคำนวณความเร็วอาจยังไม่มีความแม่นยำไม่ถึงหนึ่งร้อยเปอร์เซ็นต์ เมื่อเทียบกับความเป็นจริง

1.5 วิธีการดำเนินงาน

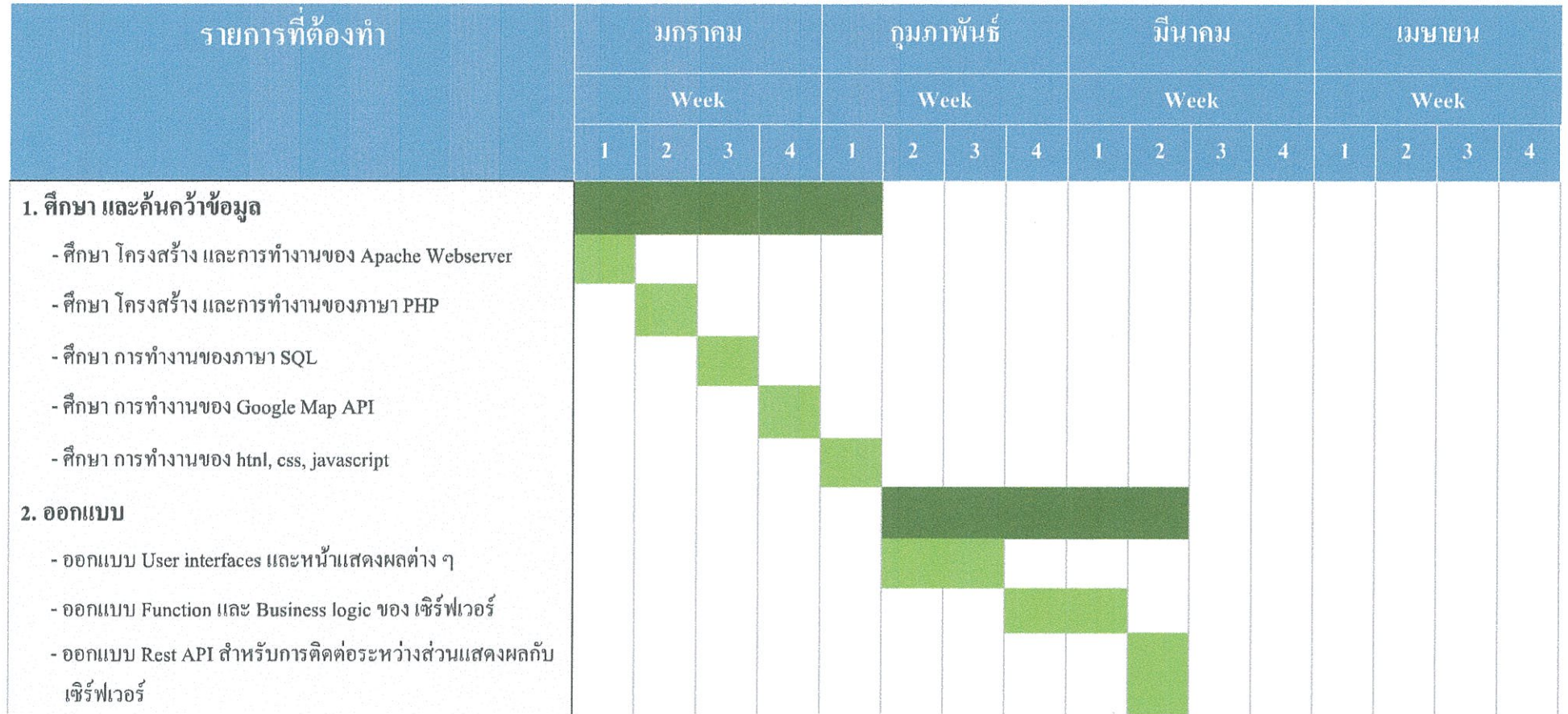
ตาราง 1.2 Gantt Chart ภาคเรียนที่ 1

รายการที่ต้องทำ	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	Week				Week				Week				Week			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษา และค้นคว้าข้อมูล																
- ศึกษา การทำงานของ MQTT Protocol																
- ศึกษา การทำงานของ GPRS/GPS Module กับ บอร์ด ไมโครคอนโทรลเลอร์																
- สั่งซื้อ GPRS/GPS Module																
- ศึกษา AT Command ที่ใช้สำหรับติดต่อ GPRS/GPS Module กับ บอร์ด Arduino																
- ศึกษาการเขียนโปรแกรมภาษา C++ บนบอร์ด Arduino																
- กำหนด Scope งานทั้งหมด																

ตาราง 1.2 Gantt Chart ภาคเรียนที่ 1 (ต่อ)

รายการที่ต้องทำ	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	Week				Week				Week				Week			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
2. ออกแบบ - ออกแบบ System Diagram ของระบบ - ออกแบบ Package ของ Data ที่ใช้ติดต่อระหว่าง บอร์ด Arduino กับ เซิร์ฟเวอร์ - ออกแบบ Usecase Diagram - ออกแบบ ER Diagram ของ Database ใน เซิร์ฟเวอร์ - ออกแบบ Sequence Diagram ของระบบ																
3. เขียนโปรแกรม และทำการทดลอง - เขียนโปรแกรมเพื่อรับข้อมูล GPS จาก GPRS/GPS Module ด้วย บอร์ด Arduino - เขียนโปรแกรมเพื่อ publish ข้อมูล GPS ผ่าน IP ที่ได้จาก GPRS/GPS Module ด้วย บอร์ด Arduino - เขียนโปรแกรม PHP ทำการ subscribe และแสดงผลข้อมูล GPS จาก MQTT Broker ที่ publish มาจาก บอร์ด Arduino																

ตาราง 1.3 Gantt Chart ภาคเรียนที่ 2



ตาราง 1.3 Gantt Chart ภาคเรียนที่ 2 (ต่อ)

รายการที่ต้องทำ	มกราคม				กุมภาพันธ์				มีนาคม				เมษายน							
	Week				Week				Week				Week							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
3. เขียนโปรแกรม และทำการทดลอง - เขียนโปรแกรมด้วย html, css, javascript ในส่วนการแสดงผลให้สวยงาม และใช้งานง่าย - เขียนโปรแกรมด้วยภาษา PHP เพื่อนำข้อมูลที่ได้จากอุปกรณ์ฮาร์ดแวร์เก็บลงในฐานข้อมูล - เขียนโปรแกรมด้วยภาษา PHP เพื่อนำข้อมูลจากฐานข้อมูลมาแสดงผลในหน้าแสดงผล - ทดลองการแก้ไข และเพิ่มข้อมูลรหัสโดยสารสาธารณะในฐานข้อมูล - ทดลองการรับข้อมูลจากอุปกรณ์ฮาร์ดแวร์ และแสดงข้อมูลบนแผนที่																				

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎี MQTT Protocol

2.1.1 คุณสมบัติของ MQTT Protocol

Message Queuing Telemetry Transport (MQTT) เป็นโปรโตคอลที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (Machine-To-Machine) ใช้โปรโตคอล TCP/IP เป็นพื้นฐานในการสื่อสารผ่านระบบเครือข่าย เพื่อสนับสนุนเทคโนโลยี IoT (Internet of Things) ทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่าง ๆ ได้โดยผ่านเครือข่ายอินเทอร์เน็ต และมีการใช้ทรัพยากรในการทำงานน้อยซึ่งถูกออกแบบมาให้เหมาะสำหรับกับงานที่มีข้อจำกัดต่าง ๆ เช่น ระบบที่มี Bandwidth และพลังงานในการติดต่อสื่อสารต่ำ หรือมีข้อจำกัดด้านความเร็วในการประมวลผล และพื้นที่ที่ใช้เก็บข้อมูล ซึ่งขนาดของ MQTT Message มีขนาดสูงสุดแค่ 256MB ทำให้การใช้พลังงานในการรับข้อมูลเข้ามาในมือถือ และการส่งข้อมูลออกจากมือถือน้อยกว่า โปรโตคอล HTTPS ดังตาราง 2.1 และ 2.2

ตาราง 2.1 เปรียบเทียบการใช้พลังงานของการใช้พลังงานระหว่างโปรโตคอล HTTPS กับ MQTT ในการรับข้อมูลเข้ามาในมือถือ¹

	3G		Wi-Fi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	16.13%	3.45%	4.23%
Messages / Hour	1708	160278	3628	263314
% Battery / Message *	0.01709	0.00010	0.00095	0.00002
Messages Received	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024

¹ Stephen Nicolas. 2555. **Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android.** [Online]. Available : <http://stephendnicholas.com/archives/1217>

ตาราง 2.2 เปรียบเทียบการใช้พลังงานของการใช้พลังงานระหว่างโปรโตคอล HTTPS กับ MQTT ในการส่งข้อมูลออกจากมือถือ²

	3G		Wi-Fi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.79%	17.80%	5.44%	3.66%
Messages / Hour	1926	21685	5229	23184
% Battery / Message *	0.00975	0.00082	0.00104	0.00016

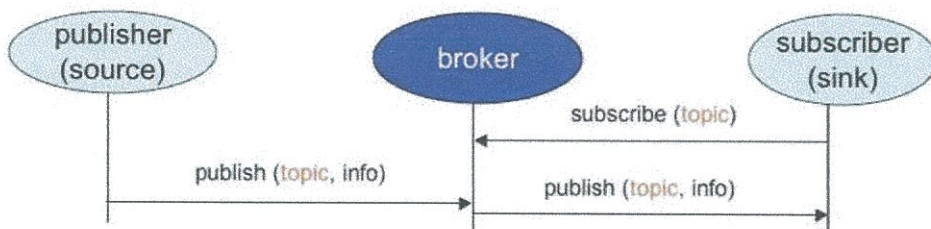
โดยโปรโตคอล MQTT นั้นจะประกอบด้วย 3 ส่วนหลักคือ

Publisher: ทำหน้าที่คอยส่งข้อมูลไปยัง Topic นั้น ๆ

Broker: ทำหน้าที่เป็นตัวกลางคอยจัดการกับข้อความ โดยอ้างอิงจาก Topic

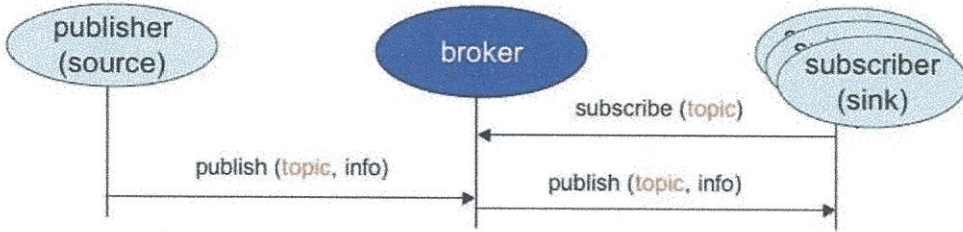
Subscriber: ทำหน้าที่คอยดูการเปลี่ยนแปลงของ Message ที่อ้างอิงด้วย Topic และมีการสื่อสารได้ 3 รูปแบบ ได้แก่

- 1) One to One, 1:1 คือ การสื่อสารในรูปแบบ 1 ต่อ 1 ดังรูป 2.1
- 2) One to Many, 1: M คือ การสื่อสารในรูปแบบ 1 ต่อ หลาย ดังรูป 2.2
- 3) Many to Many, M: N คือ การสื่อสารในรูปแบบ หลาย ต่อ หลาย ดังรูป 2.3

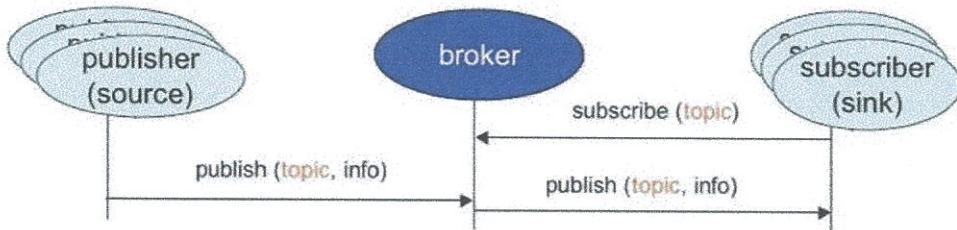


รูป 2.1 การสื่อสารในรูปแบบ 1 ต่อ 1

² Stephen Nicolas. 2555. **Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android.** [Online]. Available : <http://stephendnicholas.com/archives/1217>



รูป 2.2 การสื่อสารในรูปแบบ 1 ต่อ หลาย



รูป 2.3 การสื่อสารในรูปแบบ หลาย ต่อ หลาย

2.1.2 รูปแบบข้อความของโปรโตคอล MQTT

2.1.2.1 Fixed header

เฮดเดอร์ของข้อความแต่ละข้อความที่ส่งผ่านโปรโตคอล MQTT จะมีความยาว 2 ไบต์ซึ่งจะมีส่วนประกอบ และรูปแบบ ดังตาราง 2.3

ตาราง 2.3 รูปแบบของ Fixed header ของ MQTT Frame

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level	RETAIN	
byte 2	Remaining Length							

โดย ไบต์ที่ 1 จะบ่งบอกถึงชนิดของข้อความ และ flags ต่าง ๆ (DUP, QoS level, and RETAIN) ส่วนไบต์ที่ 2 จะบ่งบอกถึงจำนวนไบต์ที่เหลือของข้อมูล

Message Type: ใช้สำหรับบอกชนิดของข้อความ โดยมีชนิด ดังตาราง 2.4
ตำแหน่ง: ไบต์ที่ 1, บิตที่ 7-4

ตาราง 2.4 ชนิดของข้อความ

Mnemonic	Enumeration	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

DUP flag: เป็น flag ที่จะถูกกำหนดเมื่อจะมีการส่ง PUBLISH, PUBREL, SUBSCRIBE หรือ UNSUBSCRIBE ระหว่าง Client กับ เซิร์ฟเวอร์ซึ่งจะให้กับข้อความที่มี QoS ระดับที่มากกว่า 0 และต้องจำเป็นต้องมี Acknowledgment เมื่อมีการกำหนด DUP flag ในส่วนของ Variable Header ต้องมี Message ID ซึ่งจะทำให้ฝั่งผู้รับที่รับข้อความที่มีการกำหนด DUP flag สามารถตรวจสอบได้ว่าการรับข้อความซ้ำหรือไม่

ตำแหน่ง: ไบต์ที่ 1, บิตที่ 3

QoS: เป็น flag ใช้ในการวัดระดับของ การรับประกันการส่งข้อความ PUBLISH ซึ่งระดับของ QoS จะมี 3 ระดับ ดังตาราง 2.5

ตำแหน่ง: ไบต์ที่ 1, บิตที่ 2-1

ตาราง 2.5 ระดับของ QoS

QoS value	bit 2	bit 1	Description		
0	0	0	At most once	Fire and Forget	≤ 1
1	0	1	At least once	Acknowledged delivery	≥ 1
2	1	0	Exactly once	Assured delivery	$= 1$
3	1	1	Reserved		

RETAIN: เป็น flag ที่ใช้สำหรับข้อความ PUBLISH โดยเมื่อ Client ส่งข้อความ PUBLISH ไปยังเซิร์ฟเวอร์ถ้าหาก flag นี้ถูกกำหนดค่าเป็น 1 เซิร์ฟเวอร์จะทำการเก็บข้อความนั้นไว้ หลังจากการส่งข้อความนั้นให้กับ Subscriber เมื่อมีการ Subscribe อีกครั้ง ก็จะมีการส่งข้อความ retained message ที่ถูกเก็บไว้ไปให้ Subscriber ซึ่งถ้าหาก Retain flag ไม่ถูกกำหนดค่า ก็จะไม่มีการเก็บข้อความไว้เลย ทำให้ Subscriber ไม่ได้รับข้อความอะไรไป สำหรับ retained message จะถูกเก็บไว้ใน เซิร์ฟเวอร์จนกว่า เซิร์ฟเวอร์จะ restart หรือมีการส่ง retained message ที่มีขนาดความยาวของ payload เป็น 0 ใน Topic นั้น ๆ

ตำแหน่ง: ไบต์ที่ 1, บิตที่ 0

Remaining Length: จะบ่งบอกถึงจำนวนไบต์ที่เหลือของข้อมูล ซึ่งประกอบไปด้วย ข้อมูล 2 ประเภทได้แก่ variable header และ payload โดยการเก็บข้อมูลเป็นแบบ Big-endian หรือเรียงจากไบต์ที่มีนัยสำคัญสูงไปหาไบต์ที่มีนัยสำคัญต่ำ ซึ่งข้อมูลในฟิลด์นี้จะใช้ข้อมูล 1 ไบต์ สำหรับบอกความยาวของข้อความ 127 ไบต์ โดยแต่ละไบต์จะใช้ 7 บิตแรกเพื่อเก็บข้อมูล และใช้บิตสุดท้ายเพื่อ

บอกว่า มีข้อมูลต่อจากไบต์ นี้หรือไม่ ซึ่งบิตที่ 8 นี้ถูกเรียกว่า “continuation bit” ซึ่งโปรโตคอล MQTT จะสามารถใช้จำนวนไบต์ ในการแสดงความยาวของข้อความ มากที่สุด 4 ไบต์ หรือแทนขนาดความยาวของข้อความได้ 268,435,456 ไบต์ หรือ 256 MB ดังตาราง 2.6
ตำแหน่ง: ไบต์ที่ 2, บิตที่ 0

ตาราง 2.6 ค่าของ Remaining Length

Digits	From	To
1	0 (0x00)	127 (0x7F)
2	128 (0x80, 0x01)	16 383 (0xFF, 0x7F)
3	16 384 (0x80, 0x80, 0x01)	2 097 151 (0xFF, 0xFF, 0x7F)
4	2 097 152 (0x80, 0x80, 0x80, 0x01)	268 435 455 (0xFF, 0xFF, 0xFF, 0x7F)

2.1.2.2 Variable header

เซกเตอร์ที่มีขนาดไม่คงที่ซึ่งจะอยู่ระหว่าง fixed header กับ payload ซึ่งจะประกอบด้วย

Protocol name: อยู่ในข้อความ CONNECT โดยฟิลด์นี้จะเข้ารหัสด้วย UTF-8 เพื่อบ่งบอกชื่อของโปรโตคอล

Protocol version: อยู่ในข้อความ CONNECT โดยฟิลด์นี้ใช้ 1 ไบต์ในการแสดงเวอร์ชันของโปรโตคอลที่ Client ใช้งานโดยเวอร์ชันปัจจุบันจะมีการกำหนดให้เป็นเวอร์ชัน 3 (0x03) ดังตาราง 2.7

ตาราง 2.7 รูปแบบ Protocol Version 3 ของ MQTT Frame

bit	7	6	5	4	3	2	1	0
	Protocol Version							
	0	0	0	0	0	0	1	1

Connect flags: อยู่ในข้อความ CONNECT มีขนาด 1 ไบต์โดยจะบ่งบอกถึงค่าต่าง ๆ ดังตาราง 2.8

ตาราง 2.8 รูปแบบ Connect flags ของ MQTT Frame

bit	7	6	5	4	3	2	1	0
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved
								x

- 1) Clean session flag: ใช้กำหนดค่าว่าเป็นการ “Clean Start” ซึ่งถ้ากำหนดค่าเป็น 0 เซิร์ฟเวอร์จะทำการเก็บ subscriptions ของ Client ไว้หลังจากที่ Client ยกเลิกการติดต่อไปแล้ว แต่ถ้าหากกำหนดค่าเป็น 1 เซิร์ฟเวอร์จะทำการเคลียร์ข้อมูล ที่เก็บไว้ก่อนหน้าทิ้งไป
- 2) Will flag: ใช้กำหนดค่าว่าจะมีการเก็บ WillMessage ไว้ใน เซิร์ฟเวอร์โดยเมื่อมีปัญหาของการเชื่อมต่อระหว่าง Client กับ เซิร์ฟเวอร์จะมีการส่ง WillMessage ไปให้ Subscriber ซึ่งเมื่อค่านี้นี้ถูกกำหนดค่าเป็น 1 ค่าของ WillQoS และ WillRetain ในส่วนของ Connect flags byte ต้องมีการกำหนดค่าด้วย นอกจากนี้ WillTopic และ WillMessage ในส่วนของ payload ต้องถูกกำหนดค่าเช่นกัน
- 3) Will QoS : ใช้กำหนดค่าระดับของ QoS สำหรับ Will message ซึ่งฟิลด์นี้มีค่า 2 บิต โดยค่าของ WillQoS จะเป็น 0 (0x00), 1 (0x01) หรือ 2 (0x02) โดยแต่ละระดับจะอธิบายในหัวข้อ 2.1.3
- 4) Will Retain flag: ใช้กำหนดค่าว่าจะให้ เซิร์ฟเวอร์ทำการเก็บ Will message ซึ่งถ้ากำหนดค่านี้เป็น 1 เซิร์ฟเวอร์จะทำการเก็บ Will message ไว้ แต่ถ้ากำหนดค่านี้เป็น 0 เซิร์ฟเวอร์จะไม่ทำการเก็บ Will message ไว้คือ เมื่อมีการ Subscribe แล้ว Will message นั้นจะถูกลบออกจาก Queue
- 5) User name and password flags: ใช้กำหนดค่าว่าต้องมีการระบุ username และ password ในการสร้างการเชื่อมต่อหรือไม่ ซึ่งถ้ากำหนดค่าเป็น 1 แสดงว่าต้องมีการกำหนดค่าของ username และ password โดย username และ password จะอยู่ในส่วน payload ของข้อความ CONNECT

Keep Alive timer: อยู่ในข้อความ CONNECT มีขนาด 2 ไบต์ โดยจะใช้บอกถึงช่วงเวลาที่มากที่สุดที่จะมีการส่งข้อมูลจาก Client โดยกำหนดได้มากที่สุดคือ 18 ชั่วโมง หรือ 65536 วินาที ซึ่งถ้าไม่มีการส่งข้อความจาก Client ในระยะเวลาที่กำหนดไว้ แสดงว่าการเชื่อมต่อมีปัญหา

Connect return code: อยู่ในข้อความ CONACK มีขนาด 1 ไบต์ ซึ่งใช้สำหรับตอบกลับเพื่อรายงานผลของการเชื่อมต่อซึ่งมีค่าต่าง ๆ ตามตาราง 2.9

ตาราง 2.9 ค่า และ ความหมายของ Connect return code

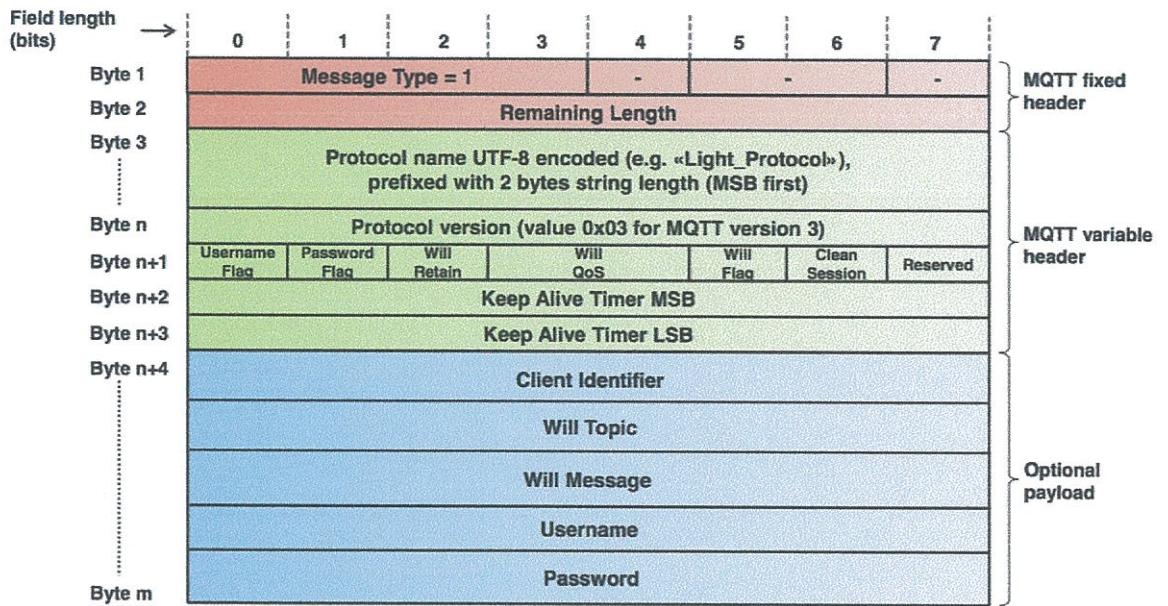
Enumeration	HEX	Meaning
0	0x00	Connection Accepted
1	0x01	Connection Refused: unacceptable protocol version
2	0x02	Connection Refused: identifier rejected
3	0x03	Connection Refused: server unavailable
4	0x04	Connection Refused: bad user name or password
5	0x05	Connection Refused: not authorized
6-255		Reserved for future use

Topic name: อยู่ในข้อความ PUBLISH โดยใช้เพื่อบ่งบอกถึงช่องทางในการส่งข้อมูลที่อยู่ในส่วนของ Payload เพื่อให้ Subscriber มา Subscribe ข้อมูลภายใต้ Topic เดียวกัน โดยในส่วนนี้จะเข้ารหัสด้วย UTF-8 และมีค่าได้สูงสุด 32,767 ตัวอักษร

2.1.2.3 Payload

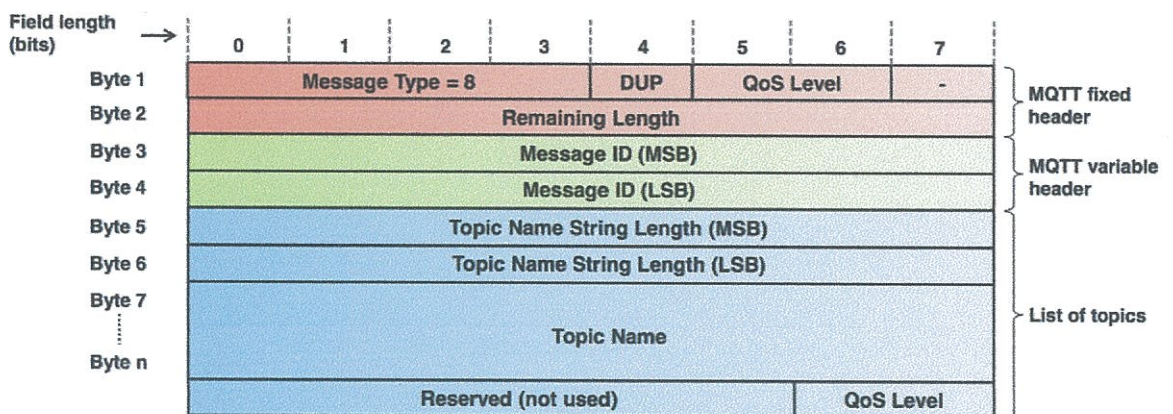
คือ ส่วนของข้อมูลที่จะถูกขนส่ง โดยในแต่ละ message จะมี payload ที่แตกต่างกันไป

CONNECT: ในส่วนของ payload จะประกอบด้วยข้อความที่เข้ารหัสด้วย UTF-8 โดยจะใช้กำหนดค่ามีลักษณะของเฟรมดังรูป 2.4 ซึ่งประกอบด้วย



รูป 2.4 MQTT Frame ของ Connect message

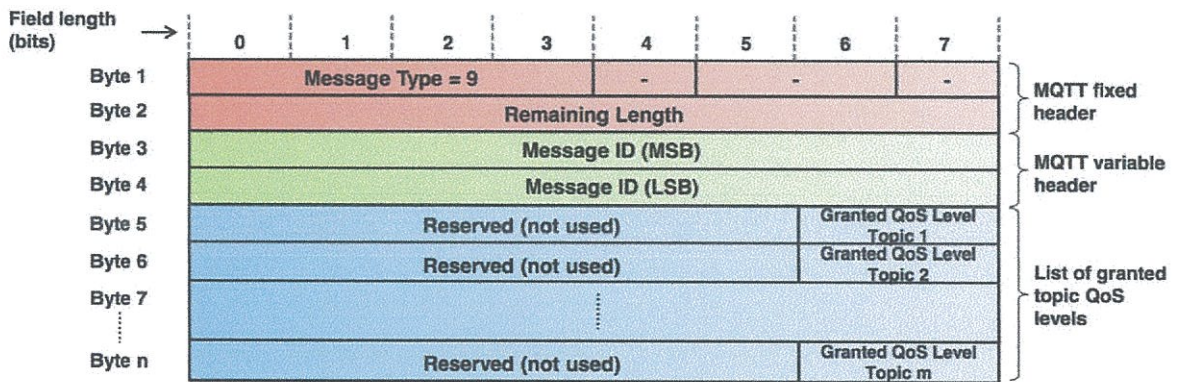
- 1) Client Identifier : มีความยาว 1 ถึง 23 ตัวอักษรเพื่อใช้บ่งบอกตัวตนของ Client ที่ทำการเชื่อมต่อกับ เซิร์ฟเวอร์ โดย Client แต่ละตัวต้องมีค่าที่ไม่ซ้ำกัน
 - 2) WillTopic : Topic ที่จะทำให้การส่ง WillMessage ไปให้ Subscriber
 - 3) WillMessage : ข้อความที่จะทำการส่งไปให้ Subscriber เมื่อการเชื่อมต่อระหว่าง Client กับ เซิร์ฟเวอร์มีปัญหา
 - 4) Username, Password : Username และ Password ที่ใช้ในการเชื่อมต่อกับ เซิร์ฟเวอร์ ซึ่งค่าต่าง ๆ ใน payload นั้นจะขึ้นอยู่กับที่กำหนด Connect flags ต่าง ๆ บน Variable header ด้วย
- SUBSCRIBE: ในส่วนของ payload จะประกอบด้วยข้อความที่เข้ารหัสด้วย UTF-8 และมีลักษณะของเฟรมดังรูป 2.5 ซึ่งประกอบไปด้วย



รูป 2.5 MQTT Frame ของ Subscribe message

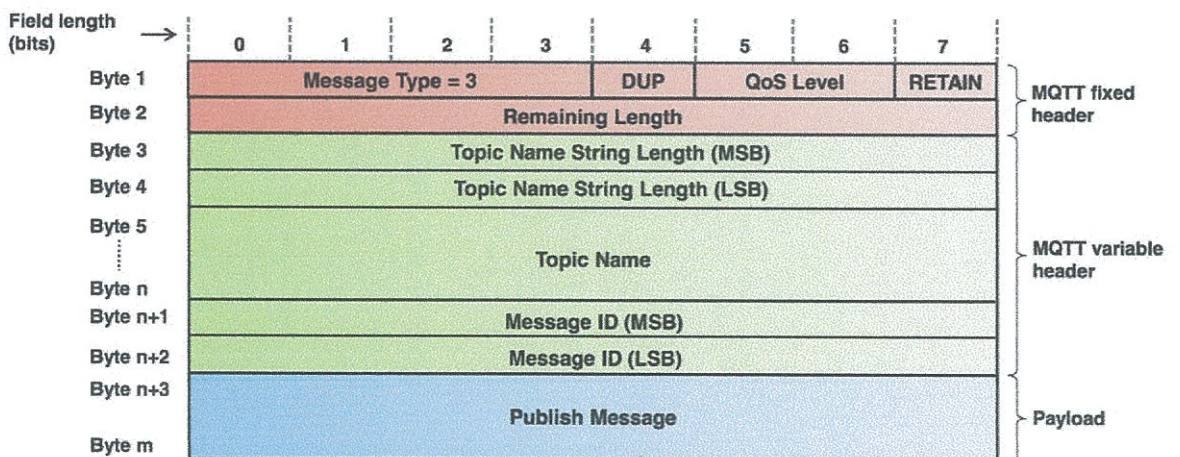
- 1) Topic Name String Length : มีขนาด 2 ไบต์โดยเป็นตัวบ่งบอกความยาวของ Topic Name ในไบต์ถัดไป
- 2) Topic Name : Topic ที่ Subscriber ต้องการทำการ Subscribe
- 3) QoS Level: มีขนาด 2 บิตใช้เพื่อบ่งบอกระดับของ QoS

SUBACK: เป็นข้อความตอบกลับจาก เซิร์ฟเวอร์เมื่อมีการส่งข้อความ SUBSCRIBE ไปให้ เซิร์ฟเวอร์โดย ใน payload จะใช้ 2 บิตเป็นตัวบ่งบอกว่า เซิร์ฟเวอร์จะยินยอมให้ Subscriber ทำการ Subscribe ในแต่ละ Topic ได้ใน QoS ระดับใด และมีลักษณะของเฟรมดังรูป 2.6



รูป 2.6 MQTT Frame ของ Suback message

PUBLISH: ในส่วนของ payload จะเป็นข้อความที่ Client ต้องการส่งเข้าไปยัง Queue ของ เซิร์ฟเวอร์ และส่วนของชื่อ Topic ที่ต้องการส่งเข้าไปจะอยู่ในส่วนของ Variable Header และมีลักษณะของเฟรมดังรูป 2.7



รูป 2.7 MQTT Frame ของ Publish message

2.1.3 ระดับของคุณภาพในการให้บริการ (QoS Level) และ Flow การทำงาน

2.1.3.1 QoS ระดับ 0

QoS ระดับ 0 ไม่รับประกันว่าข้อมูลส่งถึงปลายทางหรือไม่ และมีการส่งเพียงครั้งเดียว (At most once delivery) ซึ่งมีการทำงานดังตาราง 2.10

ตาราง 2.10 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 0

Client	Message and direction	Server
QoS = 0	PUBLISH ----->	Action: Publish message ไปยัง subscribers

2.1.3.2 QoS ระดับ 1

QoS ระดับ 1 มีการรับประกันว่าข้อมูลส่งถึงปลายทางแน่นอนแต่อาจมีการส่งหลายครั้ง และปลายทางอาจได้รับข้อมูลซ้ำ (At least once delivery) ซึ่งมีการทำงานดังตาราง 2.11

ตาราง 2.11 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 1

Client	Message and direction	Server
QoS = 1 DUP = 0 Message ID = x	PUBLISH ----->	Action: เก็บ message ลงในหน่วยความจำ Publish message ไปยัง subscribers ลบ message ออกจากหน่วยความจำ
Action: เก็บ message ลงในหน่วยความจำ		
Action: ลบ message ออกจากหน่วยความจำ	PUBACK <-----	

2.1.3.3 QoS ระดับ 2

QoS ระดับ 2 มีการรับประกันว่าข้อมูลส่งถึงปลายทางแน่นอนมีการส่งข้อมูลเพียงครั้งเดียว และปลายทางไม่ได้รับข้อมูลซ้ำ (Exactly once delivery) ซึ่งมีการทำงานดังตาราง 2.12

ตาราง 2.12 การทำงานของโปรโตคอล MQTT ใน QoS ระดับ 2

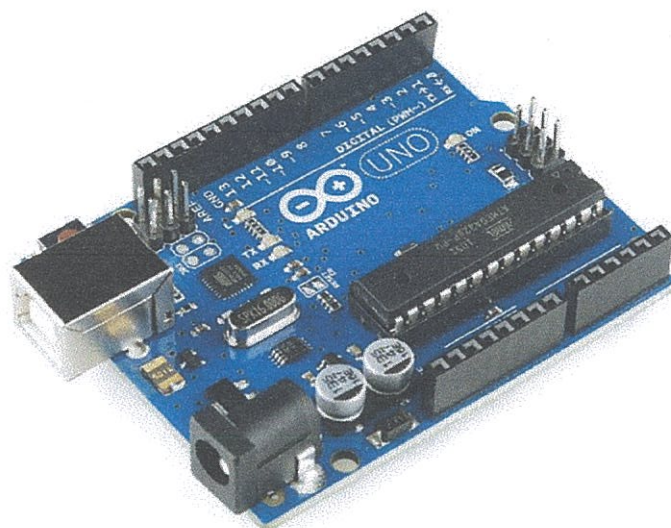
Client	Message and direction	Server
QoS = 2 DUP = 0 Message ID = x Action: เก็บ message ลงในหน่วยความจำ	PUBLISH ----->	Action: เก็บ message ลงในหน่วยความจำ หรือ Actions: เก็บ message ID ลงในหน่วยความจำ Publish message ไปยัง subscribers
	PUBREC <-----	Message ID = x
Message ID = x	PUBREL ----->	Actions: Publish message ไปยัง subscribers ลบ message ออกจากหน่วยความจำ หรือ Action: ลบ message ID ออกจากหน่วยความจำ
Action: ลบ message ออกจากหน่วยความจำ	PUBCOMP <-----	Message ID = x

2.2 ทฤษฎีพัฒนาอุปกรณ์ฮาร์ดแวร์ Vehicle Tracker

2.2.1 Arduino Board

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์ และซอฟต์แวร์ ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย และราคาถูก ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติมพัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่าง ๆ คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่าง ๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย หรือนำ Module มาเชื่อมต่อผ่านสายไฟ



รูป 2.8 Arduino Board

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

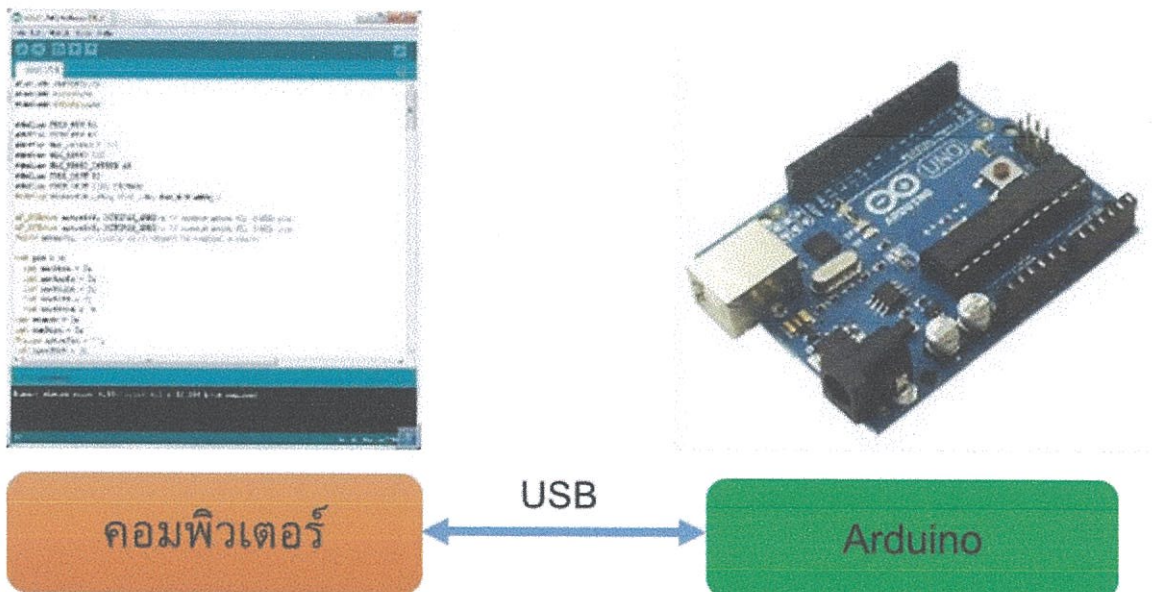
- 1) ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- 2) มี Library ให้เลือกใช้หลากหลาย
- 3) ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน

- 4) ราคาไม่แพง
- 5) Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้
- 6) หาก MCU เสีย สามารถเปลี่ยนได้ง่าย
- 7) มีพอร์ตให้ใช้งานได้ง่าย

ข้อจำกัดของ Arduino board

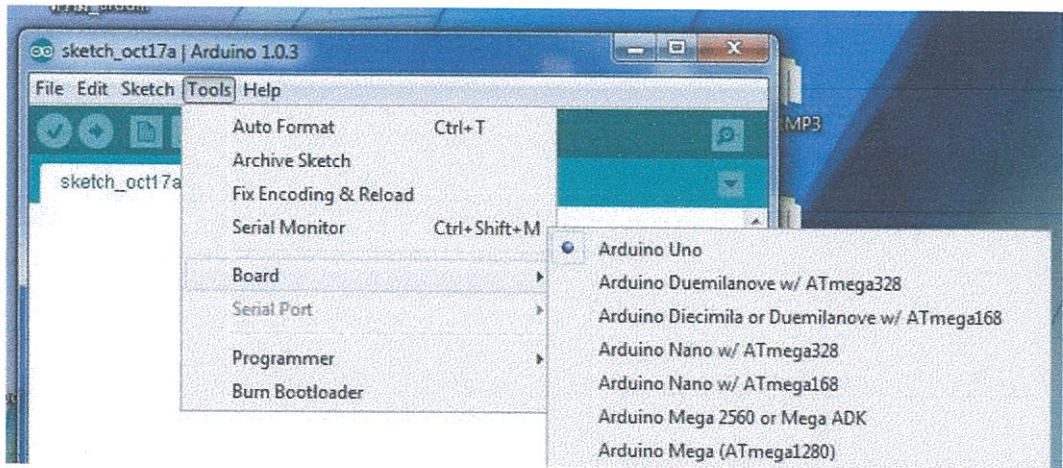
- 1) Memory มีขนาดเล็กน้อย
- 2) ตัวบอร์ด มีขนาดใหญ่
- 3) เหมาะกับการนำมาทำโปรโตไทป์มากกว่าที่จะนำไปใช้งานจริง

รูปแบบการเขียนโปรแกรมบน Arduino และการเชื่อมต่อ ดังรูป 2.9

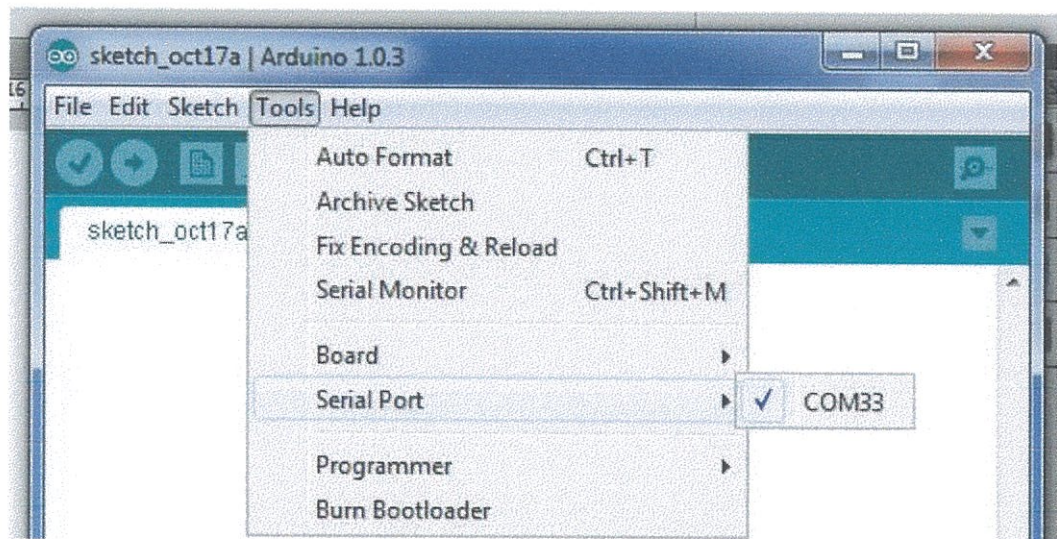


รูป 2.9 ตัวอย่างการเชื่อมต่อ Arduino กับ Computer และโปรแกรม ArduinoIDE

- 1) เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software
- 2) หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้ และหมายเลข Com port ดังรูป 2.10 และ 2.11

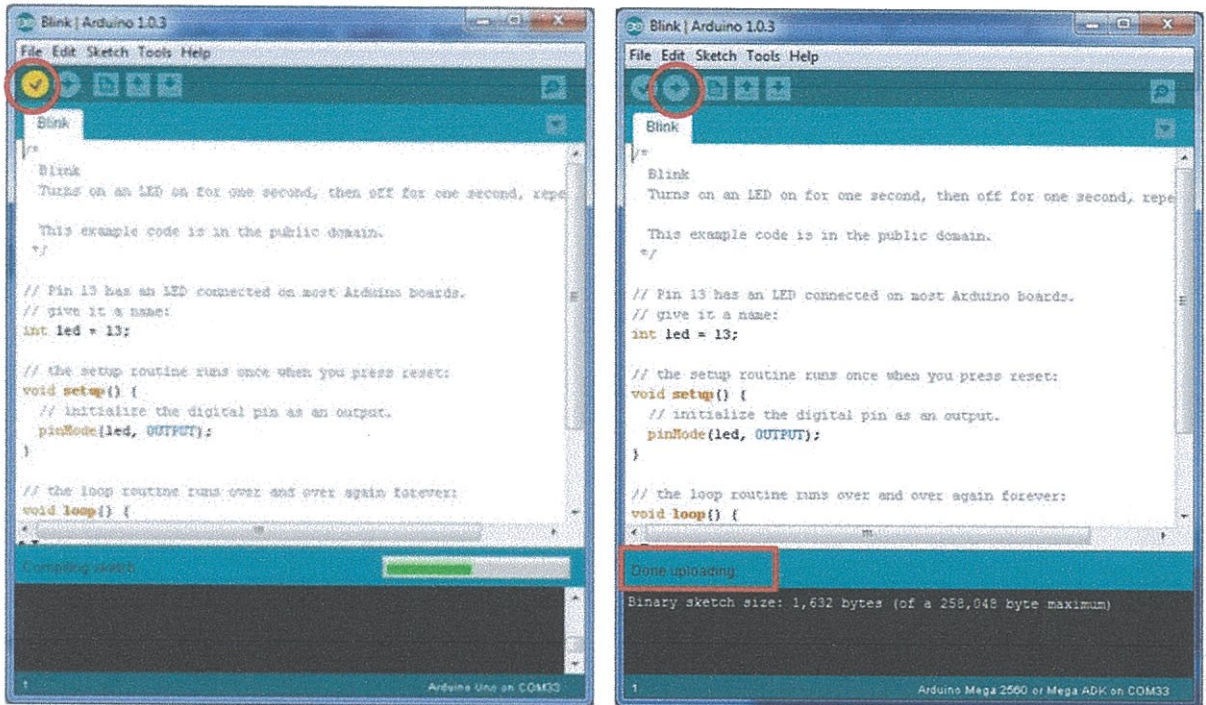


รูป 2.10 เลือกบอร์ด Arduino ที่ต้องการ upload



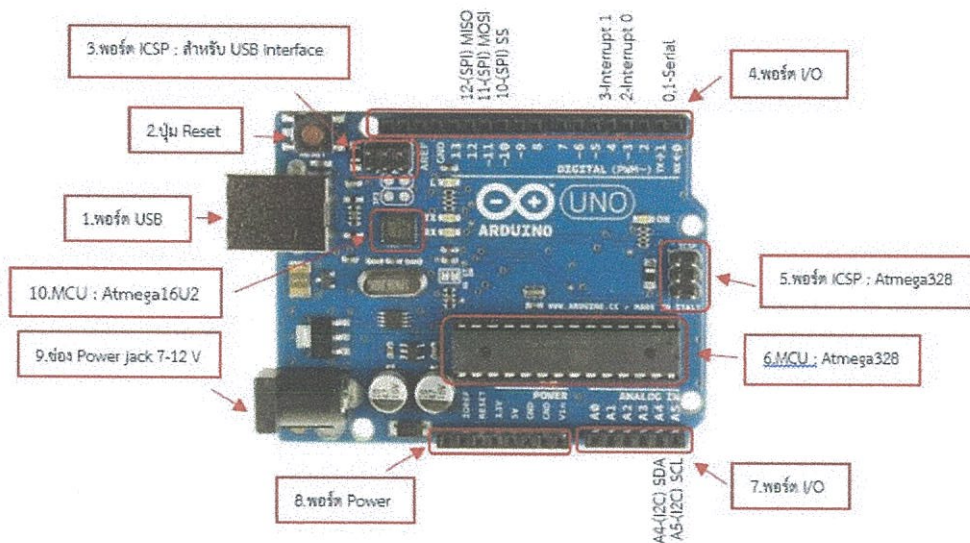
รูป 2.11 เลือกหมายเลข Comport ของบอร์ด

- 3) กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง และ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ดโปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง "Done uploading" และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที ดังรูป 2.12



รูป 2.12 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง, การ Compile และการ Upload โปรแกรม

Layout และ Pin out ของ Arduino UNO R3 Board ดังรูป 2.13



รูป 2.13 Layout และ Pin out ของ Arduino UNO R3 Board

- 1) USBPort: ใช้สำหรับต่อกับ Computer เพื่อ Upload โปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2) Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

- 3) ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
- 4) I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่น ๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- 5) ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
- 6) MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
- 7) I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- 8) Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วย ไฟเลี้ยง +3.3 V, +5V, GND, V_{cc}
- 9) Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
- 10) MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.2.2 GPS

ระบบกำหนดตำแหน่งบนโลก หรือ Global Positioning System (GPS) ระบบนี้ได้พัฒนาขึ้นโดยกระทรวงกลาโหม ประเทศสหรัฐอเมริกา ซึ่งจัดทำโครงการ Global Positioning System มาตั้งแต่ปี พ.ศ. 2521 โดยอาศัยดาวเทียม ระบบคลื่นวิทยุนำร่อง และรหัสที่ส่งมาจากดาวเทียม NAVSTAR จำนวน 24 ดวง แบ่งเป็นชุด ชุดละ 4 ดวง โดยทำการโคจรอยู่รอบโลกวันละ 2 รอบ และมีตำแหน่งอยู่เหนือพื้นโลกที่ความสูงประมาณ 20,200 กิโลเมตร

2.2.2.1 องค์ประกอบหลักของ GPS

ระบบกำหนดตำแหน่งบนโลก ประกอบด้วย 3 ส่วนหลัก คือ ส่วนอวกาศ (Space segment) ส่วนสถานีควบคุม (Control segment) และส่วนผู้ใช้ (User segment)

- 1) ส่วนอวกาศ (Space segment) เป็นส่วนที่อยู่บนอวกาศ ประกอบด้วยดาวเทียม 24 ดวง โดยมี 21 ดวง แบ่งเป็น 6 วงโคจร วงโคจรละ 4 ดวง อยู่สูงจากพื้นดินประมาณ 20,200 กิโลเมตร ทำหน้าที่ส่งสัญญาณคลื่นวิทยุจากอวกาศ
- 2) ส่วนสถานีควบคุม (Control segment) ประกอบไปด้วยสถานีภาคพื้นดินที่ควบคุมระบบที่กระจายอยู่ตามส่วนต่าง ๆ ของโลกโดยแบ่งออกเป็นสถานีควบคุมหลักตั้งอยู่ที่ฐานทัพอากาศในเมืองโคโลราโดสปริงส์ (Colorado Springs) มลรัฐโคโลราโดของสหรัฐอเมริกาสถานีติดตามดาวเทียม 5 แห่ง ทำการรังวัดติดตามดาวเทียมตลอดเวลา สถานีรับส่งสัญญาณ 3 แห่ง

- 3) ส่วนผู้ใช้ (User segment) ประกอบด้วยเครื่องรับสัญญาณ หรือเครื่องรับจีพีเอส GPS ซึ่งมีหลายขนาดสามารถพกพาติดตัว หรือติดไว้ในรถ เรือ เครื่องบิน ได้

2.2.2.2 หลักการของ GPS

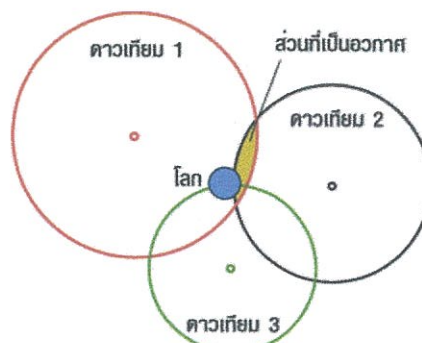
การคำนวณระยะทางระหว่างดาวเทียมกับเครื่อง GPS ซึ่งต้องใช้ระยะทางจากดาวเทียมอย่างต่ำ 3 ดวง เพื่อให้ได้ตำแหน่งที่แน่นอนซึ่งเมื่อเครื่อง GPS สามารถรับสัญญาณจากดาวเทียมได้ 3 ดวงขึ้นไปแล้วจะมีคำนวณระยะทางระหว่างดาวเทียมถึงเครื่อง GPS โดยจากสูตรคำนวณทางฟิสิกส์คือ ความเร็ว X เวลา = ระยะทาง

โดยดาวเทียมทั้ง 3 ดวงจะส่งสัญญาณที่เหมือนกันมายังเครื่อง GPS โดยความเร็วแสง (186,000 ไมล์ต่อวินาที) แต่ระยะเวลาในการรับสัญญาณได้จากดาวเทียมแต่ละดวงนั้นไม่เท่ากัน เนื่องจากระยะทางไม่เท่ากัน เช่น

ดาวเทียม 1 : ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่อง GPS คือ 0.10 วินาที ระยะทางระหว่างดาวเทียมกับ GPS คือ 18,600 ไมล์ (186,000 ไมล์ต่อวินาที X 0.10 วินาที = 18,600 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในวงกลมที่มีรัศมี 18,600 ไมล์ ซึ่งจะเห็นว่าดาวเทียมเพียงดวงเดียวยังไม่สามารถบอกตำแหน่งที่แน่นอนได้

ดาวเทียม 2 : ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่อง GPS คือ 0.08 วินาที ระยะทางระหว่างดาวเทียมกับ GPS คือ 13,200 ไมล์ (186,000 ไมล์ต่อวินาที X 0.08 วินาที = 13,200 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในจุด Intersect ระหว่างวงกลมจากดาวเทียมดวงแรกกับดาวเทียมดวงที่ 2

ดาวเทียม 3 : ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่อง GPS คือ 0.06 วินาที ระยะทางระหว่างดาวเทียมกับ GPS คือ 11,160 ไมล์ (186,000 ไมล์ต่อวินาที X 0.06 วินาที = 11,160 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในจุด Intersect ระหว่างวงกลมจากดาวเทียมทั้ง 3 ดวง ดังรูป 2.14



หลักการคำนวณของ GPS

รูป 2.14 หลักการทำงานของ GPS

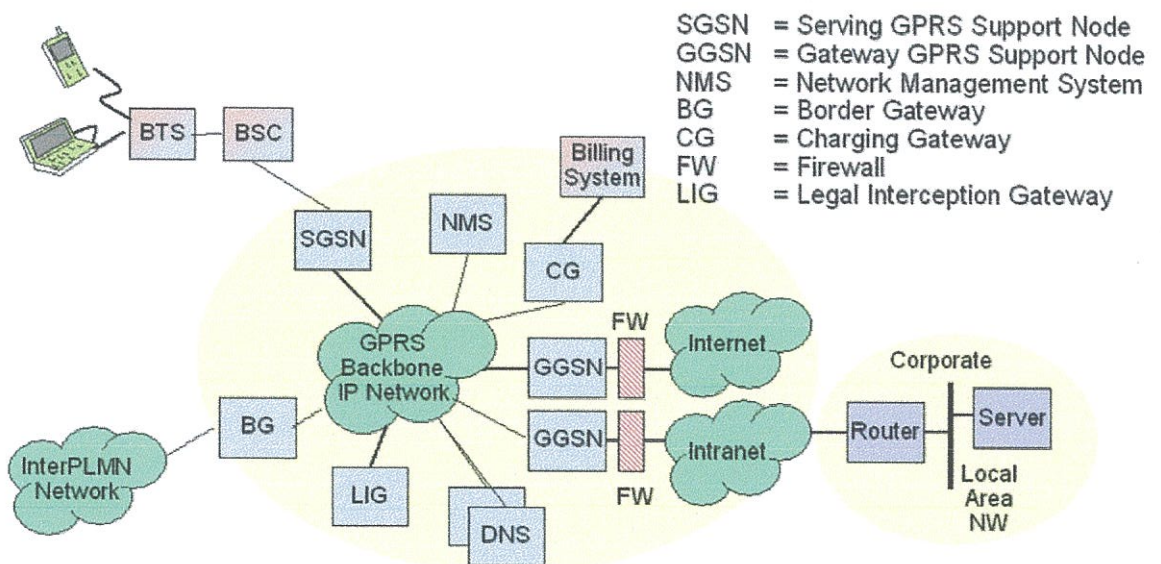
จะเห็นได้ว่าจะเหลือตำแหน่งอยู่ 2 จุดที่บริเวณวงกลมทั้ง 3 ตัดกันคือตำแหน่งที่อยู่ในอวกาศ ซึ่งแน่นอนว่าไม่สามารถไปอยู่ในอวกาศได้ตำแหน่งนี้จะถูกตัดทิ้งอัตโนมัติโดย

2.2.3 GPRS

GPRS ย่อมาจาก General Packet Radio Service ซึ่งเป็นการส่งข้อมูลแบบเป็น Packet สามารถเข้าสู่ข้อมูลที่หลากหลายภายในเวลาที่รวดเร็วขึ้นกว่าระบบเดิม ๆ สูงสุดถึง 10 เท่า นอกจากนี้ลักษณะหนึ่งที่สำคัญของ GPRS คือ "Always On" นั่นหมายความว่าสามารถเข้าสู่โลกอินเทอร์เน็ตได้ตลอดเวลา โดยไม่มีความจำเป็นต้องทำการ Log On หรือใช้ User Name และ Password เหมือนการเข้าสู่อินเทอร์เน็ตทั่วไปแต่อย่างใด ทั้งนี้ยังสามารถที่จะเปลี่ยน โหมดมาทำการโทรออก หรือรับสายเข้าโดยยังสามารถเชื่อมต่อกับโลกอินเทอร์เน็ตในระบบ GPRS อยู่ตลอดเวลา ข้อมูลที่รับส่งผ่านเครือข่าย GPRS จะถูกตัดแบ่งเป็น packet ย่อย ๆ ก่อนในแต่ละ packet จะมีข้อมูลระบุถึงที่มาที่สัมพันธ์กันเพื่อใช้ในการประกอบกลับขึ้นมาเป็นข้อมูลเดิมอีกครั้งซึ่งการนำข้อมูลมาต่อเข้าด้วยกันจะใช้วิธีดูจากความสัมพันธ์ของแต่ละส่วน ซึ่งอาจจะมีวิธีที่แตกต่างกันไป

GPRS จะต้องการ โหนดใหม่เพิ่มเติมอีก 2 โหนดสำหรับจัดการกับทราฟฟิกแบบ packet คือ

- 1) GGSN (Gateway GPRS Service Node) ทำหน้าที่เป็น gateway เชื่อมต่อระหว่างเครือข่าย GPRS กับเครือข่ายข้อมูลทั่วไปเช่น IP และ X.25 ซึ่งรวมถึงการเชื่อมต่อกับเครือข่าย GPRS อื่น ๆ เพื่อการ roaming ด้วย
- 2) SGSN (Serving GPRS Service Node) ทำหน้าที่เกี่ยวกับการเชื่อมต่อเส้นทาง (routing) ระหว่าง SGSN ในแต่พื้นที่สำหรับผู้ใช้ทุกคนในพื้นที่ให้บริการ ดังรูป 2.15



รูป 2.15 ภาพรวมการทำงาน GPRS

2.2.4 SIM908 Module

SIM908 Module เป็น Quad-Band GSM/GPRS Module โดยเป็นการผสมผสานและนำเทคโนโลยี GPS เพื่อใช้สำหรับการค้นหาเส้นทางผ่านดาวเทียม โดยการนำ GPRS/GPS มาใช้ใน SMT package จะทำให้เราสามารถประหยัดเวลา และค่าใช้จ่ายในการพัฒนา application เพื่อให้รองรับการใช้งานนี้ได้ โดยการใช้งาน function ของ GPS จะทำให้เราสามารถค้นหา และแทรคเส้นทางที่ต้องการได้อย่างราบรื่น ทุกที่ทุกเวลาโดยไม่มีกรขาดหายของสัญญาณ ตัวอย่าง SIM908 Module ดังรูป 2.16



รูป 2.16 SIM908 Module

2.2.4.1 คุณสมบัติทั่วไป

Quad-Band 850/900/1800/1900MHz

GPRS multi-slot class 10

GPRS mobile station class B

Compliant to GSM phase 2/2+

- Class 4 (2 W @ 850/900 MHz)

- Class 1 (1 W @ 1800/1900MHz)

Control via AT commands

SIM application toolkit

Supply voltage range:

- GPRS: 3.2 ~ 4.8 V

- GPS: 3.0 ~ 4.5V

Low power consumption

Dimensions: 30*30*3.2mm

Weight: 5.2 g

Operation temperature: -40 °C to +85 °C

2.2.4.2 ข้อมูลจำเพาะสำหรับการแลกเปลี่ยนข้อมูล

GPRS class 8/10: max. 85.6 kbps (downlink)

PBCCH support

Coding schemes CS 1, 2, 3, 4

CSD up to 14.4 kbps

USSD

Non transparent mode

PPP-stack

Integrated TCP/IP stack

2.2.4.3 ข้อมูลจำเพาะสำหรับ GPS

Receiver type

42-channel

GPS L1 C/A code,

High-performance STE engine Sensitivity

Tracking: -160 dBm

Cold starts: -143 dBm

Time-To-First-Fix

Horizontal position: <2.5m CEP

Acquisition 77mA

Tracking 76mA

2.2.4.4 การเชื่อมต่อ

80-pad with SMT type

Interface to external SIM 3V/ 1.8V

Dual analog audio interfaces

RTC backup

SPI interface

Serial interface and a debug interface for GSM/GPRS

Debug interface for GPS NMEA information output

Two separate antenna connectors for GSM/GPRS&GPS

2.2.4.5 Compatibility

AT cellular command interface

2.2.4.6 การเชื่อมต่อเข้ากับ SIM908 Module

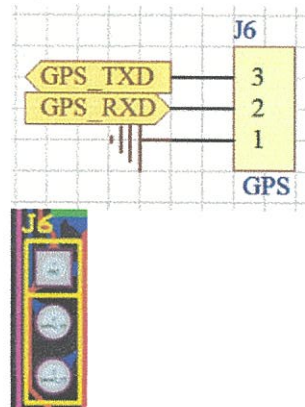
โดยเลือกใช้ SIM908 Module รุ่น MD-SIM908 V.2 ของบริษัทไมโครไดรฟ์ ซึ่งมีวิธีการเชื่อมต่อดังต่อไปนี้

- 1) การเชื่อมต่อกับ GPS กลุ่ม Pin J6 ดังรูป 2.17

Pin1 - GND

Pin2 - GPS รับสัญญาณ Rx RS232

Pin3 - GPS ส่งสัญญาณ Tx RS232



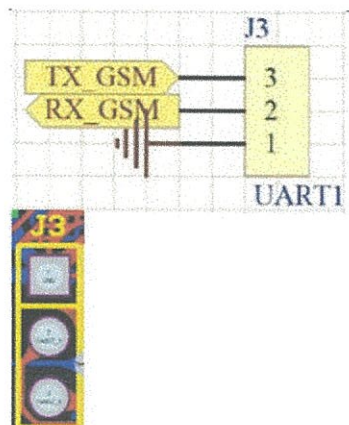
รูป 2.17 GPS กลุ่ม Pin J6 ของ SIM908 Module

- 2) การเชื่อมต่อกับ GSM กลุ่ม Pin J3 ดังรูป 2.18

Pin1 - GND

Pin2 - GSM รับสัญญาณ Rx RS232

Pin3 - GSM ส่งสัญญาณ Tx RS232



รูป 2.18 GSM กลุ่ม Pin J3 ของ SIM908 Module

3) การเชื่อมต่อกับขั้วต่อแบตเตอรี่ กลุ่ม Pin J1 ดังรูป 2.19

ถ้าไม่ต้องการใช้ไฟจากสาย USB สามารถใช้งานจากแบตเตอรี่ภายนอกได้

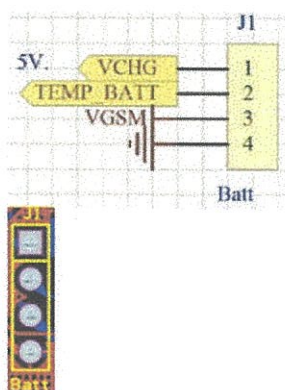
Pin1 - ขั้วชาร์จแบตเตอรี่ สามารถนำไฟจาก J5_2 มาชาร์จได้ด้วยการจัมพ์

สายระหว่าง Pin

Pin2 - อุณหภูมิจากแบตเตอรี่

Pin3 - สายแบตเตอรี่เอาท์พุท

Pin4 - GND



รูป 2.19 ขั้วต่อแบตเตอรี่ กลุ่ม Pin J1 ของ SIM908 Module

4) การเชื่อมต่อกับ UART กลุ่ม Pin J8

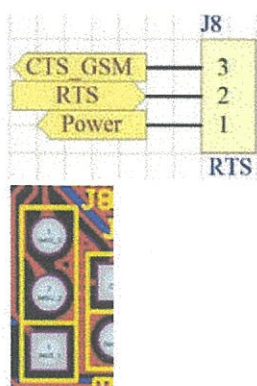
ถ้าไม่ต้องการใช้ไฟจากสาย USB สามารถใช้งานจากแบตเตอรี่ภายนอกได้

และถ้าต้องการให้สามารถเปิด Power จากโปรแกรมได้จะต้องทำการ short ระหว่าง Pin1 และ Pin2

Pin1 - Power

Pin2 - RST

Pin3 - CTS_GSM



รูป 2.20 UART กลุ่ม Pin J8 ของ SIM908 Module

2.2.5 AT Command

AT Command คือ ชุดคำสั่งมาตรฐานที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่าง ๆ เช่น โมเด็ม หรือ อุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่า หรือสั่งงานอุปกรณ์เหล่านั้นให้ทำงานตามที่ต้องการ และสำหรับการติดต่อกับโทรศัพท์มือถือจะใช้ชุดคำสั่งที่เรียกว่า GSM AT COMMAND โดยจุดประสงค์ที่ใช้ AT Command เพื่อให้ Arduino สามารถติดต่อกับ SIM908 Module

ตัวอย่างคำสั่งที่ใช้งาน

- 1) AT: ตรวจสอบความพร้อมของอุปกรณ์ ถ้าสามารถติดต่อกันได้ อุปกรณ์จะตอบกลับมาว่า OK
- 2) AT+IPR=9600: ตั้งค่า Baud Rate 9600
- 3) AT+CGPSPWR=1: เปิดให้ GPS ทำงาน
- 4) AT+CGPSIPR=9600: ตั้งค่า GPS Baud Rate 9600
- 5) AT+CGPSSTATUS?: รับค่าสถานะของ GPS
- 6) AT+CGPSINF=0: รับข้อมูล GPS
- 7) ATB: Communications standard option
- 8) AT+CBST: Select bearer service type
- 9) AT+CEER: Extended error report
- 10) AT+CRLP: Radio link protocol
- 11) AT+DS: Data compression

2.2.6 wMQTT Library

MQTT มีคุณสมบัติสำหรับการส่งข้อมูลที่มีขนาดเล็กที่ดี ระหว่าง อุปกรณ์ฮาร์ดแวร์ กับ เซิร์ฟเวอร์ หรืออุปกรณ์ด้วยกันเอง ซึ่งผู้พัฒนา Library นี้ต้องให้ ข้อมูล มีขนาดเล็กที่สุดเท่าที่สามารถเป็นไปได้ และสามารถใช้ได้กับทุก platform นอกจากนี้ยังต้องการให้มีความต่อเนื่องกันระหว่าง อุปกรณ์กับเซิร์ฟเวอร์ ฟังก์ชันของ Library นี้เป็นการเตรียม ข้อมูล MQTT ซึ่งจะถูกนำไปใช้ในการส่งผ่านไปยังเซิร์ฟเวอร์ โดยเชื่อมต่อด้วย TCP/IP ซึ่งทุก ๆ ฟังก์ชันในการเตรียม ข้อมูล MQTT มีพื้นฐานมาจาก MQTT V3

เหตุผลที่เลือกใช้ Library นี้คือ เป็น Library ที่สามารถทำงานร่วมกับ GPRS Module บน Arduino ได้

2.3 ทฤษฎีพัฒนาระบบ แสดงผลข้อมูลบนแผนที่

2.3.1 Apache

คือ ซอฟต์แวร์ทำหน้าที่เป็นเครื่องบริการเว็บ (Web server) โดยให้บริการ โพรโทคอล HTTP ที่ port 80 ลักษณะเด่น คือเป็น ซอฟต์แวร์ที่เป็น Open source ติดตั้งมาพร้อมกับ ระบบปฏิบัติการ Linux และมีใช้กันอย่างแพร่หลายมากที่สุดในโลก
หน้าที่หลักของเว็บเซิร์ฟเวอร์

- 1) คอยจัดการ Request ก็คือ การร้องขอข้อมูล
- 2) คอยจัดการ Response ก็คือ การส่งข้อมูลกลับไป
- 3) คอยจัดการ process และจัดลำดับของ request และ response
- 4) คอยเก็บ logs ที่มีการ access เข้ามารวมกระทั่ง error ต่าง ๆ ที่ เว็บเซิร์ฟเวอร์พบ

สามารถนำ module มาใช้ร่วมกับ เว็บเซิร์ฟเวอร์ได้ ยกตัวอย่างเช่น Apache นั้น ไม่สามารถ run ไฟล์ .php ได้ ต้อง เรียกใช้ module php

2.3.2 PHP

เป็นภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-side script) ใช้สำหรับทำ Website และแสดงผลออกมาในรูปแบบ html โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และภาษาเพิร์ล ซึ่ง PHP เป็นภาษาแบบ open source ทำให้นักพัฒนาเว็บไซต์สามารถเขียนเว็บเพจที่มีการโต้ตอบ ได้อย่างรวดเร็ว

2.3.3 Laravel Framework

PHP Framework ที่เต็มเปี่ยมไปด้วยพลังที่ทำให้คุณสามารถเขียน โค้ดที่ดูสะอาดตา และสามารถอ่าน และทำความเข้าใจได้โดยง่าย ยังสามารถดาวน์โหลดมาใช้งานได้ฟรี ออกแบบมาเพื่อพัฒนาเว็บแอปพลิเคชันในรูปแบบ MVC พัฒนาโดยมีผู้นำทีมคือนาย Taylor Otwell ภายใต้อิทธิพลของ MIT และ source code ได้ถูกเก็บไว้บน host ของ Github

จุดเด่นของ Laravel

- 1) Bundle (สิ่งที่บรรจุมาด้วยกัน) Laravel ได้มาพร้อมกับ แพคเกจของระบบ ที่ทำให้เราสามารถนำมาใช้กับ เว็บแอปพลิเคชันของเราได้เลย จึงทำให้เราประหยัดเวลาในการเขียนโค้ด และลดจำนวนการเขียนโปรแกรมลงอย่างมาก
- 2) Class Autoloading (โหลด Class อัตโนมัติ) ระบบจะทำการโหลด Class ของ PHP มาใช้งานอัตโนมัติ โดยไม่ต้องกำหนดค่าการโหลดใช้งานเอง ในการโหลดระบบจะป้องกันการโหลดในส่วนประกอบ (component) ที่ไม่ใช้งาน และจะเลือกโหลดเฉพาะส่วนประกอบที่นำมาใช้งานเท่านั้น

- 3) View Composer (ส่วนของ View) ส่วนนี้จะเป็นส่วนของโค้ด (HTML) ที่นำมาเรียงติดต่อกัน และจะทำงาน (run) หลังจากประกอบกันเสร็จเรียบร้อยแล้ว เช่น เราแบ่งส่วน header, container, sidebar, footer เป็นต้น
- 4) Unit testing (หน่วยทดสอบ) Laravel ยินยอมให้ผู้ใช้งานสามารถสร้าง unit test ขึ้นมาเพื่อทดสอบงานของตัวเองได้โดยผ่าน Artisan utility
- 5) The Eloquent ORM (ชุดคำสั่งในการ Query) เป็นเครื่องมือที่ช่วยในการ Query ข้อมูลต่าง ๆ ในฐานข้อมูล
- 6) Reverse Routing (เราที่ตั้งค์) ในส่วนนี้จะทำให้คุณสามารถกำหนดชื่อของ URL เพื่อที่จะชี้ไปยังส่วนต่าง ๆ ตามต้องการ
- 7) Restful Controller (กรองชนิดตามการส่งคำขอ) ช่วยให้เราสามารถกรองชนิดการส่งคำร้องขอจากฟอร์มทั้งแบบ Post และ Get
- 8) The IoC container (Inversion of Control) เป็นส่วนในการจัดเก็บ Library ภายนอกที่เราจะนำเข้ามาใช้

ข้อเสียของ Laravel

- 1) เนื่องจากมีเครื่องมือให้ใช้มากมาย ทำให้ผู้เริ่มต้น ต้องใช้เวลามากนิดหน่อย กว่าจะทำความคุ้นเคยได้
- 2) บางครั้ง เครื่องมือที่มีมาให้ ก็ไม่ได้จำเป็นทั้งหมด ทำให้เรามีส่วนของโค้ดที่ไม่ได้ใช้งานอยู่
- 3) Laravel ไม่ใช่ PHP Framework ที่เร็วที่สุด (แต่ก็ไม่ได้ช้าที่สุด) ผู้ใช้งาน ต้องมีความรู้ในการปรับแต่งระบบ ให้ทำงานร่วมกับ Laravel ได้ดี
- 4) เนื่องจาก Laravel ไม่ได้โดดเด่นเรื่องความเร็ว หรือประสิทธิภาพในการทำงาน ทางออกคือ มีเซิร์ฟเวอร์ที่ปรับแต่งขึ้นมาเพื่อให้บริการสำหรับเว็บแอปพลิเคชัน ที่พัฒนาด้วย Laravel โดยเฉพาะ ที่นิยมที่สุดในขณะนี้คือ Forge ร่วมกับ DigitalOcean หรือ Linode

2.3.4 MySQL

คือ ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL ทำหน้าที่เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือ หรือ โปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้ เช่น ทำงานร่วมกับเครื่องบริการเว็บ (Web Server) เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่งเครื่องบริการ (Server-Side Script) และ MySQL เป็นซอฟต์แวร์แบบ Open source ที่มีทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ ซึ่งพัฒนาโดยบริษัทมายเอสคิวแอลเอบี (MySQL AB)

ชนิดของข้อมูลที่สนับสนุน

ชนิดข้อมูลที่ MySQL สนับสนุนแบ่งเป็นสามประเภทหลักใหญ่ ๆ

- 1) ชนิดข้อมูลที่เป็นตัวเลข
 - a) BIT
 - b) TINYINT
 - c) SMALLINT
 - d) MEDIUMINT
 - e) INT
 - f) BIGINT
 - g) Float
- 2) ชนิดข้อมูลที่เกี่ยวข้องกับวันที่ และเวลา
 - a) DATETIME
 - b) DATE
 - c) TIMESTAMP
 - d) TIME
 - e) YEAR
- 3) ชนิดข้อมูลที่เกี่ยวข้องกับตัวอักษร
 - a) CHAR
 - b) VARCHAR
 - c) BINARY
 - d) VARBINARY
 - e) BLOB
 - f) TEXT
 - g) ENUM
 - h) SET

2.3.5 Google Maps API

Google Maps API เป็นคอลเล็กชันของ API ที่ช่วยให้คุณสามารถวางข้อมูลของคุณเอง ซ้อนลงบนแผนที่ของ Google ที่กำหนดเอง คุณสามารถสร้างเว็บ และแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ ที่น่าสนใจด้วยแพลตฟอร์มการทำแผนที่ ที่มีประสิทธิภาพของ Google รวมทั้งดึงภาพจาก ดาวเทียม ภาพ Street View โพรไฟล์ระดับความสูง เส้นทางการขับขี่ ข้อมูลประชากรในรูปแบบแผนที่ การวิเคราะห์ และฐานข้อมูลเกี่ยวกับสถานที่ต่าง ๆ มากมาย ด้วยการครอบคลุมพื้นที่ในวงกว้าง และแม่นยำ

มี features ต่าง ๆ มากมายให้เรียกใช้ เช่น

- 1) การปรับแต่งแผนที่ (Styled Map)
- 2) ชุดควบคุมแผนที่ (Map Control)
- 3) ชุดเครื่องมือวาดภาพบนแผนที่ (Drawing)
- 4) การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
- 5) การคำนวณความสูงของจุดพิกัด (Elevation Service)
- 6) การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (Geocoding Service)
- 7) การดึงข้อมูล POI (Point of Interest) คือ ข้อมูลสถานที่ต่าง ๆ ที่ Google รวบรวมไว้ให้ เช่น โรงแรม ห้างสรรพสินค้า โรงเรียน สถานที่ราชการต่าง ๆ และอื่น ๆ อีกมากมาย (Places API) มาใช้งานใน application เรา
- 8) Street View

2.3.6 Restful Web service

Representational State Transfer (REST) นั้นเป็นสถาปัตยกรรมการสื่อสารข้อมูลรูปแบบหนึ่งที่ใช้แพร่กระจายสื่อ เป็นแนวทางใหม่ในการสร้าง Web Service แบบเรียบง่าย โดยเรียกใช้ผ่านทาง HTTP Method GET / POST / PUT / DELETE และส่งข้อมูลที่มีปริมาณข้อมูลที่รับส่ง น้อยกว่าการใช้ Protocol SOAP ซึ่งข้อดีข้อนี้ของ REST ทำให้ Developer หลาย ๆ คนหันมาสนใจการเขียนโปรแกรมแบบใช้ RESTful Web Service กันมากขึ้น เพราะมีผลกับเรื่องประสิทธิภาพ ของการใช้งานโปรแกรมเป็นอย่างมาก แต่เนื่องจากเรื่อง REST นี้เพิ่งเกิดขึ้นมาเมื่อปี 2000 ทำให้ยังไม่มีมาตรฐานที่กำหนดให้บังคับใช้งานเหมือน Protocol SOAP เดิม ซึ่งจะสังเกตได้ว่าจะไม่มี REST Specification อยู่บน W3C และไม่มี REST Developer Toolkit ถ้าหากอยากใช้ RESTful ก็ต้องกำหนดเงื่อนไขจากภาษาที่เขียนเอง โดยภาษาในยุคปัจจุบันนี้มีการรองรับ RESTful Web Service หรืออีกชื่อหนึ่งคือ RESTful Web API กันเป็นส่วนใหญ่แล้ว

พื้นฐานของ REST

- 1) การใช้งาน resource ต่าง ๆ
- 2) ผ่าน HTTP protocol
- 3) ผ่าน URI ที่สื่อสารได้ชัดเจน
- 4) ผ่าน HTTP method ที่เหมาะสม

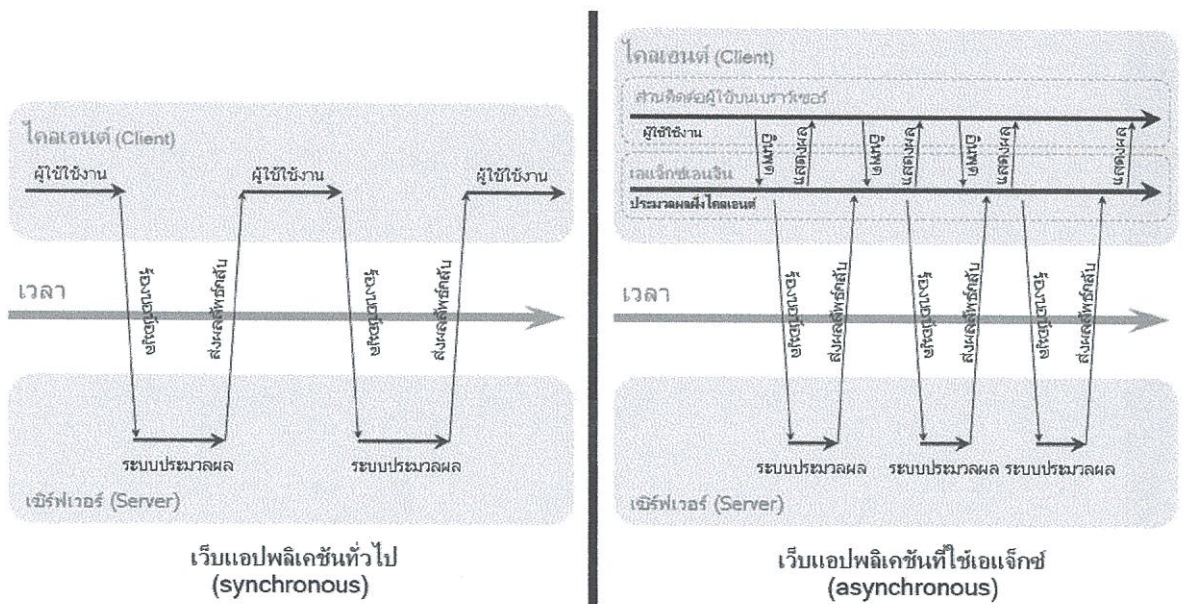
Method การทำงาน

- 1) HTTP GET สำหรับการดึงข้อมูล
- 2) HTTP POST สำหรับการสร้างข้อมูล
- 3) HTTP PUT สำหรับการแก้ไขข้อมูล
- 4) HTTP DELETE สำหรับลบข้อมูล

2.3.7 Ajax (Asynchronous JavaScript and XML)

เป็นเทคนิคในการพัฒนาเว็บแอปพลิเคชัน เพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็ว และการใช้งานโดยรวม ตัวอย่างของเว็บแอปพลิเคชันที่ใช้เทคนิคของ AJAX ที่เรามีโอกาสได้ใช้งานได้แก่ Gmail, Google Suggest, Google Maps, Flickr, MS Outlook Web Access เป็นต้น

วิธีการทำงานของเว็บแอปพลิเคชันแบบเดิมนั้น โดยปกติแล้วเมื่อผู้ใช้งานทำการร้องขอข้อมูลจาก เซิร์ฟเวอร์ตัวเว็บเบราว์เซอร์ทำการส่งข้อมูลการร้องขอโดยใช้โปรโตคอล HTTP เพื่อติดต่อกับเว็บเซิร์ฟเวอร์ และที่เว็บเซิร์ฟเวอร์ทำการประมวลผลจากการร้องขอที่ได้รับ และส่งผลลัพธ์เป็น HTML กลับไปให้ผู้ใช้งาน วิธีการข้างต้นเป็นวิธีการแบบการร้องขอ และการตอบรับ (Request and Response) ซึ่งผู้ใช้งานต้องรอ ระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ ซึ่งเป็นหลักการทำงานแบบ Synchronous แต่การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิคเอแจ็กซ์เป็นการทำงานแบบ Asynchronous หรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดย เซิร์ฟเวอร์ทำการส่งผลลัพธ์เป็นเว็บเพจให้ผู้ใช้งานทันทีโดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเว็บเพจที่ผู้ใช้งานได้รับจะทำการดึงข้อมูลในส่วนต่าง ๆ ที่หลัง หรือดึงข้อมูลก็ต่อเมื่อผู้ใช้งานต้องการเท่านั้น



รูป 2.21 การทำงานของ AJAX

2.4 เทคโนโลยี Internet of Things

Internet of Things (IoT) คือ เทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์ และเครื่องมือต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ ตู้เย็น โทรทัศน์ และอื่น ๆ เข้าไว้ด้วยกัน โดยเครื่องมือต่าง ๆ จะสามารถเชื่อมโยง และสื่อสารกันได้โดยผ่านระบบอินเทอร์เน็ต ในอนาคตผู้บริโภคทั่วไปเริ่มคุ้นเคยกับเทคโนโลยีที่ทำให้พวกเขาสามารถควบคุมสิ่งของต่าง ๆ ทั้งจากในบ้าน และสำนักงาน หรือจากที่ไหนก็ได้ เช่น การควบคุมอุณหภูมิภายในบ้าน การเปิดปิดไฟ ไปจนถึงการสั่งให้เครื่องทำกาแฟ เริ่มต้มกาแฟ แต่อย่างไรก็ตาม ยังมีเทคโนโลยีอื่น ๆ ที่จำเป็นต้องถูกพัฒนา ก่อนที่ IoT จะเป็นความจริงขึ้นมา เช่น ระบบตรวจจับต่าง ๆ (Sensors) รูปแบบการเชื่อมต่อระหว่างอุปกรณ์ และระบบที่ฝังตัวอยู่ในคอมพิวเตอร์

ปัจจุบันมีอุปกรณ์มากมายถูกผลิตขึ้นมาเพื่อให้นักพัฒนาสามารถนำไปใช้ต่อยอดสร้างชิ้นงานเชื่อมต่อเข้ากับ Internet of Things โดยมีคุณสมบัติที่สามารถเชื่อมต่อระบบเครือข่ายคอมพิวเตอร์ไร้สายได้ แบ่งออกเป็นกลุ่ม ๆ ดังนี้

- 1) กลุ่มที่ 1 บอร์ดไมโครคอนโทรลเลอร์สำหรับ Internet of Things บอร์ดไมโครคอนโทรลเลอร์กลุ่มนี้ ผู้ผลิตออกแบบ และสร้างขึ้นมาให้เราสามารถพัฒนาโปรแกรมลงในไมโครคอนโทรลเลอร์บนบอร์ด ควบคุมอินพุต และเอาต์พุตเชื่อมต่ออุปกรณ์เพื่อตรวจจับ หรือแสดงผลได้ตามต้องการ พร้อมใส่อุปกรณ์ที่เชื่อมต่อเครือข่ายแบบไร้สาย Wi-Fi มาบนบอร์ดไม่จำเป็นต้องต่อเพิ่มเติมเอง
- 2) กลุ่มที่ 2 โมดูลสื่อสารไร้สาย Internet of Things โมดูลบางตัวที่มีคุณสมบัติสามารถเป็นอุปกรณ์ Internet of Things ได้ ยกตัวอย่างเช่น โมดูล ESP8266 สามารถทำงาน Standalone เนื่องจากภายในมีไมโครคอนโทรลเลอร์อยู่ (แต่ไม่สามารถโปรแกรมแบบทั่วไปได้) ต้องพัฒนาเฟิร์มแวร์ให้ทำงานตามความต้องการ กลุ่มนักพัฒนาที่สนใจได้แบ่งปันข้อมูลดังกล่าวไว้ ผู้ใช้อาจดาวน์โหลดเฟิร์มแวร์ที่มีการทำงานตามที่เรากำลังมาติดตั้งใช้งาน
- 3) กลุ่มที่ 3 บอร์ดไมโครคอนโทรลเลอร์ + โมดูลสื่อสารไร้สาย ทำงานร่วมกันเป็น Internet of Things กลุ่มนี้เป็นการประยุกต์ใช้บอร์ดไมโครคอนโทรลเลอร์ต่าง ๆ ร่วมกับโมดูลสื่อสารไร้สาย เขียนแอปพลิเคชันให้ทำงานในลักษณะคล้ายกับกลุ่มที่ 1 เพียงแต่เราต้องจับคู่เชื่อมต่อบอร์ดกับโมดูลต่าง ๆ เอง ยกตัวอย่างเช่น การใช้ Arduino ร่วมกับ Wi-Fi Shield หรือ UART Wi-Fi การใช้ Raspberry Pi กับ USB Wi-Fi Dongle เพียงเท่านี้ก็สามารถเชื่อมต่อกับระบบเครือข่ายได้

ในที่นี้ใช้ Arduino board ร่วมกับ SIM908 Module ซึ่งเป็น GPRS/GPS Module โดยนำอุปกรณ์ดังกล่าวไปติดตั้งบนรถตู้โดยสารสาธารณะ และเขียนโปรแกรมควบคุมการทำงานเพื่อทำการรับ-ส่งตำแหน่งของรถแต่คันไปทำการประมวลผลต่อไป

บทที่ 3

การวิเคราะห์ออกแบบและพัฒนาระบบ

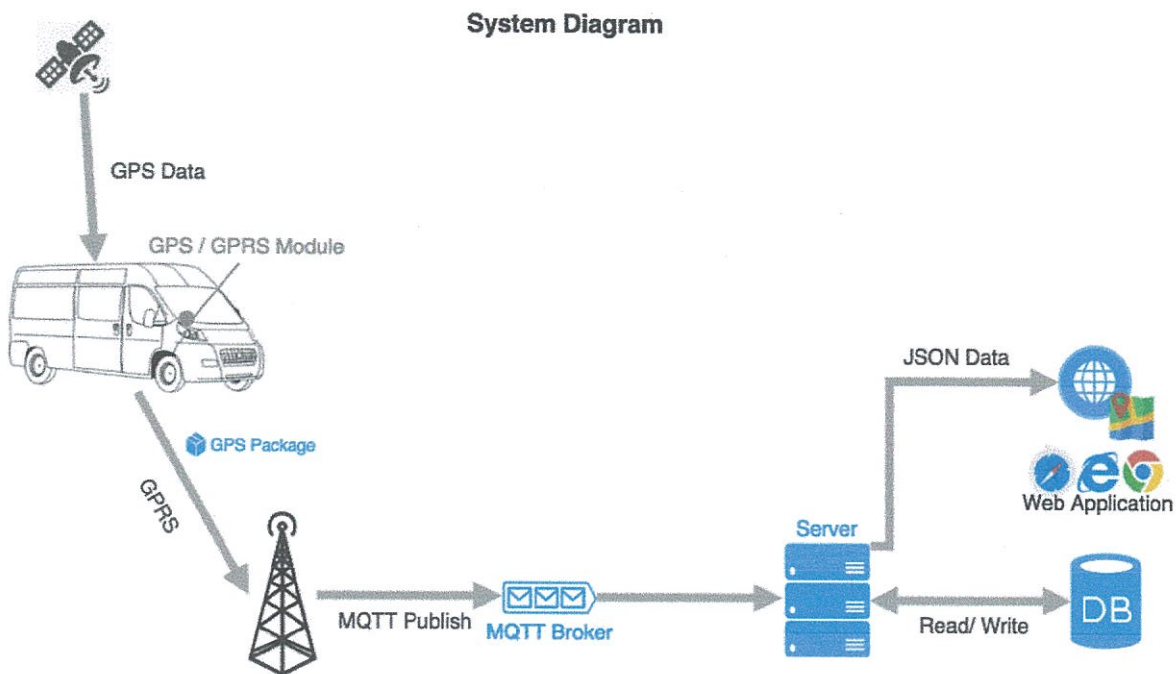
3.1 ความต้องการของระบบ

3.1.1 Specification

- 1) แสดงผลหมายเลขทะเบียนรถ, รหัสประจำตัวประชาชนคนขับรถผู้โดยสารสาธารณะ, ชื่อคนขับรถผู้โดยสารสาธารณะ และตำแหน่งรถผู้โดยสารสาธารณะในแผนที่ออนไลน์
- 2) สามารถวิเคราะห์ และแสดงผลความเร็วของรถผู้โดยสารสาธารณะในแต่ละเส้นทางได้
- 3) สามารถค้นหารถผู้โดยสารสาธารณะด้วย หมายเลขทะเบียนรถเพื่อติดตามรถผู้โดยสารสาธารณะบนแผนที่ออนไลน์ได้
- 4) บนแผนที่ออนไลน์จะแสดงรถผู้โดยสารสาธารณะทุกคันที่อยู่ในขอบเขตของแผนที่ที่แสดงอยู่
- 5) สามารถดูเส้นทางการเดินทางของรถผู้โดยสารสาธารณะแต่ละคันบนแผนที่ออนไลน์
- 6) ผู้ดูแลระบบสามารถแก้ไขข้อมูลของรถผู้โดยสารสาธารณะ และผู้ขับขี่ในฐานะข้อมูล

3.2 โครงสร้างในการพัฒนาระบบ

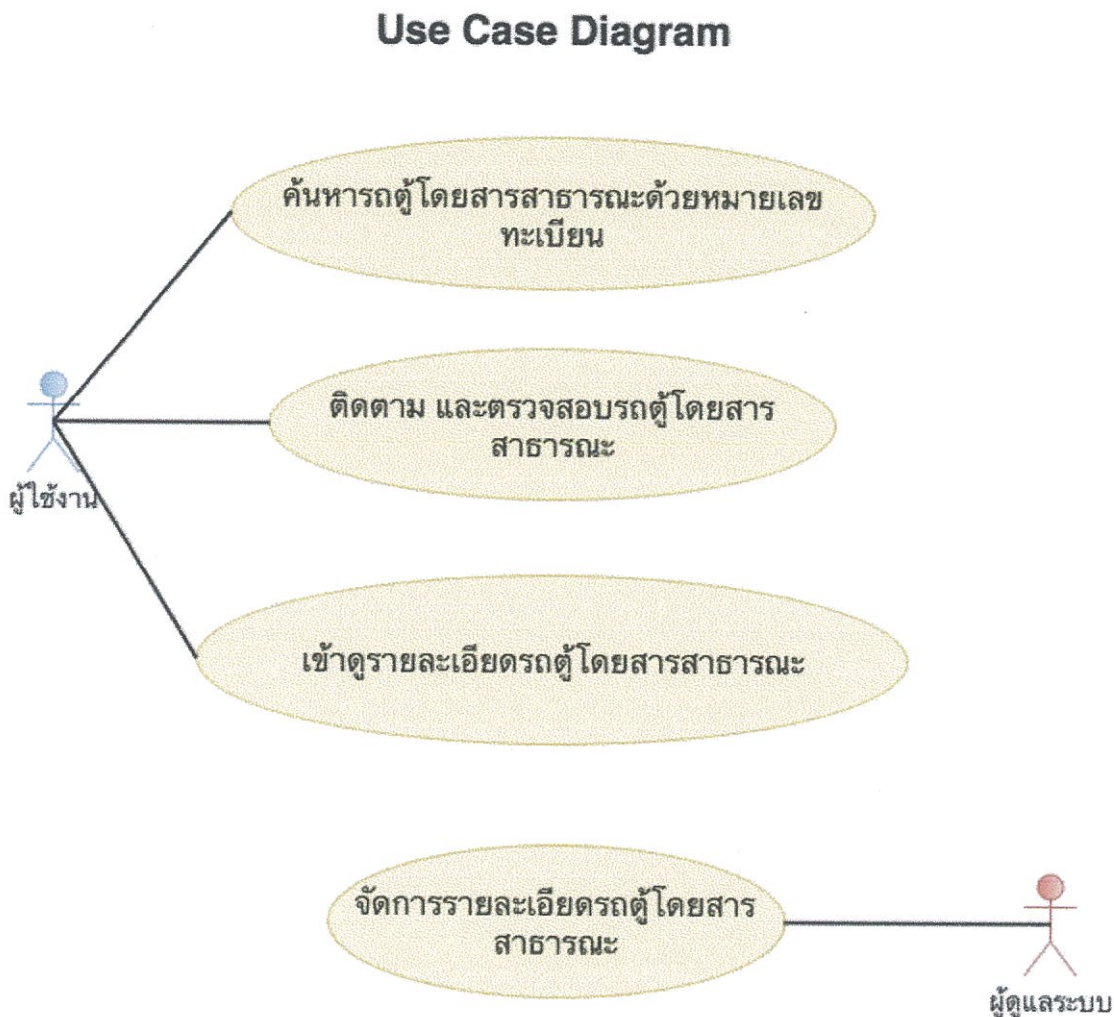
3.2.1 ภาพรวมของระบบ



รูป 3.1 ภาพรวมของระบบ

3.2.2 ความสามารถของระบบ

ความสามารถของระบบแบ่งตามประเภทผู้ใช้ซึ่งมี 2 ประเภท คือ ผู้ใช้ทั่วไป และผู้ดูแลระบบซึ่งทำเป็นแผนภาพ ดังรูป 3.2



รูป 3.2 ความสามารถของระบบ

รายละเอียดความสามารถของระบบในแต่ละส่วนอธิบายได้ ดังตาราง 3.1 - 3.4

ตาราง 3.1 ความสามารถของระบบในการจัดการรายละเอียดรถตู้โดยสารสาธารณะ

Use Case Name	จัดการรายละเอียดรถตู้โดยสารสาธารณะ
Use Case Purpose	ผู้ดูแลระบบสามารถจัดการรายละเอียดของรถตู้โดยสารสาธารณะในแต่ละคัน
Pre-conditions	ผู้ใช้งานต้องทำการลงทะเบียน และเข้าสู่ระบบในฐานะผู้ดูแลระบบ

ตาราง 3.1 ความสามารถของระบบในการจัดการรายละเอียดรถตู้โดยสารสาธารณะ (ต่อ)

Post-conditions	ผู้ดูแลระบบสามารถจัดการข้อมูลของรถตู้โดยสารสาธารณะได้
Limitations	ผู้ใช้งานทั่วไปไม่สามารถเข้าถึงได้
Primary Scenario	<ol style="list-style-type: none"> 1. เข้าสู่การจัดการรถตู้โดยสารสาธารณะ 2. ผู้ดูแลระบบต้องการเพิ่มข้อมูลรถตู้โดยสารสาธารณะ หรือแก้ไขข้อมูลรถตู้โดยสารสาธารณะ 3. กรอกข้อมูลรถตู้โดยสารสาธารณะที่ต้องการเพิ่ม/กรอกหมายเลขทะเบียนรถตู้ 4. ระบบร้องขอรหัสผ่านเพื่อทำการแก้ไขข้อมูล
Alternative Scenario	<p>3a ข้อมูลรถตู้โดยสารสาธารณะที่ต้องการเพิ่มซ้ำกับข้อมูลที่มีอยู่แล้ว</p> <ol style="list-style-type: none"> 1. ระบบทำการตรวจสอบข้อมูลรถตู้โดยสารสาธารณะในฐานข้อมูล แสดงข้อผิดพลาดให้กับผู้ใช้ 2. ร้องขอข้อมูลใหม่ <p>3b ไม่มีหมายเลขทะเบียนรถตู้โดยสารสาธารณะที่กรอก</p> <ol style="list-style-type: none"> 1. ระบบตรวจสอบหมายเลขทะเบียนไม่พบ แสดงข้อผิดพลาดให้กับผู้ใช้ 2. ระบบกลับไปสู่หน้าเลือกการทำงานอีกครั้ง <p>4a ผู้ใช้ทำการกรอกรหัสผ่านผิด</p> <ol style="list-style-type: none"> 1. ระบบตรวจสอบรหัสผ่านของผู้ใช้ไม่ถูกต้อง แสดงข้อผิดพลาดให้กับผู้ใช้ 2. ระบบร้องขอรหัสผ่านอีกครั้ง

ตาราง 3.2 ความสามารถของระบบในการติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ

Use Case Name	ติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ
Use Case Purpose	ผู้ใช้งานสามารถติดตามรถตู้โดยสารสาธารณะคันที่ต้องการได้ตลอดเวลา
Pre-conditions	ผู้ใช้ต้องทำการลงทะเบียนเข้าสู่ระบบ
Post-conditions	ผู้ใช้รับรู้เส้นทางการเดินรถของรถตู้โดยสารสาธารณะคันที่ต้องการติดตามได้

ตาราง 3.2 ความสามารถของระบบในการติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ (ต่อ)

Limitations	ใช้เวลาในการรอข้อมูล 1 วินาที
Primary Scenario	<ol style="list-style-type: none"> 1. ผู้ใช้กรอกทะเบียนรถตู้โดยสารสาธารณะที่ต้องการติดตาม 2. ระบบตรวจสอบทะเบียน 3. ระบบแสดงข้อมูลตำแหน่ง และความเร็วของรถตู้
Alternative Scenario	<p>2a ระบบไม่พบทะเบียนรถตู้โดยสารสาธารณะที่ต้องการติดตาม</p> <ol style="list-style-type: none"> 1. ระบบทำการตรวจสอบทะเบียนรถตู้โดยสารสาธารณะในฐานข้อมูลไม่พบ แสดงข้อผิดพลาดให้กับผู้ใช้ 2. ระบบกลับไปสู่หน้าหลัก <p>3a ระบบทำการดึงข้อมูลรถตู้โดยสารสาธารณะใหม่ทุก 5 วินาที</p>

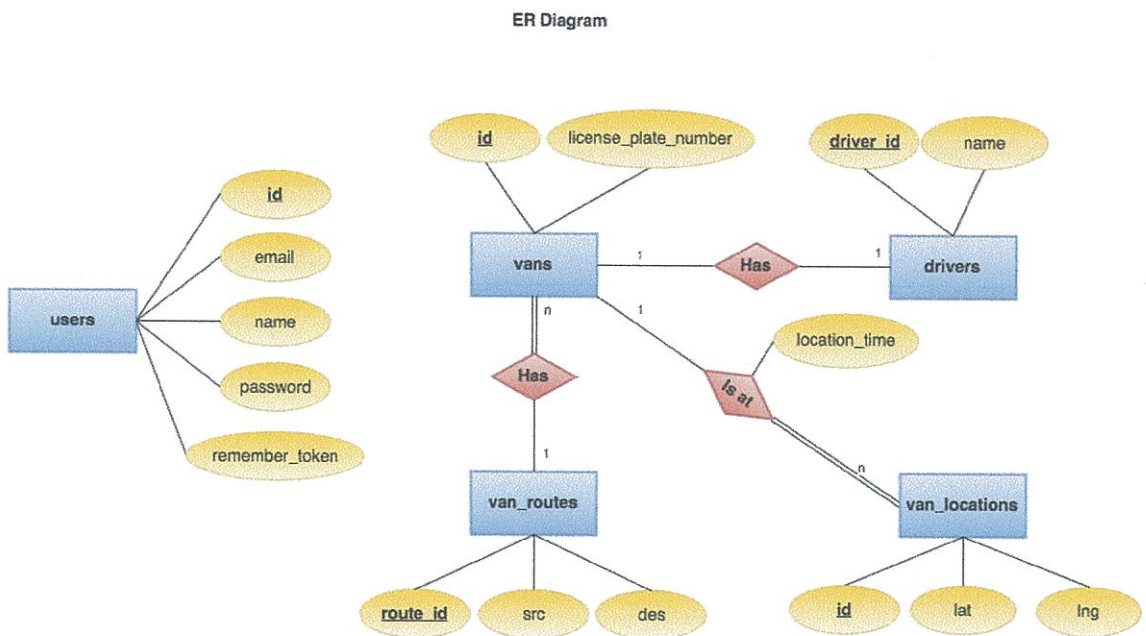
ตาราง 3.3 ความสามารถของระบบในการค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน

Use Case Name	ค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน
Use Case Purpose	ผู้ใช้ค้นหาข้อมูลรถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน
Pre-conditions	ผู้ใช้งานต้องทำการลงทะเบียนเข้าสู่ระบบ
Post-conditions	ผู้ใช้ได้รับข้อมูลของรถตู้ที่ทำการค้นหา
Limitations	-
Primary Scenario	<ol style="list-style-type: none"> 1. เข้าสู่การค้นหาข้อมูลรถตู้โดยสารสาธารณะ 2. กรอกเลขทะเบียนรถตู้โดยสารสาธารณะ <p>ระบบแสดงข้อมูลรถตู้โดยสารสาธารณะที่ทำการเลือก</p>
Alternative Scenario	<p>2a เลขทะเบียนรถตู้ไม่พบบนฐานข้อมูล</p> <ol style="list-style-type: none"> 1. ระบบทำการตรวจสอบเลขทะเบียนไม่พบแสดงข้อผิดพลาดให้กับผู้ใช้ 2. ระบบกลับไปสู่หน้าหลัก

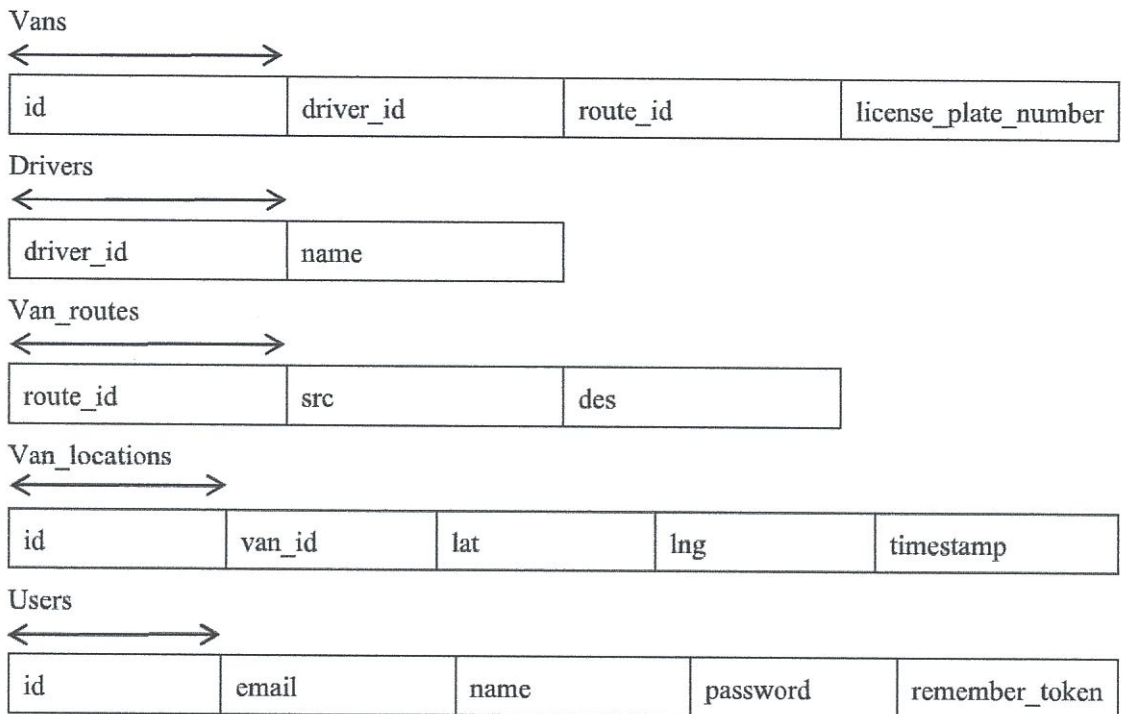
ตาราง 3.4 ความสามารถของระบบในการเข้าดูรายละเอียดรถตู้โดยสารสาธารณะ

Use Case Name	เข้าดูรายละเอียดรถตู้โดยสารสาธารณะ
Use Case Purpose	ผู้ใช้งานสามารถดูข้อมูลต่าง ๆ ของรถตู้โดยสารสาธารณะ และคนขับได้
Pre-conditions	ผู้ใช้งานต้องทำการลงทะเบียนเป็นผู้ใช้ระบบ
Post-conditions	ผู้ใช้งานได้รับข้อมูลรถตู้โดยสารสาธารณะที่ทำการเลือก
Limitations	-
Primary Scenario	1. ผู้ใช้งานทำการเลือกรถตู้ที่ต้องการดูข้อมูลที่แสดงอยู่บนแผนที่ 2. ระบบแสดงหน้าต่างที่แสดงข้อมูลของรถตู้ที่ผู้ใช้งานเลือก
Alternative Scenario	-

3.2.3 การออกแบบฐานข้อมูล



รูป 3.3 Entity-Relationship



ตาราง 3.5 รูปแบบของตาราง Vans ในฐานข้อมูล

Key	Attribute	Type	Null able	Description
PK	id	INT	false	หมายเลข ID สร้าง โดยการ Auto increment
FK	driver_id	VARCHAR	false	หมายเลขประจำตัวประชาชนของคนขับ
FK	route_id	VARCHAR	false	หมายเลขสายของรถตู้
-	license_plate_number	VARCHAR	false	หมายเลขทะเบียนรถตู้

ตาราง 3.6 รูปแบบของตาราง Drivers ในฐานข้อมูล

Key	Attribute	Type	Null able	Description
PK	driver_id	VARCHAR	false	หมายเลขประจำตัวประชาชนของคนขับรถตู้
-	name	VARCHAR	false	ชื่อของคนขับรถตู้

ตาราง 3.7 รูปแบบของตาราง Van_routes ในฐานข้อมูล

Key	Attribute	Type	Null able	Description
PK	route_id	VARCHAR	false	หมายเลขสายของรถตู้
-	src	VARCHAR	false	ชื่อสถานที่ของจุดเริ่มต้นของรถตู้ที่เริ่มออกเดินทาง
-	des	VARCHAR	false	ชื่อสถานที่ของจุดปลายทางของรถตู้ที่จะไปถึงที่หมาย

ตาราง 3.8 รูปแบบของตาราง Van_locations ในฐานข้อมูล

Key	Attribute	Type	Null able	Description
PK	id	INT	false	หมายเลข ID สร้างโดยการ Auto increment
FK	van_id	INT	false	หมายเลข ID ของ Van ใช้อ้างอิงถึงรถตู้
-	latitude	DOUBLE	false	ตำแหน่งละติจูดบนโลก
-	longitude	DOUBLE	false	ตำแหน่งลองจิจูดบนโลก
-	timestamp	DATETIME	false	เวลาที่ระบบได้รับตำแหน่งของรถตู้

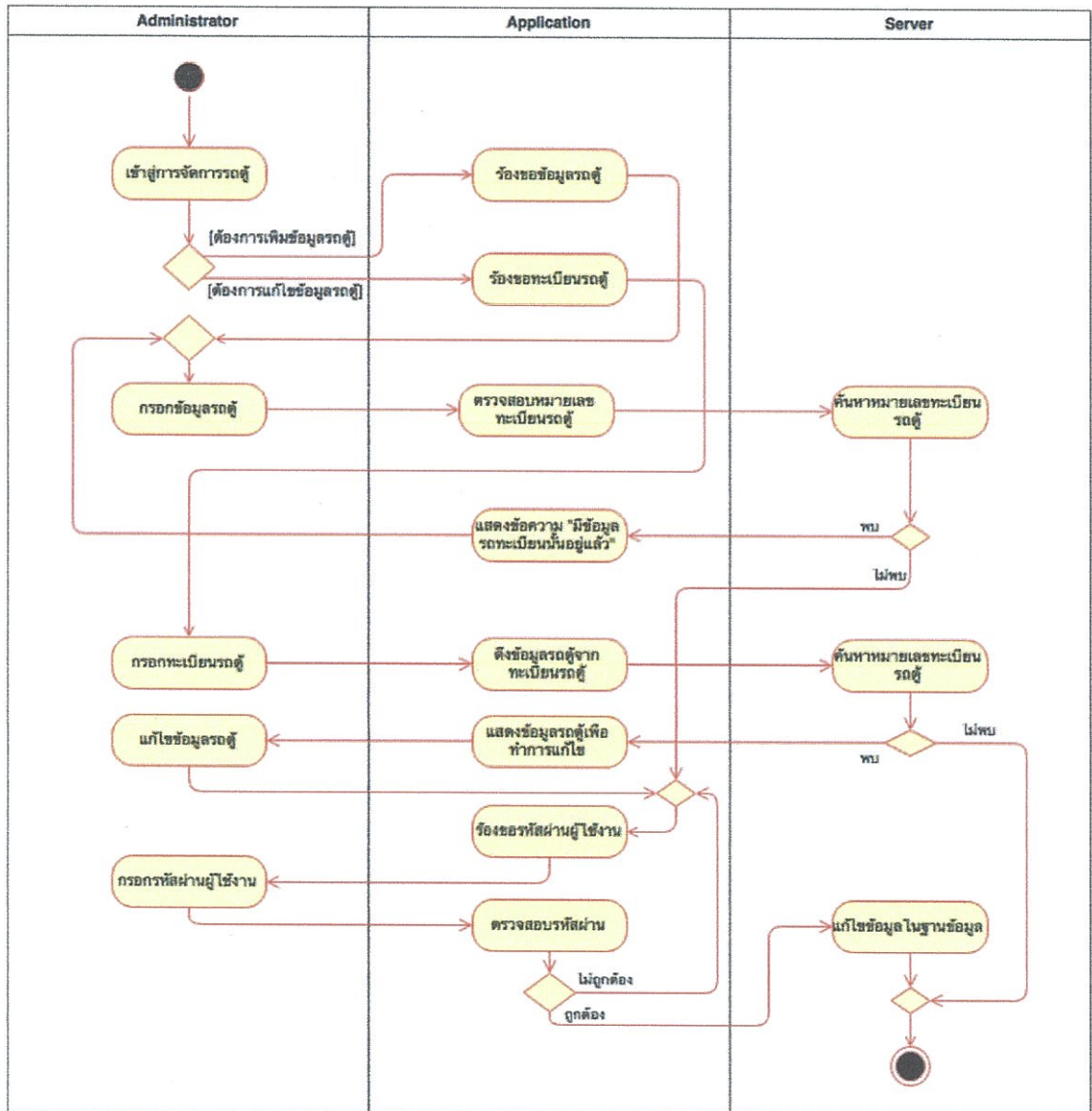
ตาราง 3.9 รูปแบบของตาราง Users ในฐานข้อมูล

Key	Attribute	Type	Null able	Description
PK	id	INT	false	หมายเลข ID สร้างโดยการ Auto increment
-	email	VARCHAR	false	อีเมลของผู้ใช้งาน
-	name	VARCHAR	false	ชื่อของผู้ใช้งาน
-	password	VARCHAR	false	พาสเวิร์ดของผู้ใช้งาน
-	remember_token	VARCHAR	true	token ที่ใช้จดจำว่าผู้ใช้งานทำการ login แล้ว โดยจะถูกสร้างใหม่ทุกครั้งเมื่อผู้ใช้งานทำการ login

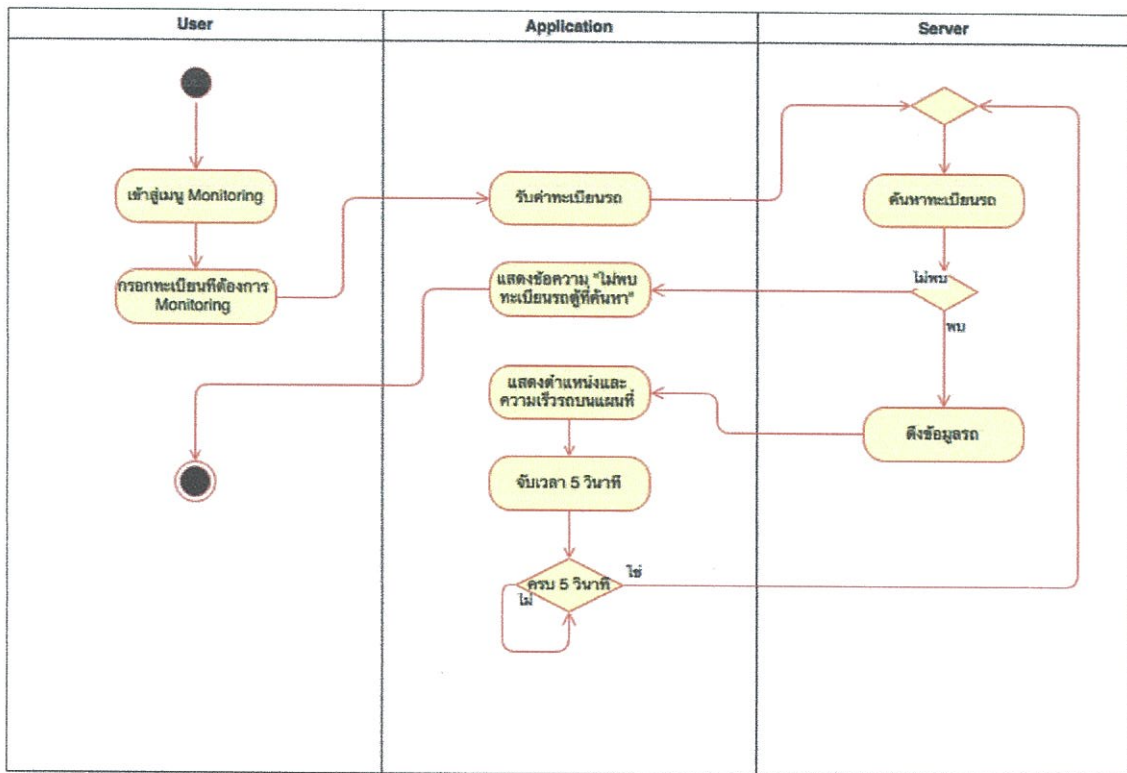
3.2.4 Activity Diagram

Activity Diagram ของระบบมี 4 Diagram ซึ่งมีการทำงาน

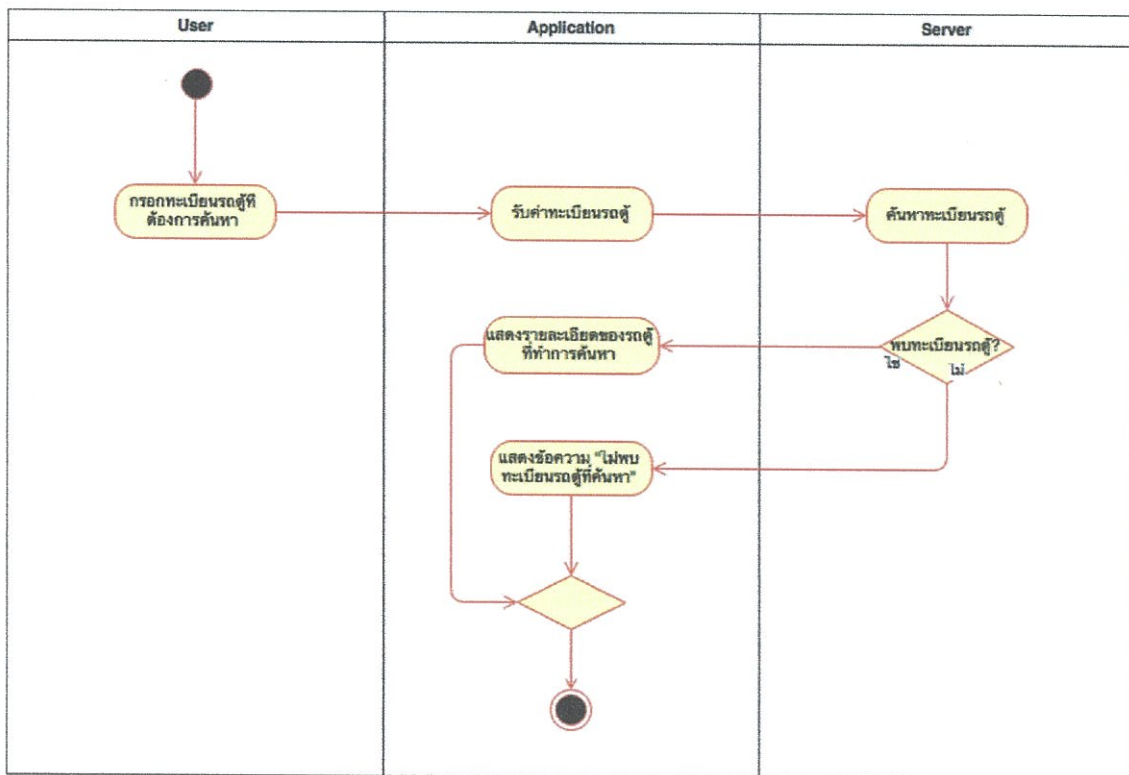
- 1) Activity Diagram การจัดการรายละเอียดครดตู้โดยสารสาธารณะ ดังรูป 3.4
- 2) Activity Diagram การติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ ดังรูป 3.5
- 3) Activity Diagram การค้นหาหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน ดังรูป 3.6
- 4) Activity Diagram การเข้าดูรายละเอียดครดตู้โดยสารสาธารณะ ดังรูป 3.7



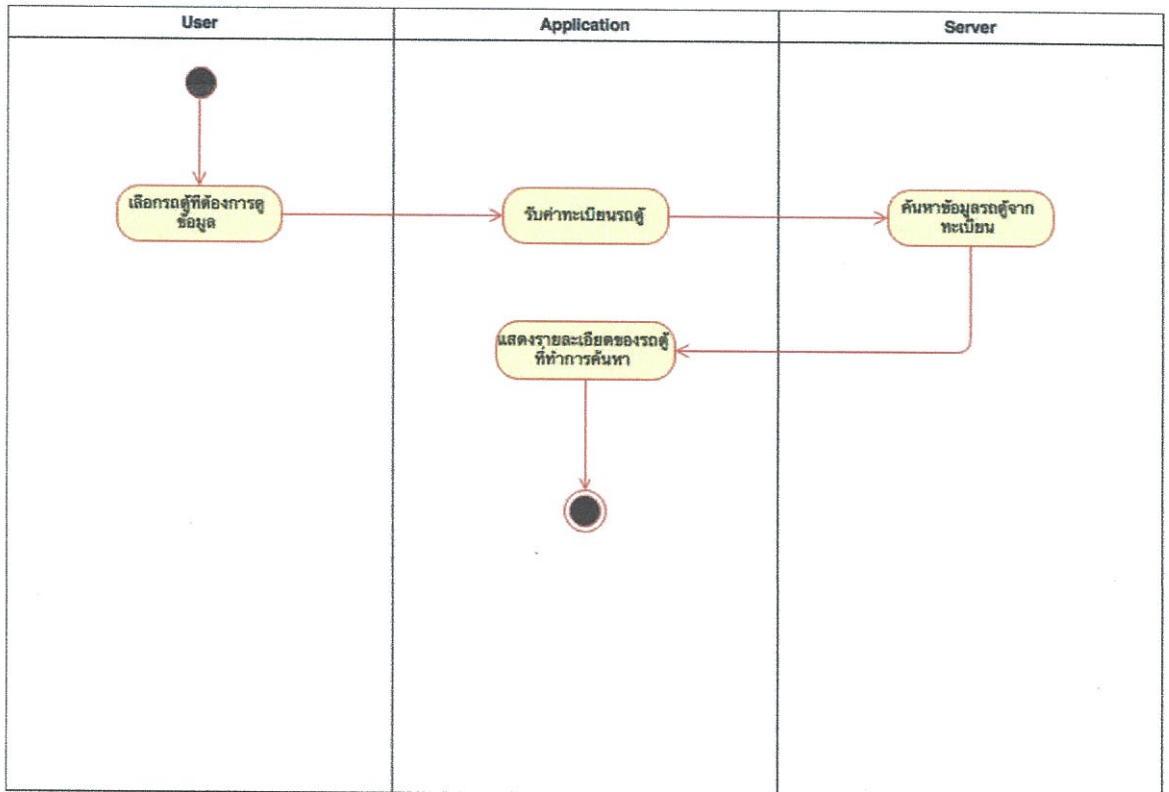
รูป 3.4 Activity Diagram การจัดการรายละเอียดครดตู้โดยสารสาธารณะ



รูป 3.5 Activity Diagram การติดตาม และตรวจสอบรถตู้โดยสารสาธารณะ



รูป 3.6 Activity Diagram การค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน



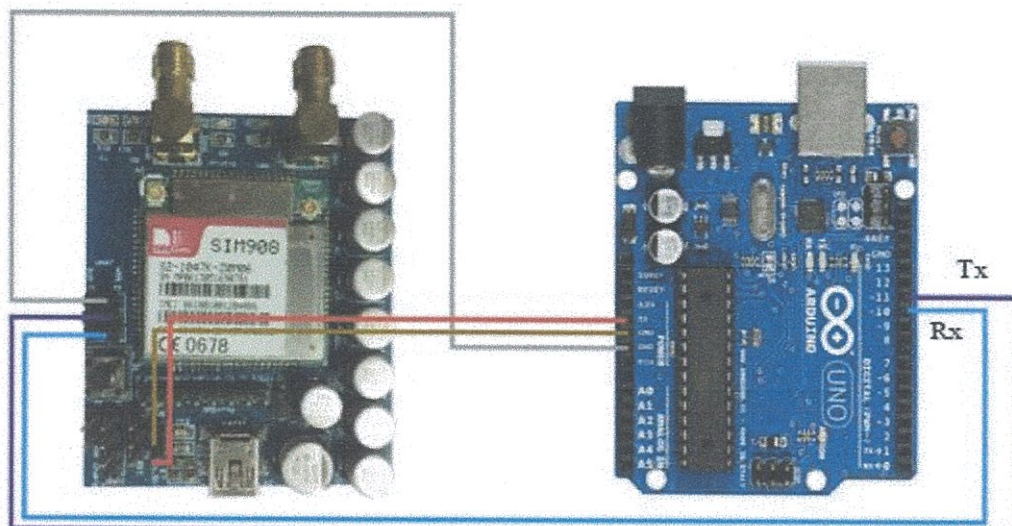
รูป 3.7 Activity Diagram การเข้าดูรายละเอียดรถตู้โดยสารสาธารณะ

3.2.5 การออกแบบการทำงานของอุปกรณ์ฮาร์ดแวร์

ทำการเชื่อมต่อ Arduino กับ SIM908 Module โดยทำการเชื่อมต่อสายดังนี้

- 1) สายสีม่วง เชื่อมต่อระหว่าง Pin11 ของ Arduino ซึ่งเป็น Tx กับ Pin2 (J3) ซึ่งเป็น Rx GSM ของ SIM908 Module
- 2) สายสีน้ำเงิน เชื่อมต่อระหว่าง Pin10 ของ Arduino ซึ่งเป็น Rx กับ Pin3 (J3) ซึ่งเป็น Tx GSM ของ SIM908 Module
- 3) สายสีเทา เชื่อมต่อระหว่าง Ground ของ Arduino กับ Pin1 (J3) ซึ่งเป็น Ground GSM ของ SIM908 Module
- 4) สายสีแดง เชื่อมต่อระหว่าง Vcc ของ Arduino กับ Pin2 (J5) ซึ่งเป็น Vcc ของ SIM908 Module ทั้งหมด
- 5) สายสีน้ำตาล เชื่อมต่อระหว่าง Ground ของ Arduino กับ Pin1 (J5) ซึ่งเป็น Ground ของ SIM908 Module ทั้งหมด

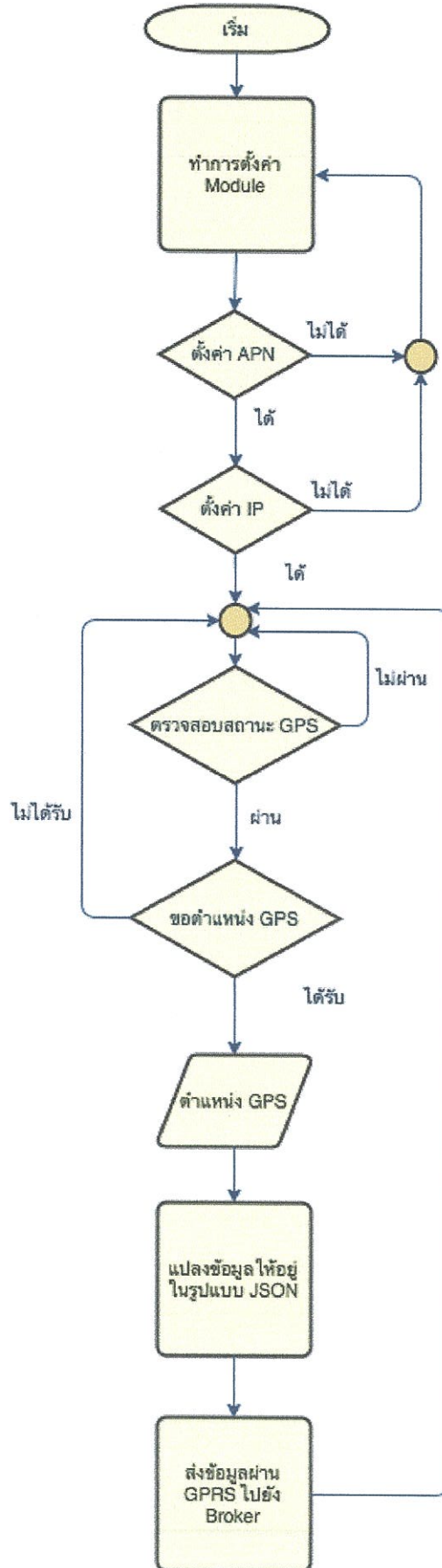
ตัวอย่างการเชื่อมต่อ ดังรูป 3.8



รูป 3.8 การเชื่อมต่อระหว่าง Arduino กับ SIM908 Module

ทำการติดต่อระหว่าง Arduino กับ SIM908 Module ด้วยคำสั่งภาษา C++ โดยมีฟังก์ชันการทำงานดังต่อไปนี้

- 1) ให้ Arduino ทำการส่ง AT Command ไปยัง SIM908 Module เพื่อทำการเปิดการทำงาน และตั้งค่าการทำงานต่าง ๆ และ APN
- 2) ให้ Arduino ทำการส่ง AT Command ไปยัง SIM908 Module เพื่อเป็นการรับค่าของ ip address ไว้ใช้ในการเชื่อมต่อกับ Internet
- 3) ให้ Arduino ทำการส่ง AT Command ไปยัง SIM908 Module เพื่อตั้งค่าโหมดการทำงาน และถามสถานะของ GPS ว่าพร้อมทำงานหรือไม่ หากพร้อม GPS จะตอบกลับ Arduino ว่า 2D Fix หรือ 3D Fix เป็นการบอกว่า GPS สามารถหาดำแหน่งได้แล้ว ในรูปแบบ 2 มิติ หรือ 3 มิติ
- 4) ให้ Arduino ทำการส่ง AT Command ไปยัง SIM908 Module เพื่อรับค่าตำแหน่งจาก GPS โดยนำค่าในตัวแปรชื่อ frame ไว้และทำการตัดค่าใน frame ออกเป็นแต่ละส่วนซึ่งมีค่าดังนี้ Latitude, Longitude และ Time และเหล่านี้มาทำการเชื่อมต่อกับข้อมูลที่ทำกรกำหนดค่าไว้ซึ่งมีค่า id (หมายเลขรถตู้) โดยให้อยู่ในรูปแบบของ JSON
- 5) ให้ Arduino ทำการส่ง AT Command ไปยัง SIM908 Module เพื่อให้เปิด TCP Connection กับ Server และส่งข้อมูลที่ต้องการออกไป



รูป 3.9 Flow Chart การทำงานของอุปกรณ์ฮาร์ดแวร์

3.2.6 การออกแบบ MQTT Packet

ในส่วนของ Payload ข้อมูลจะอยู่ในรูปแบบ json โดยมีค่าต่าง ๆ ตามตัวอย่าง 3.1

ตัวอย่าง 3.1 ข้อมูลรูปแบบ json

```
{
  "Van": {
    "id": INT,
    "loca": {
      "lat": FLOAT,
      "long": FLOAT
    },
    "Time": INT
  }
}
```

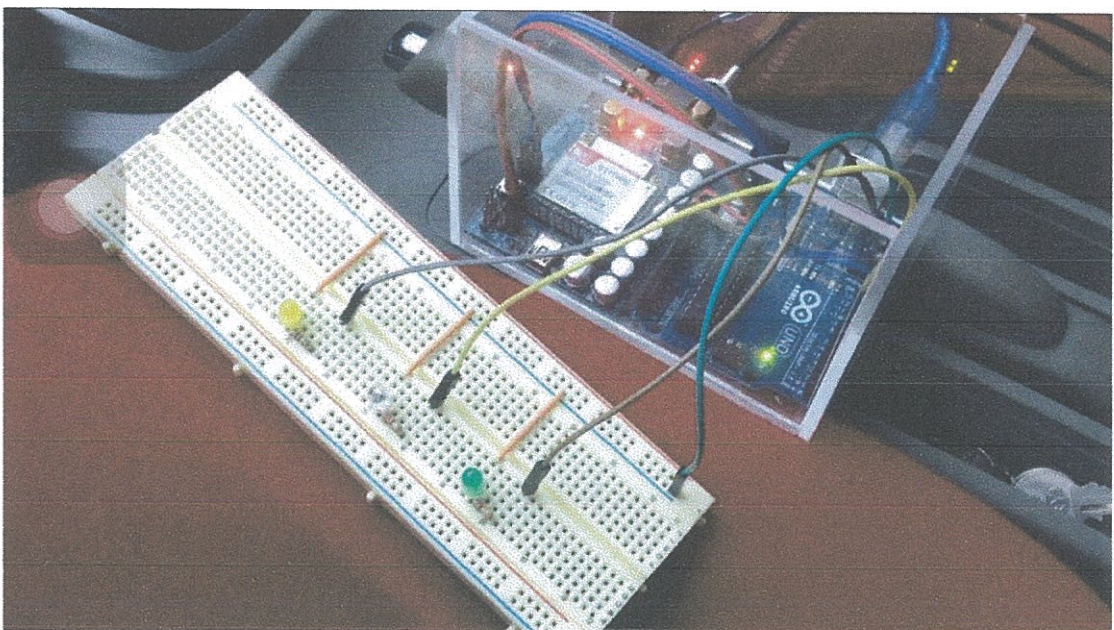
บทที่ 4

การทดลองและผลการทดลอง

4.1 การรับข้อมูลจากอุปกรณ์ฮาร์ดแวร์และการแสดงข้อมูลของรถบนแผนที่โดยการนำข้อมูลจากฐานข้อมูลพร้อมแสดงรายละเอียดของรถ

การทดลองจะทำการรับข้อมูลที่อุปกรณ์ฮาร์ดแวร์ที่นำไปติดตั้งบนรถตู้โดยสารสาธารณะ และทำการส่ง MQTT Packet ไปยัง MQTT Broker จากนั้นเซิร์ฟเวอร์จะทำการ Subscribe เพื่อเก็บข้อมูลของ MQTT Packet ลงในฐานข้อมูล และเว็บเซิร์ฟเวอร์ก็จะทำการนำข้อมูลในฐานข้อมูลนั้นมาแสดงบนแผนที่ออนไลน์ ผ่าน Web browser

- 1) นำอุปกรณ์ฮาร์ดแวร์ไปติดตั้งบนรถเสียบสาย USB เพื่อจ่ายไฟให้กับอุปกรณ์ฮาร์ดแวร์ และทำการเปิด Module SIM908 เพื่อเริ่มใช้งานอุปกรณ์ฮาร์ดแวร์จะมีไฟ LED แสดงสถานะ 3 ดวง ประกอบด้วยไฟ LED สีเขียว สีแดง และสีเหลือง โดยไฟ LED สีเขียวจะใช้บ่งบอกการตั้งค่า Module ให้พร้อมใช้งานหากถูกต้องตามที่ตั้งไว้ ไฟ LED สีเขียวจะติดค้าง และไฟ LED สีแดงใช้บ่งบอกว่า Module พร้อมสำหรับการรับตำแหน่ง GPS หากพร้อม GPS จะตอบกลับ Arduino ว่า 2D Fix หรือ 3D Fix ถ้าหากสามารถหาตำแหน่งได้แล้ว ในรูปแบบ 2 มิติ หรือ 3 มิติ ไฟสีแดงจะกระพริบ และทำการสร้าง Data ในรูปแบบ JSON Format ไฟสีเหลืองจะบ่งบอกถึงการส่งข้อมูลไปยัง Broker สำเร็จ ถ้าหากส่งข้อมูลออกไปได้สำเร็จไฟสีเหลืองจะกระพริบ



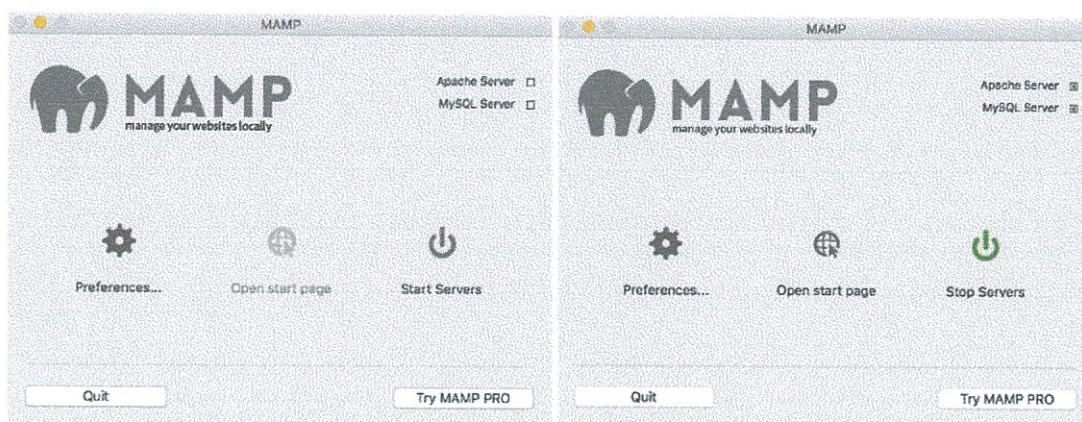
รูป 4.1 อุปกรณ์ฮาร์ดแวร์ที่ทำการติดตั้งบนรถ

- 2) ทำการรัน laravel server ด้วยคำสั่ง php artisan serve เว็บไซต์จะถูกรันบน port 8000

```
Laravel — php • php artisan serve — 80x24
Last login: Thu Apr 14 12:20:05 on console
[Theerats-MacBook-Pro:~ Theerat$ cd ~/CE/Project/Laravel/
[Theerats-MacBook-Pro:Laravel Theerat$ php artisan serve
Laravel development server started on http://localhost:8000/
```

รูป 4.2 ผลการรัน Laravel Server

- 3) เปิดโปรแกรม MAMP และกด Start Servers เพื่อทำการเปิด Service ของ MySQL Server



ก)

ข)

รูป 4.3 หน้าต่างโปรแกรม MAMP

ก) ก่อนกดปุ่ม Start Servers

ข) หลังกดปุ่ม Start Servers

- 4) ทำการรัน submqtt_with_db.php เพื่อทำการ Subscribe MQTT Packet ที่ส่งมาจากอุปกรณ์ฮาร์ดแวร์ และทำการแปลงจาก JSON Data มาเป็น PHP Object เพื่อทำการเก็บข้อมูลลงฐานข้อมูลด้วยคำสั่ง php submqtt_with_db.php

โปรแกรม 4.1 submqtt_with_db.php

```
<?php
require("../phpMQTT/phpMQTT.php");

$mqtt = new phpMQTT("161.246.5.195",1883, "phpMQTT");
//Change client name to something unique
```

โปรแกรม 4.1 submqtt_with_db.php (ต่อ)

```

if (!$mqtt->connect($clean = true, $will = NULL, "",
"")){
    exit(1);
}

$topics['test'] = array("qos"=>0, "function"=>"procmsg");
$mqtt->subscribe($topics,0);

while ($mqtt->proc()) {
}

$mqtt->close();

function procmsg($topic,$msg) {
    date_default_timezone_set("UTC");
    date_default_timezone_set("Asia/Bangkok");
    global $done;
    $done = 1;

    $json = json_decode($msg);
    $vanObj = $json->Van;
    $vanLocation = $vanObj->loca;
    $id = $vanObj->id;

    $latVal = doubleval($vanLocation->lat)/100;
    $latDegree = doubleval(floor($latVal));
    $latDeci = (((($latVal - $latDegree)*100)/60.0);
    $lat = $latDegree + $latDeci;

    $lngVal = doubleval($vanLocation->long)/100;
    $lngDegree = doubleval(floor($lngVal));
    $lngDeci = (((($lngVal - $lngDegree)*100)/60);
    $lng = $lngDegree + $lngDeci;

    $time = $vanObj->Time;

    $servername = "localhost";
    $username = "root";
    $password = $secret;
    $dbname = "forge";
    // Create connection
    $conn = new mysqli($servername, $username,
$password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn-
>connect_error);    }
}

```

โปรแกรม 4.1 submqtt_with_db.php (ต่อ)

```

    $sql = "INSERT INTO van_locations (van_id, lat, lng,
location_time)
    VALUES ($id, $lat, $lng, $time)";

    if ($conn->query($sql) === TRUE) {
        $utcTime = date('c', mktime(intval(substr($time,
8, 2)), intval(substr($time, 10, 2)),
intval(substr($time, 12)), intval(substr($time, 4, 2)),
intval(substr($time, 6, 2)), intval(substr($time, 0,
4))));
        echo "Create new data van no." . $id . " at
" . $utcTime . " : " . $lat . " , " . $lng;
        echo "\n";
    } else {
        echo "Error: " . $sql . "\n" . $conn->error;
    }
    $conn->close();
}

function hasTimedout() {
    global $start_time;
    return (time() - $start_time > 2); //waits up to 10 sec
}
?>

```

ข้อมูลที่อยู่ในข้อมูลที่ทำการส่งมาจากอุปกรณ์ฮาร์ดแวร์ และที่รับมาจากอุปกรณ์ฮาร์ดแวร์ ดังรูป 4.4 ข้อมูลที่เซิร์ฟเวอร์ ทำการ Subscribe จาก MQTT Broker และทำการเก็บข้อมูลลงในฐานข้อมูลดังรูป 4.5 ข้อมูลในฐานข้อมูลดังรูป 4.6 โดยข้อมูลในกรอบสีแดงของรูป 4.4, 4.5 และ 4.6 มีความสอดคล้องกัน

```

/dev/cu.usbmodem1421 (Arduino Uno)
Send
Message is sent
send CIPCLOSE answer ERROR
AT+CIPSTART="TCP","161.246.5.195","1883"
send MQTT answer ERROR
send CIPCLOSE answer ERROR
loop
1
send AT Status answer 3D
{"Van":{"id":"1","loca":{"lat":"1343.026870","long":"10048.945552"},"Time":"20160422112838"}}
2
send AT Status answer 3D
{"Van":{"id":"1","loca":{"lat":"1343.025455","long":"10048.944232"},"Time":"20160422112846"}}
AT+CIPSTART="TCP","161.246.5.195","1883"
TCP Start OK
send CIPSEND answer OK
Broker Connect
Message is sent
send CIPCLOSE answer ERROR
AT+CIPSTART="TCP","161.246.5.195","1883"
send MQTT answer ERROR
send CIPCLOSE answer ERROR
loop
1
send AT Status answer 3D
{"Van":{"id":"1","loca":{"lat":"1343.016834","long":"10048.972363"},"Time":"20160422112926"}}
2
send AT Status answer 3D
{"Van":{"id":"1","loca":{"lat":"1343.027132","long":"10048.972106"},"Time":"20160422112933"}}
AT+CIPSTART="TCP","161.246.5.195","1883"
TCP Start OK
send CIPSEND answer OK
Broker Connect
send AT Send answer OK
Message is sent
send CIPCLOSE answer OK
AT+CIPSTART="TCP","161.246.5.195","1883"
TCP Start OK
send CIPSEND answer OK
Broker Connect
send AT Send answer OK
Message is sent
send CIPCLOSE answer OK
loop
1
send AT Status answer 3D

```

รูป 4.4 หน้าต่างการทำงานของอุปกรณ์ฮาร์ดแวร์

```

Theerats-MacBook-Pro:MQTT Theerat$ php submqtt_with_db.php
Create new data van no.1 at 2016-04-22T11:16:56+07:00 : 13.7331645167 ,100.79573
675
Create new data van no.1 at 2016-04-22T11:17:15+07:00 : 13.7331647833 ,100.79582
4067
Create new data van no.1 at 2016-04-22T11:25:41+07:00 : 13.71707055 ,100.8156910
33
Create new data van no.1 at 2016-04-22T11:27:57+07:00 : 13.7170941833 ,100.81580
79
Create new data van no.1 at 2016-04-22T11:28:38+07:00 : 13.7171145 ,100.8157592
Create new data van no.1 at 2016-04-22T11:29:26+07:00 : 13.7169472333 ,100.81620
605
Create new data van no.1 at 2016-04-22T11:29:33+07:00 : 13.7171188667 ,100.81620
1767
Create new data van no.1 at 2016-04-22T11:29:49+07:00 : 13.7187103167 ,100.81544
9567
Create new data van no.1 at 2016-04-22T11:29:57+07:00 : 13.7196563833 ,100.81495
3517
Create new data van no.1 at 2016-04-22T11:30:37+07:00 : 13.72439965 ,100.8126467
Create new data van no.1 at 2016-04-22T11:30:45+07:00 : 13.7252207833 ,100.81224
4217

```

รูป 4.5 หน้าต่างการรัน submqtt_with_db.php

(MySQL 5.5.42) local/forge/van_locations

id	van_id	lat	lng	location_time	created_at	updated_at
515	1	13.7292153833	100.77589585	2016-04-22 11:10:29	2016-04-22 18:11:07	0000-00-00 00:00:00
518	1	13.7331645167	100.79573675	2016-04-22 11:16:56	2016-04-22 18:17:23	0000-00-00 00:00:00
517	1	13.7331647833	100.795824067	2016-04-22 11:17:15	2016-04-22 18:17:36	0000-00-00 00:00:00
518	1	13.71707055	100.815691033	2016-04-22 11:25:41	2016-04-22 18:26:09	0000-00-00 00:00:00
519	1	13.7170941833	100.8158079	2016-04-22 11:27:57	2016-04-22 18:28:10	0000-00-00 00:00:00
520	1	13.7171145	100.8157592	2016-04-22 11:28:38	2016-04-22 18:29:08	0000-00-00 00:00:00
521	1	13.7169472333	100.81620605	2016-04-22 11:29:26	2016-04-22 18:29:38	0000-00-00 00:00:00
522	1	13.7171188567	100.816201767	2016-04-22 11:29:33	2016-04-22 18:29:41	0000-00-00 00:00:00
523	1	13.7181703167	100.815449567	2016-04-22 11:29:49	2016-04-22 18:30:02	0000-00-00 00:00:00
524	1	13.7196563833	100.814953517	2016-04-22 11:29:57	2016-04-22 18:30:07	0000-00-00 00:00:00
525	1	13.72439965	100.8126467	2016-04-22 11:30:37	2016-04-22 18:30:49	0000-00-00 00:00:00
526	1	13.7252207833	100.812244217	2016-04-22 11:30:45	2016-04-22 18:30:53	0000-00-00 00:00:00
527	1	13.7295243167	100.8101417	2016-04-22 11:31:23	2016-04-22 18:31:35	0000-00-00 00:00:00
528	1	13.7306202833	100.809390133	2016-04-22 11:31:31	2016-04-22 18:31:40	0000-00-00 00:00:00
529	1	13.7329280667	100.806511933	2016-04-22 11:31:49	2016-04-22 18:32:01	0000-00-00 00:00:00
530	1	13.7394146333	100.805320467	2016-04-22 11:31:57	2016-04-22 18:32:12	0000-00-00 00:00:00
531	1	13.73378455	100.801711167	2016-04-22 11:32:20	2016-04-22 18:32:32	0000-00-00 00:00:00
532	1	13.7336497833	100.800014293	2016-04-22 11:32:28	2016-04-22 18:32:43	0000-00-00 00:00:00
533	1	13.7327345	100.792506883	2016-04-22 11:33:05	2016-04-22 18:33:17	0000-00-00 00:00:00
534	1	13.7326265333	100.791127267	2016-04-22 11:33:14	2016-04-22 18:33:21	0000-00-00 00:00:00
535	1	13.73241385	100.7885976	2016-04-22 11:33:29	2016-04-22 18:33:52	0000-00-00 00:00:00
536	1	13.7321735167	100.78616125	2016-04-22 11:33:46	2016-04-22 18:33:56	0000-00-00 00:00:00
537	1	13.7318157833	100.776055883	2016-04-22 11:35:34	2016-04-22 18:35:53	0000-00-00 00:00:00
538	1	13.7318147167	100.77777215	2016-04-22 11:35:41	2016-04-22 18:35:56	0000-00-00 00:00:00
539	1	13.7321006333	100.785243767	2016-04-22 11:36:14	2016-04-22 18:36:33	0000-00-00 00:00:00
540	1	13.7322951833	100.78692967	2016-04-22 11:36:22	2016-04-22 18:36:37	0000-00-00 00:00:00
541	1	13.7331844333	100.7961768	2016-04-22 11:38:13	2016-04-22 18:38:25	0000-00-00 00:00:00
542	1	13.7330184	100.7947001	2016-04-22 11:38:21	2016-04-22 18:38:30	0000-00-00 00:00:00
543	1	13.7325755167	100.790646433	2016-04-22 11:38:45	2016-04-22 18:38:54	0000-00-00 00:00:00
544	1	13.7320999167	100.785124717	2016-04-22 11:39:24	2016-04-22 18:39:47	0000-00-00 00:00:00
545	1	13.7319230667	100.78198555	2016-04-22 11:39:42	2016-04-22 18:39:50	0000-00-00 00:00:00
546	1	13.7318315	100.7791991	2016-04-22 11:39:58	2016-04-22 18:40:17	0000-00-00 00:00:00
547	1	13.7316579833	100.7750371	2016-04-22 11:40:31	2016-04-22 18:40:50	0000-00-00 00:00:00
548	1	13.7303574833	100.775033367	2016-04-22 11:41:05	2016-04-22 18:41:18	0000-00-00 00:00:00

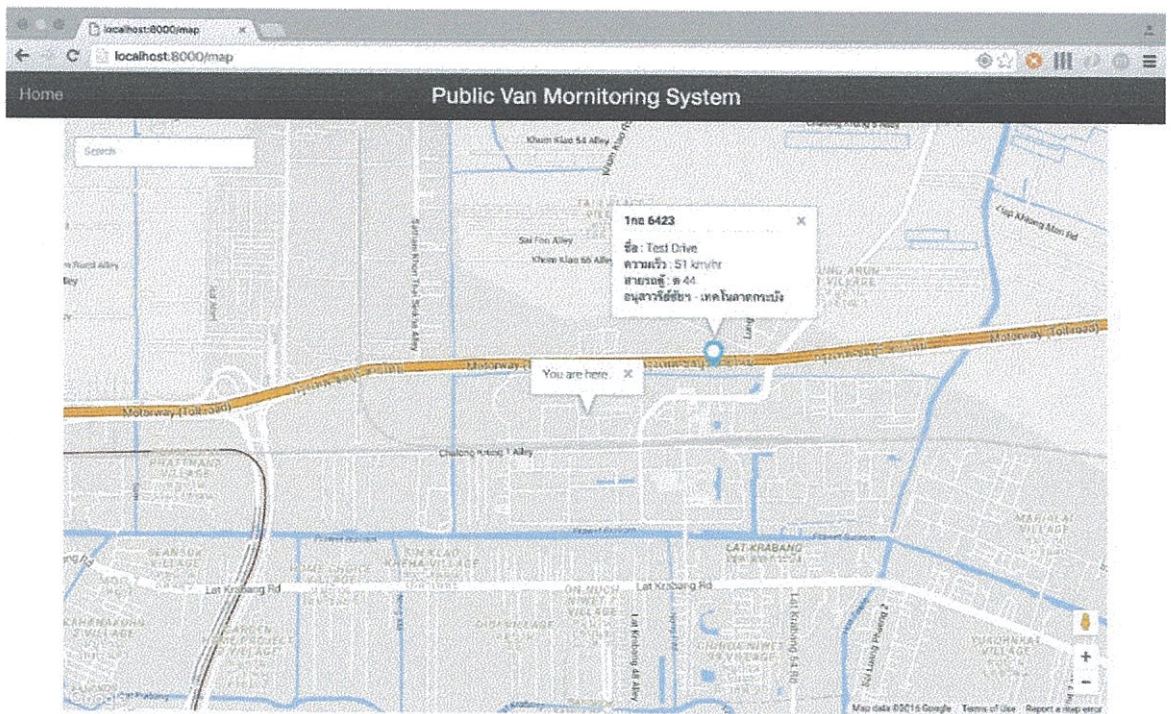
TABLE INFORMATION

- create: 3/22/2558 BE
- engine: InnoDB
- rows: 38
- size: 16.0 KB
- encoding: UTF8
- auto_increment: 553

38 rows in table

รูป 4.6 ข้อมูลในฐานข้อมูลตาราง van_locations

5) เข้าหน้าหลักผ่าน Web browser ด้วยโดเมน <http://localhost:8000/map>



รูป 4.7 หน้าหลักของเว็บไซต์

หน้าเว็บแสดงข้อมูลตำแหน่งที่เราอยู่ เมื่อทำการคลิกที่ marker จะมีหน้าต่างแสดงข้อมูลของรถคู่โดยสารสาธารณะที่อยู่ในแผนที่ ซึ่งมีการคำนวณความเร็วในหน่วย Km/Hr ด้วยฟังก์ชัน calculate_speed โดยใช้ตำแหน่งละติจูด และลองจิจูด 2 ค่าล่าสุดที่ถูกบันทึกลงในฐานข้อมูล

โปรแกรม 4.2 ฟังก์ชัน calculate_speed

```
public static function calculate_speed($lat1, $lon1,
$time1, $lat2, $lon2, $time2) {
    // Convert degrees to radians
    $lat1 = $lat1 * M_PI / 180.0;
    $lon1 = $lon1 * M_PI / 180.0;
    $lat2 = $lat2 * M_PI / 180.0;
    $lon2 = $lon2 * M_PI / 180.0;

    // radius of earth in metres
    $r = 6378100;

    // P
    $rho1 = $r * cos($lat1);
    $z1 = $r * sin($lat1);
    $x1 = $rho1 * cos($lon1);
    $y1 = $rho1 * sin($lon1);

    // Q
    $rho2 = $r * cos($lat2);
    $z2 = $r * sin($lat2);
    $x2 = $rho2 * cos($lon2);
    $y2 = $rho2 * sin($lon2);

    // Dot product
    $dot = ($x1 * $x2 + $y1 * $y2 + $z1 * $z2);
    $cos_theta = $dot / ($r * $r);
    $theta = acos($cos_theta);

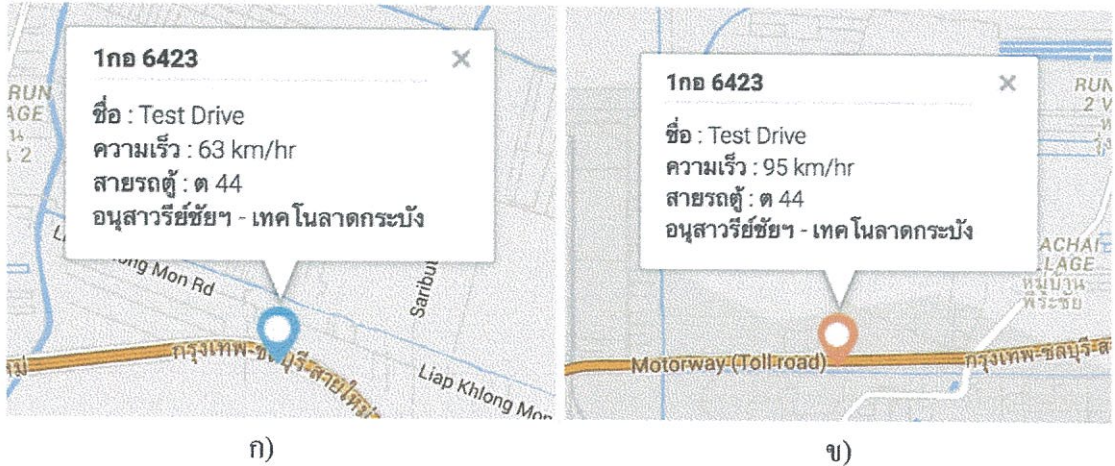
    // Distance in Metres
    $dist = $r * $theta;

    //Delta time
    $deltaTime = (strtotime($time2) - strtotime($time1));

    $speed_mps = $dist / $deltaTime;
    $speed_kmph = ($speed_mps * 3600.0) / 1000.0;

    if($speed_kmph - floor($speed_kmph) > 0.5) {
        return ceil($speed_kmph);
    }
    else return floor($speed_kmph); }
```

หน้าเว็บทำการแสดงข้อมูลตำแหน่งที่เราอยู่ และเมื่อทำการ click ที่ Marker จะมีหน้าต่างแสดงข้อมูลต่าง ๆ ของรถคู่โดยสารสาธารณะ ลักษณะของ Marker จะแบ่งเป็น 2 รูปแบบ คือ Marker สีฟ้า ที่บ่งบอกถึงรถคู่โดยสารสาธารณะนั้นมีความเร็วในการเคลื่อนที่ต่ำกว่า 90 Km/Hr ส่วน Marker สีแดง ใช้ในการบ่งบอกว่ารถคู่โดยสารสาธารณะนั้นมีความเร็วในการเคลื่อนที่สูงกว่า 90 Km/Hr ที่อยู่ในแผนที่ ดังรูปที่ 4.8



รูป 4.8 หน้าต่างแสดงข้อมูลของรถคู่โดยสารสาธารณะ

- ก) Marker ที่แสดงความเร็วในการเคลื่อนที่ต่ำกว่า 90 Km/Hr
- ข) Marker ที่แสดงความเร็วในการเคลื่อนที่สูงกว่า 90 Km/Hr

4.2 การเพิ่มและแก้ไขข้อมูลของรถคู่โดยสารสาธารณะผ่านระบบ Admin

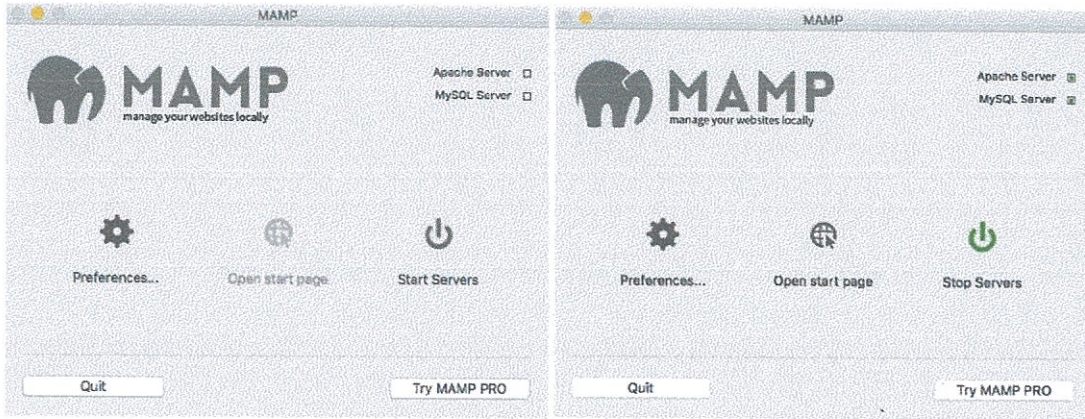
ทดลองการ login เข้าสู่ระบบ Admin ด้วย Username และ Password เพื่อเปลี่ยนข้อมูลต่างของรถคู่โดยสารสาธารณะที่จะแสดงบนแผนที่

- 1) ทำการรัน laravel server ด้วยคำสั่ง php artisan serve เว็บไซต์จะถูกรันบน port 8000

```
Laravel — php * php artisan serve — 80x24
Last login: Thu Apr 14 12:20:05 on console
[Theerats-MacBook-Pro:~ Theerat$ cd ~/CE/Project/Laravel/
[Theerats-MacBook-Pro:Laravel Theerat$ php artisan serve
Laravel development server started on http://localhost:8000/
```

รูป 4.9 ผลการรัน Laravel Server

- 2) เปิดโปรแกรม MAMP และกด Start Servers เพื่อทำการเปิด Service ของ MySQL Server



ก)

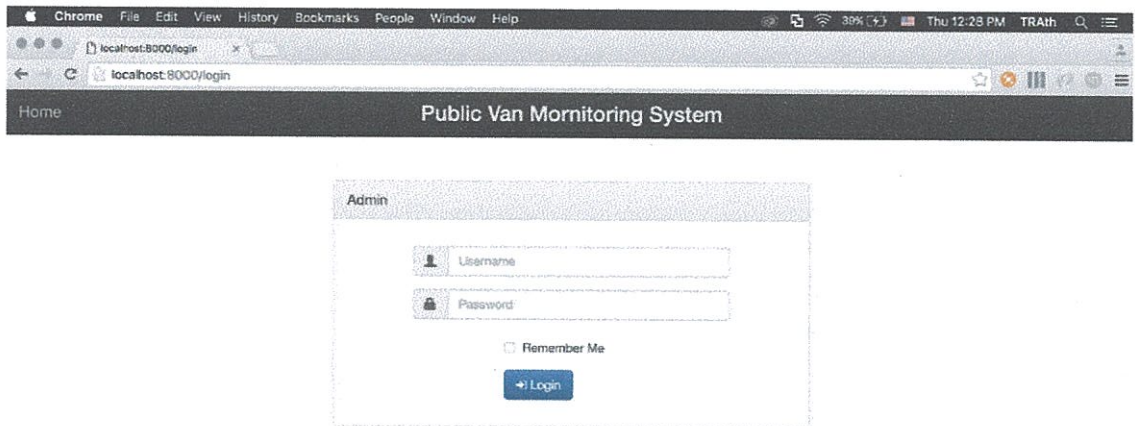
ข)

รูป 4.10 หน้าต่างโปรแกรม MAMP

ก) ก่อนกดปุ่ม Start Servers

ข) หลังกดปุ่ม Start Servers

- 3) เข้าไปยัง <http://localhost:8000/login> เพื่อทำการใส่ username และ password เพื่อทำการ login เข้าสู่ระบบ



รูป 4.11 หน้า login ระบบ Admin ของเว็บไซต์

- 4) เมื่อทำการ login สำเร็จจะแสดงหน้าตารางที่มีข้อมูลของรถตู้โดยสารสาธารณะอยู่ และ Admin สามารถทำการแก้ไขข้อมูลของรถตู้โดยสารสาธารณะแต่ละคัน หรือทำการเพิ่มข้อมูลรถตู้ใหม่โดยสารสาธารณะได้

The screenshot shows a web browser window displaying the 'Public Van Monitoring System' Admin interface. The page contains a table with 12 rows of van data. Each row includes an ID, License Plate Number, Driver ID, Driver Name, and Route Number, along with an 'edit' button.

#	License Plate Number	Driver ID	Driver Name	Route Number	
1	กบ 6423	1234567890123	Test Drive	๓ 44	edit
2	กข 1111	1122334455123	นาย ก ร	A11	edit
3	ก 1	1234567890122	นาย ชี ี	A12	edit
4	ก 2	1234567890126	นาย Test Edit	๓ 44	edit
5	A 1234	1234567890123	Test Drive	๓ 44	edit
6	A 111	1234567890121	นาย ชี ยามลกุล	A11	edit
7	A 1	1234567890130	นาย พัทท	๓ 44	edit
8	กข 1234	1234567890111	นาย กก ข	๓ 44	edit
9	A 15	1234567890120	นาย A BC	A 15	edit
10	H 1234	1234567890000	TestAdd	123	edit
11	H 12	1234567890001	TestAdd2	๓ 44	edit
12	H 123	1234567890002	Test3	A12	edit

รูป 4.12 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์

- 5) เมื่อทำการเลื่อนมาที่ล่างสุดของตาราง Admin สามารถเพิ่มข้อมูลของรถตู้โดยสารสาธารณะใหม่โดยการกดปุ่ม Add ได้ซึ่งอยู่ด้านล่างสุดของตาราง และจะมีหน้าต่างขึ้นมาเพื่อให้ Admin ทำการกรอกข้อมูลของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปพื้นฐานข้อมูลจากนั้นกดปุ่ม Save เพื่อทำการบันทึกข้อมูลนั้นลงไปยังฐานข้อมูล

The screenshot shows the same Admin interface as Figure 4.12, but with a 'Create New Van' modal form open in the center. The form has input fields for 'License plate number', 'Driver id', and 'Driver name', and a 'Route' dropdown menu. A 'Save' button is located at the bottom right of the modal.

รูป 4.13 หน้าต่างเพิ่มข้อมูลรถตู้โดยสารสาธารณะในระบบ

ข้อมูลที่ทำการป้อนเรียบร้อยแล้วดังรูป 4.14 เมื่อทำการกดปุ่ม Save จะถูกเก็บลงในฐานข้อมูลดังรูป 4.15 และนำมาแสดงในหน้าตารางข้อมูลรถตู้โดยสารสาธารณะดังรูป 4.16

Create New Van ✕

License plate number :

Driver id :

Driver name :

Route :

รูป 4.14 ข้อมูลของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปในระบบ

id	license_plate_number	route_id	driver_id	created_at	updated_at
1	Ino 6423	n 55	1234567890123	2016-04-24 15:15:12	2016-04-24 08:15:12
2	nn 1111	A11	1122334455123	2016-03-23 16:01:42	2016-03-23 09:01:42
3	n 1	A12	1234567890122	2016-04-06 16:06:59	2016-01-31 15:23:14
4	n 2	n 44	1234567890126	2016-03-23 20:04:07	2016-03-23 13:04:07
5	A 1234	n 44	1234567890123	2016-03-23 17:23:52	2016-03-23 10:23:52
6	A 111	A11	1234567890121	2016-03-23 09:21:51	2016-01-23 09:21:51
7	A 1	n 44	1234567890130	2016-03-31 22:22:39	2016-03-31 15:22:39
8	nn 1234	n 44	1234567890111	2016-03-23 10:19:06	2016-03-23 10:19:06
9	A 15	A 15	1234567890120	2016-03-30 09:29:29	2016-03-30 09:29:29
10	H 1234	123	1234567890000	2016-03-31 14:26:26	2016-03-31 14:26:26
11	H 12	n 44	1234567890001	2016-03-31 14:27:29	2016-03-31 14:27:29
12	H 123	A12	1234567890002	2016-03-31 14:28:58	2016-03-31 14:28:58
13	A 456	A12	1234567890003	2016-03-31 14:29:34	2016-03-31 14:29:34
14	nn 1100	n 44	11111111111111	2016-04-24 10:14:04	2016-04-24 10:14:04

รูป 4.15 ข้อมูลในฐานข้อมูลตาราง vans

id	license plate	driver id	driver name	route	edit
3	n 1	1234567890122	นาย ชบ	A12	edit
4	n 2	1234567890126	นาย Test Edit	n 44	edit
5	A 1234	1234567890123	Name Surname	n 44	edit
6	A 111	1234567890121	นาย ชื่อ นามสกุล	A11	edit
7	A 1	1234567890130	นาย พกท	n 44	edit
8	nn 1234	1234567890111	นาย กกชช	n 44	edit
9	A 15	1234567890120	นาย A BC	A 15	edit
10	H 1234	1234567890000	TestAdd	123	edit
11	H 12	1234567890001	TestAdd2	n 44	edit
12	H 123	1234567890002	Test3	A12	edit
13	A 456	1234567890003	นาย ก กก	A12	edit
14	nn 1100	11111111111111	Test Add	n 44	edit

รูป 4.16 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการเพิ่มข้อมูล

- 6) ทำการกดปุ่ม Edit ข้อมูลของรถตู้โดยสารสาธารณะที่ต้องการเปลี่ยนแปลงข้อมูล และในหน้าต่างการแก้ไขข้อมูลจะแสดงข้อมูลเดิมของรถตู้โดยสารสาธารณะที่ต้องการเปลี่ยนแปลงข้อมูล จากนั้นทำการกรอกข้อมูลที่ต้องการเปลี่ยนแปลง และ กดปุ่ม Save

ก)

ข)

รูป 4.17 หน้าต่างแก้ไขข้อมูลของรถตู้โดยสารสาธารณะในระบบ

ก) ก่อนการแก้ไขข้อมูล

ข) หลังการแก้ไขข้อมูล

หลังจากกดปุ่ม Save ข้อมูลในตารางจะถูกเปลี่ยนแปลงไปดังรูป 4.19 จากนั้นเข้าไปดูที่หน้าแผนที่ และจะเห็นได้ว่าข้อมูลได้เปลี่ยนแปลงไปตามที่เราแก้ไขข้อมูลดังรูป 4.20

Public Van Monitoring System

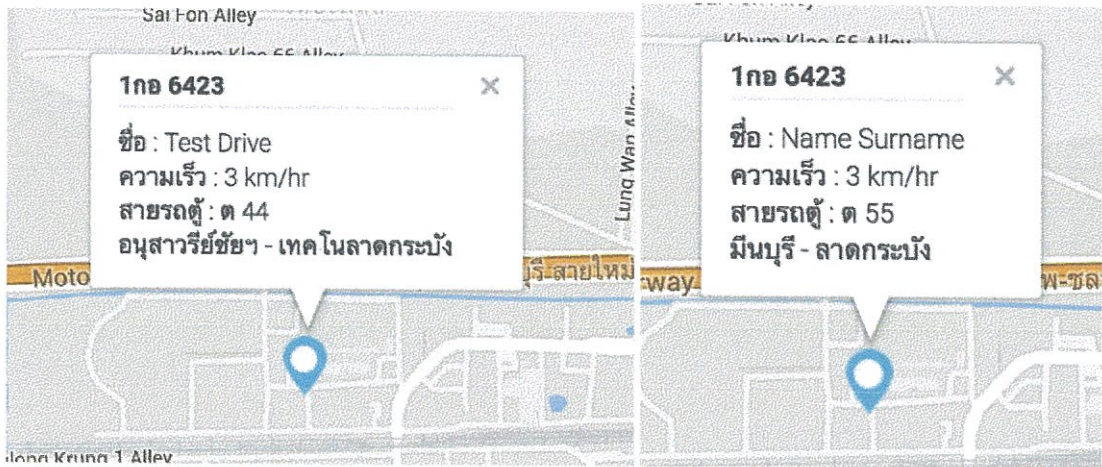
#	License Plate Number	Driver ID	Driver Name	Route Number	
1	กบ 6423	1234567890123	Test Drive	ก 44	edit
2	กบ 1111	1122334455123	นาย ก ข	A11	edit
3	ก 1	1234567890122	นาย ซี ยี	A12	edit
4	ก 2	1234567890126	นาย Test Edit	ก 44	edit
5	A 1234	1234567890123	Test Drive	ก 44	edit
6	A 111	1234567890121	นาย ชื่อ นามสกุล	A11	edit
7	A 1	1234567890130	นาย พัทธ	ก 44	edit
8	กบ 1234	1234567890111	นาย กค ขค	ก 44	edit
9	A 15	1234567890120	นาย A BC	A 15	edit
10	H 1234	1234567890000	TestAdd	123	edit
11	H 12	1234567890001	TestAdd2	ก 44	edit
12	H 123	1234567890002	Test3	A12	edit

รูป 4.18 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์ก่อนการแก้ไขข้อมูล

Public Van Monitoring System

#	License Plate Number	Driver ID	Driver Name	Route Number	
1	กบ 6423	1234567890123	Name Surname	ก 55	edit
2	กบ 1111	1122334455123	นาย ก ข	A11	edit
3	ก 1	1234567890122	นาย ซี ยี	A12	edit
4	ก 2	1234567890126	นาย Test Edit	ก 44	edit
5	A 1234	1234567890123	Name Surname	ก 44	edit
6	A 111	1234567890121	นาย ชื่อ นามสกุล	A11	edit
7	A 1	1234567890130	นาย พัทธ	ก 44	edit
8	กบ 1234	1234567890111	นาย กค ขค	ก 44	edit
9	A 15	1234567890120	นาย A BC	A 15	edit
10	H 1234	1234567890000	TestAdd	123	edit
11	H 12	1234567890001	TestAdd2	ก 44	edit
12	H 123	1234567890002	Test3	A12	edit

รูป 4.19 หน้าตารางข้อมูลรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์หลังการแก้ไขข้อมูล



ก)

ข)

รูป 4.20 หน้าต่างข้อมูลของรถตู้โดยสารสาธารณะ

ก) ก่อนการแก้ไขข้อมูล

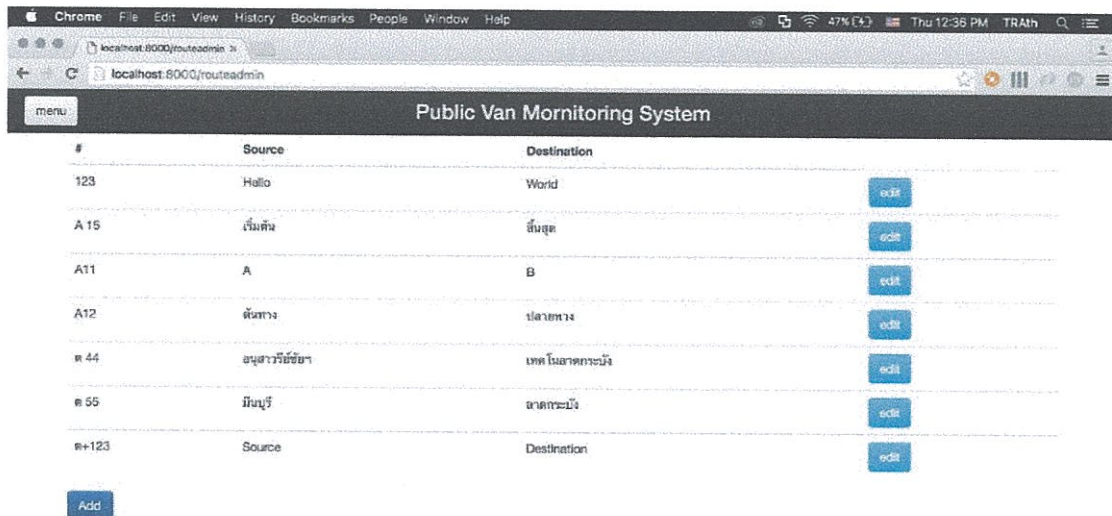
ข) หลังการแก้ไขข้อมูล

- 7) เมื่อกดปุ่ม Menu แล้ว Admin สามารถทำการเปลี่ยนการแก้ไขไปยังข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะ หรือทำการ logout ได้

#	License Plate Number	Driver ID	Driver Name	Route Number
1	1กบ 6423	1234567890123	Test Drive	ต 44
2	กน 1111	1122334455123	นาย ก ข	A11
3	ก 1	1234567890122	นาย ช มี	A12
4	ก 2	1234567890126	นาย Test Edit	ต 44
5	A 1234	1234567890123	Test Drive	ต 44
6	A 111	1234567890121	นาย ชื่อ นามสกุล	A11
7	A 1	1234567890130	นาย พิภพ	ต 44
8	กน 1234	1234567890111	นาย ก ข	ต 44
9	A 15	1234567890120	นาย A BC	A 15
10	H 1234	1234567890000	TestAdd	123
11	H 12	1234567890001	TestAdd2	ต 44
12	H 123	1234567890002	Test3	A12

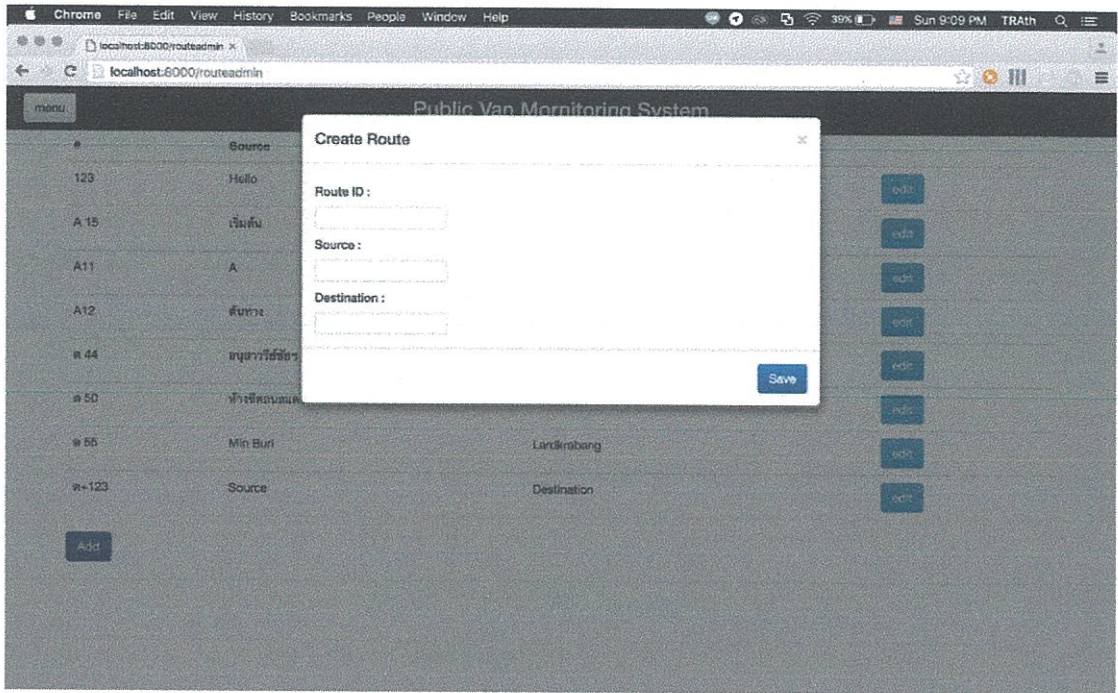
รูป 4.21 หน้าต่าง Tab Menu ของระบบ Admin

เมื่อทำการเลือก Route จาก Tab Menu หรือคลิกที่ Route Number ในตารางข้อมูลรถตู้โดยสารสาธารณะ Admin สามารถทำการแก้ไข หรือเพิ่มข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะได้



รูป 4.22 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์

- 8) ทำการกดปุ่ม Add จะสามารถเพิ่มข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะได้ ซึ่งอยู่ด้านล่างสุดของตาราง และมีหน้าต่างขึ้นมาเพื่อให้ Admin ทำการกรอกข้อมูลของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปในฐานะข้อมูลจากนั้นกดปุ่ม Save เพื่อทำการบันทึกข้อมูลนั้นลงไปยังฐานข้อมูล



รูป 4.23 หน้าต่างเพิ่มข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ

ข้อมูลที่ทำกรป้อนเรียบร้อยแล้วดังรูป 4.24 เมื่อทำการกดปุ่ม Save จะถูกเก็บลงในฐานข้อมูลดังรูป 4.25 และนำมาแสดงในหน้าตารางข้อมูลรถตู้โดยสารสาธารณะดังรูป 4.26

Create Route

✕

Route ID :

Source :

Destination :

รูป 4.24 ข้อมูลของสายการเดินทางของรถตู้โดยสารสาธารณะที่ต้องการเพิ่มลงไปในระบบ

Sequel Pro (MySQL 5.5.42) localforge/van_routes

Search: route_id

route_id	src	des	created_at	updated_at
123	Hello	World	2016-03-23 12:37:09	2016-03-23 12:37:09
A 15	เริ่มต้น	สิ้นสุด	2016-03-31 22:29:35	2016-03-31 15:29:35
A11	A	B	2016-03-31 22:29:24	2016-03-31 15:29:24
A12	ต้นทาง	ปลายทาง	2016-03-23 12:42:36	2016-03-23 12:42:36
ท 44	อนุสาวรีย์ชัย	เขต โขสาคกรณัง	2016-03-22 14:17:52	0000-00-00 00:00:00
ท 50	ห้างชิดชนกเสวร์	พิศนชัย (ซอยจินดา)	2016-04-24 12:03:00	2016-04-24 12:03:00
ท 55	มีนบุรี	ลาดกระบัง	2016-03-31 22:29:02	2016-03-31 15:29:02
ท+123	Source	Destination	2016-04-14 12:36:03	2016-04-14 05:36:05

TABLE INFORMATION

- Created: 3/22/2559 BE
- Engine: InnoDB
- Rows: 8
- Size: 18.0 KB
- Encoding: utf8

8 rows in table

รูป 4.25 ข้อมูลในฐานข้อมูลตาราง van_routes

Chrome local:localhost:8000/routeadmin

Public Van Monitoring System

#	Source	Destination	edit
123	Hello	World	edit
A 15	เริ่มต้น	สิ้นสุด	edit
A11	A	B	edit
A12	ต้นทาง	ปลายทาง	edit
ท 44	อนุสาวรีย์ชัย	เขต โขสาคกรณัง	edit
ท 50	ห้างชิดชนกเสวร์	พิศนชัย (ซอยจินดา)	edit
ท 55	มีนบุรี	ลาดกระบัง	edit
ท+123	Source	Destination	edit

Add

รูป 4.26 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์ หลังการเพิ่มข้อมูล

- 9) ทำการกดปุ่ม Edit ข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะที่ต้องการเปลี่ยนแปลงข้อมูล และในหน้าต่างการแก้ไขข้อมูลจะแสดงข้อมูลสายการเดินทางเดิมของรถตู้โดยสาร

สาธารณะที่ต้องการเปลี่ยนแปลงข้อมูล จากนั้นทำการกรอกข้อมูลที่ต้องการเปลี่ยนแปลง และกดปุ่ม Save

ก)

ข)

รูป 4.27 หน้าต่างแก้ไขข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ

- ก) ก่อนการแก้ไขข้อมูล
- ข) หลังการแก้ไขข้อมูล

หลังจากกดปุ่ม Save ทำให้ข้อมูลในตารางถูกเปลี่ยนแปลงไปดังรูป 4.29 จากนั้นเข้าไปดูที่หน้าแผนที่ และจะเห็นได้ว่าข้อมูลได้เปลี่ยนแปลงไปตามที่แก้ไขข้อมูลดังรูป 4.30

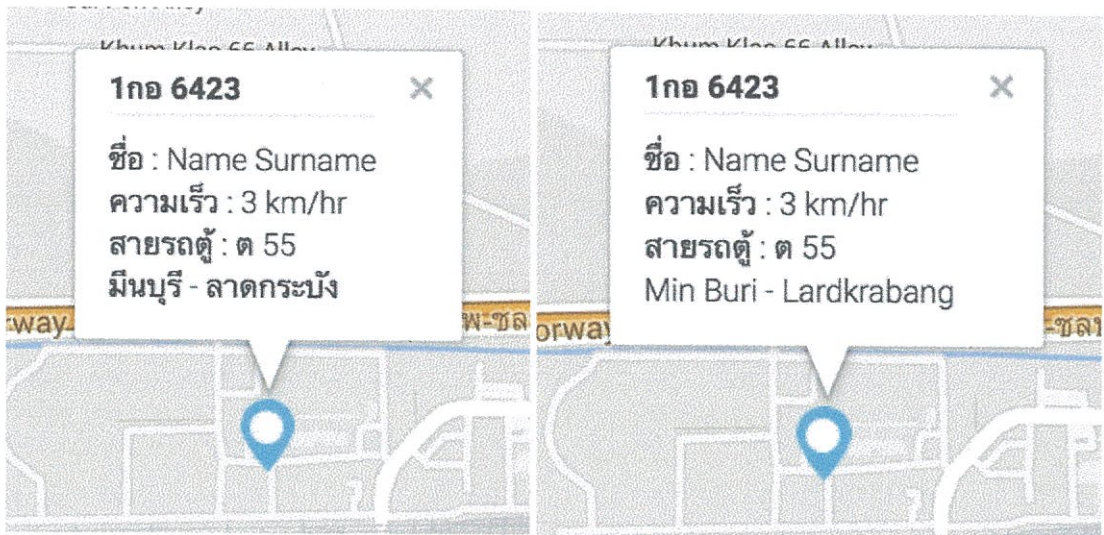
#	Source	Destination	
123	Hello	World	edit
A 15	เชียงใหม่	สิ้นสุด	edit
A11	A	B	edit
A12	ต้นทาง	ปลายทาง	edit
๓ 44	อนุสาวรีย์ชัย	เขตโสภาพรรณัง	edit
๓ 50	ท่าเรือคลองเตย	หัวตะพาน (ซอยจินดา)	edit
๓ 55	มีนบุรี	ลาดกระบัง	edit
๓+123	Source	Destination	edit

รูป 4.28 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์ ก่อนการแก้ไขข้อมูล

#	Source	Destination	
123	Hello	World	edit
A 15	เริ่มต้น	สิ้นสุด	edit
A11	A	B	edit
A12	พื้นที่	ปลายทาง	edit
ค 44	อนุสาวรีย์ชัย	เขต โนนกระแส	edit
ค 50	ท่าเรือคลองเตว	หัวหมาก (ซอยจันทรา)	edit
ค 55	Min Buri	Lardkrabang	edit
ค+123	Source	Destination	edit

Add

รูป 4.29 หน้าตารางข้อมูลสายการเดินทางของรถตู้โดยสารสาธารณะในระบบ Admin ของเว็บไซต์
หลังการแก้ไขข้อมูล



ก)

ข)

รูป 4.30 หน้าต่างข้อมูลของรถตู้โดยสารสาธารณะ

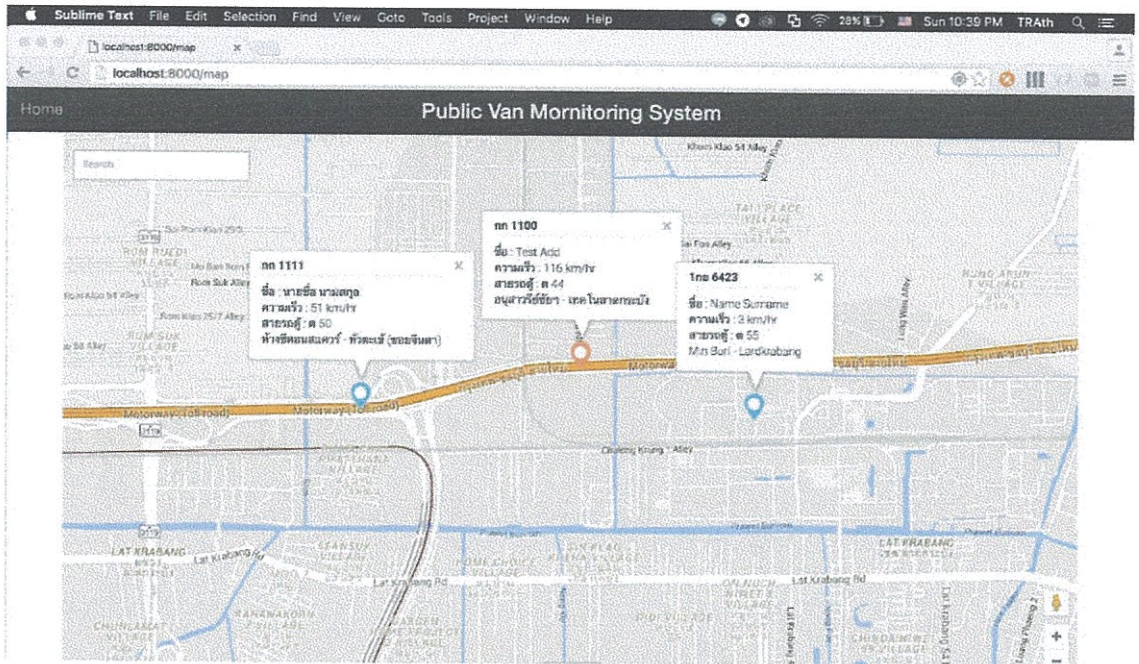
ก) ก่อนการแก้ไขข้อมูลสายการเดินทาง

ข) หลังการแก้ไขข้อมูลสายการเดินทาง

4.3 การค้นหารถตู้โดยสารสาธารณะบนแผนที่ด้วยหมายเลขทะเบียนรถ

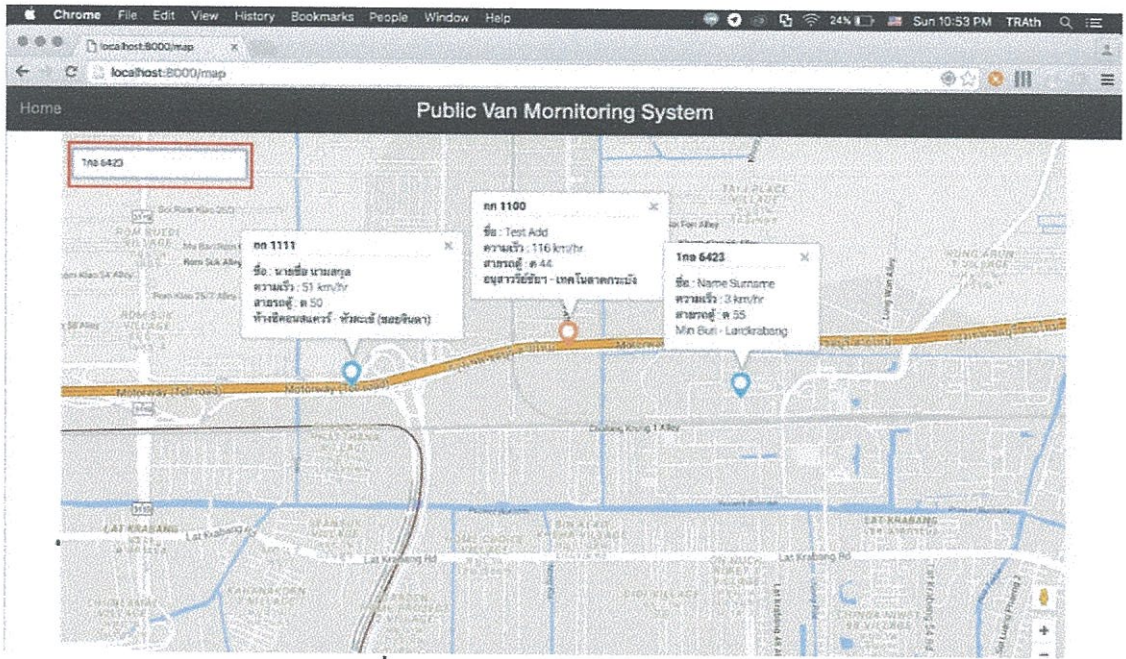
เนื่องจากบนแผนที่จะแสดงข้อมูลของรถตู้โดยสารสาธารณะหลายคัน เราจึงสามารถค้นหารถตู้โดยสารสาธารณะด้วยหมายเลขทะเบียน เพื่อให้แผนที่ทำการแสดงรถตู้โดยสารสาธารณะที่ต้องการดูเพียงคันเดียว

- 1) เข้าไปที่หน้าหลักของเว็บไซต์เพื่อทำการเปิดแผนที่ และดูข้อมูลของรถตู้โดยสารสาธารณะทั้งหมด



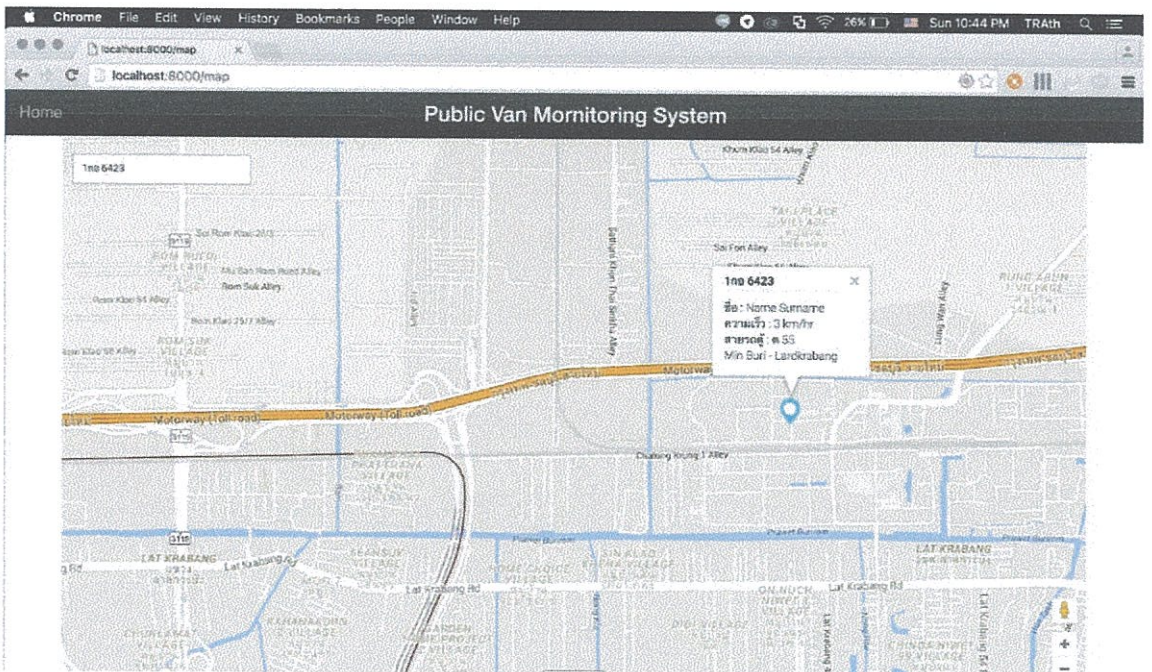
รูป 4.31 หน้าหลักของเว็บไซต์ที่แสดงข้อมูลรถตู้โดยสารสาธารณะทั้งหมด

- 2) ทำการใส่หมายเลขทะเบียนรถของรถตู้โดยสารสาธารณะที่ต้องการค้นหาใน Search Box มุมซ้ายบนของแผนที่จากนั้นทำการกด Enter เพื่อเริ่มค้นหา

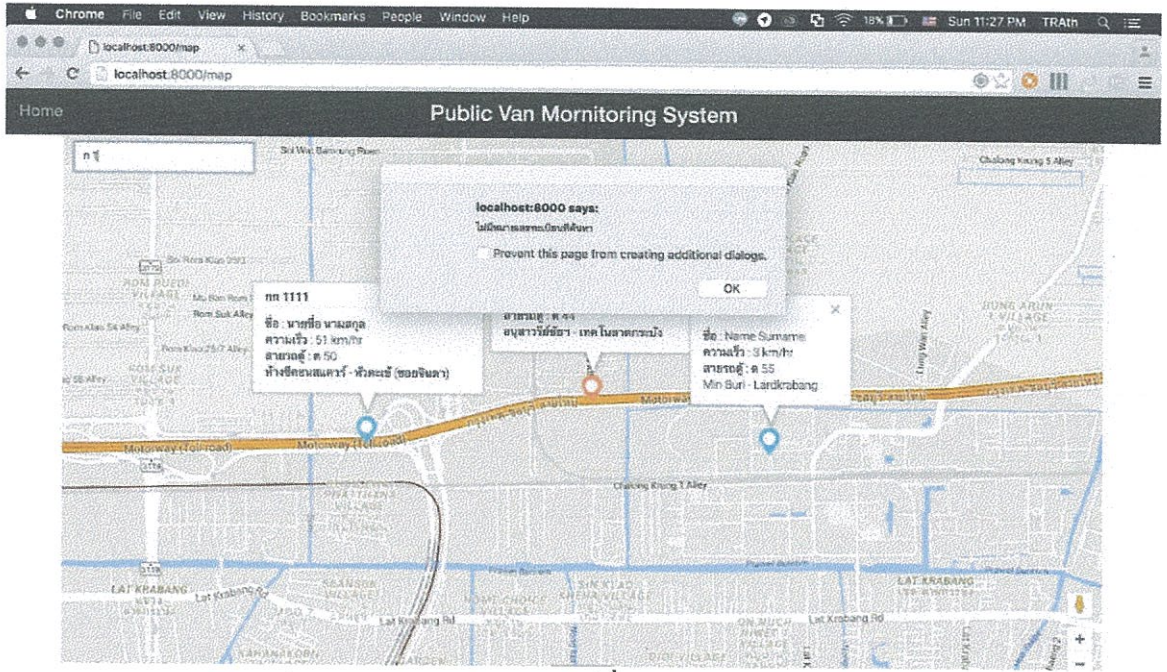


รูป 4.32 หน้าหลักของเว็บไซต์ที่มีการใส่หมายเลขทะเบียนของรถตู้โดยสารสาธารณะที่ต้องการค้นหา

เมื่อทำการค้นหาสำเร็จ ในแผนที่จะเหลือ Marker เพียงอันเดียว คือ รถตู้โดยสารสาธารณะที่มีหมายเลขทะเบียนที่ตรงตามใน Search Box ดังรูป 4.33 แต่ถ้าหากค้นหาไม่สำเร็จจะแสดงหน้าจอแจ้งเตือนขึ้นมาดังรูป 4.34



รูป 4.33 หน้าหลักของเว็บไซต์ที่ทำการค้นหาสำเร็จ



รูป 4.34 หน้าหลักของเว็บไซต์ที่ทำการค้นหาไม่สำเร็จ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงการ

ในโครงการมีการออกแบบโดยแยกเป็น 2 ส่วนหลัก คือ ส่วนอุปกรณ์ฮาร์ดแวร์ และส่วนการให้บริการเว็บเซิร์ฟเวอร์

5.1.1 ส่วนอุปกรณ์ฮาร์ดแวร์

สร้างอุปกรณ์ฮาร์ดแวร์ ที่ประกอบไปด้วย Arduino Board และ SIM908 Module ซึ่งติดต่อกันด้วย AT Command โดยโปรแกรมพัฒนา Arduino Board จะใช้โปรแกรม ภาษา C++ ในการทำงานต่าง ๆ และนำไปติดตั้งบนรถตู้โดยสารสาธารณะ เพื่อทำการสร้าง MQTT Packet จากข้อมูลที่ได้จากตำแหน่ง GPS กับข้อมูลจากการตั้งค่าต่าง ๆ ด้วย AT Command และทำการ Publish MQTT Packet ไปยัง MQTT Broker เพื่อให้มีการประมวลผลข้อมูลนั้นต่อไป ซึ่งจะอยู่ในส่วนการประมวลผล

5.1.2 ส่วนการให้บริการเว็บเซิร์ฟเวอร์

จะแบ่งได้เป็น 2 ส่วนในการประมวลผล คือ

5.1.2.1 MQTT Broker

ทำการติดตั้ง Raspberry PI ด้วย Mosquitto MQTT ซึ่งเป็น Open source เพื่อให้บริการเป็น MQTT Broker โดยจะทำหน้าที่เป็นตัวกลางระหว่างอุปกรณ์ฮาร์ดแวร์ และเว็บเซิร์ฟเวอร์ซึ่งส่วน MQTT Broker นี้จะทำการรับ MQTT Packet ที่ถูก Publish มาจากอุปกรณ์ฮาร์ดแวร์ และทำการ Publish MQTT Packet ต่อไปยัง เซิร์ฟเวอร์ ที่ทำการ Subscribe เพื่อให้ นำข้อมูลไปแสดงผล

5.1.2.2 Apache Server และ MySQL Database

เป็นส่วนการประมวลผลหลักโดยทำการรอ Subscribe MQTT Packet จาก MQTT Broker และนำข้อมูลจาก MQTT Packet มาเก็บไว้ในฐานข้อมูล จากนั้นทำการนำข้อมูลจากฐานข้อมูลมาแสดงบนหน้าเว็บแอปพลิเคชัน โดยส่วนของ Web Server ใช้โปรแกรมภาษา PHP ในการทำงานต่าง ๆ และใช้ภาษา SQL ในส่วนการติดต่อระหว่าง Web Server กับฐานข้อมูล ซึ่งส่วนของ Web Server จะมีการคำนวณความเร็ว และมีการใช้ Logic ต่าง ๆ ในการนำข้อมูลมาแสดงผลให้ผู้ใช้งาน

5.1.2.3 หน้าเว็บไซต์

จะแสดงผลให้ผู้ใช้งานสามารถดูตำแหน่ง และข้อมูลที่สำคัญต่าง ๆ ของรถตู้โดยสารสาธารณะบนแผนที่ออนไลน์ได้ ซึ่งผู้ใช้งานสามารถใช้งานระบบได้ผ่าน Web browser ใน

รูปแบบของเว็บแอปพลิเคชัน โดยใช้งาน Google map API จัดการในการแสดงผลส่วนของแผนที่ ใช้ hmtl,css ในการจัดตกแต่งหน้าแสดงผลให้สวยงาม และมีการควบคุมการทำงานต่าง ๆ ใน Web browser ของผู้ใช้งานด้วย javascript

จากการนำส่วนประกอบต่าง ๆ รวมเข้าด้วยกัน เราสามารถ ดู ติดตามตำแหน่งของรถตู้โดยสารสาธารณะได้ตลอด และยังทราบถึงความเร็วของรถในขณะนั้น หากมีความเร็วเกินที่กฎหมายกำหนดไว้ จะแสดงให้เห็นในทันที ซึ่งทำให้ผู้ขับรถตู้โดยสารสาธารณะต่าง ๆ นั้น มีความระวังตัวมากขึ้น และไม่ยากที่จะขับรถเร็วเกินที่กฎหมายกำหนด

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ส่วนอุปกรณ์ฮาร์ดแวร์

เนื่องจาก library ที่ใช้ในการส่ง MQTT Message ในโครงการนี้คือ weightless MQTT Client มีข้อจำกัดคือ MQTT Message ที่ส่งนั้นห้ามเกิน 127 chars ซึ่งทำให้ไม่สามารถส่งหลายตำแหน่งใน 1 Message ได้ จึงต้องแบ่งออกเป็นหลาย ๆ Message แทน

5.2.2 ส่วนประมวลผล

ในส่วนของการติดต่อ Web Server ที่ทำงานด้วยภาษา PHP ผ่านโปรโตคอล MQTT นั้น มีเอกสารอ้างอิงในการพัฒนาอยู่ จึงทำให้ต้องมีการศึกษาการทำงานของโปรโตคอล MQTT ให้เข้าใจถึงระดับเฟรมเพื่อให้เกิดความเข้าใจในการติดต่อระหว่าง MQTT Broker และ Subscriber ทำให้ง่ายต่อการพัฒนา

5.3 แนวทางการพัฒนาต่อ

5.3.1 ส่วนอุปกรณ์ฮาร์ดแวร์

โครงการชิ้นนี้เป็นการใช้ Hardware สำเร็จรูป ในการสร้างอุปกรณ์ติดตามรถตู้โดยสารสาธารณะ เพื่อเป็นอุปกรณ์ตัวอย่างในการพัฒนาต่อ ซึ่งในการพัฒนาต่อนั้นทำการศึกษาการทำงานของอุปกรณ์ต่าง ๆ และทำการสร้างอุปกรณ์ขึ้นมาเองเพื่อลดต้นทุนการผลิตอุปกรณ์ หรือทำการพัฒนาอุปกรณ์ให้มีความสามารถมากขึ้นเช่น การเก็บภาพการเดินทาง หรือพฤติกรรมการขับรถของคนขับ โดยเพิ่มกล้องให้กับอุปกรณ์

5.3.2 ส่วนประมวลผล

การออกแบบส่วนประมวลผลในส่วนของ Web Server ทำให้ง่ายต่อการพัฒนาต่อไปในอนาคต โดยอาจทำการเพิ่มขีดความสามารถต่าง ๆ ให้ตอบสนองผู้ใช้งานมากขึ้น สามารถรองรับการใช้งานของผู้ใช้งานได้มากขึ้น และยังสามารถนำข้อมูลจากฐานข้อมูลมาใช้ในการคำนวณทางสถิติต่าง ๆ ได้ในส่วนของ MQTT Broker ก็สามารถพัฒนามาใช้ MQTT Broker ที่สร้างเอง และสามารถทำการปรับแต่งการทำงานต่าง ๆ ให้เหมาะสมกับโครงการมากขึ้น

บรรณานุกรม

กรมการขนส่งทางบก. 2555. ข่าวกรมการขนส่งทางบก. [Online].Available :

[http://www.dlt.go.th/th/attachments/plan48-51/3653_Press%20%20%20แถลงข่าวรถตู้โดยสารที่บรรทุกผู้โดยสารเกินกำหนด\[1\].pdf](http://www.dlt.go.th/th/attachments/plan48-51/3653_Press%20%20%20แถลงข่าวรถตู้โดยสารที่บรรทุกผู้โดยสารเกินกำหนด[1].pdf)

กระปุกดอทคอม. รถตู้ชีวแชมป์ รถโดยสารสาธารณะเกิดอุบัติเหตุสูงสุดในรอบ 3 ปี. [Online].

Available : <http://highlight.kapook.com/view/96417>

Mindphp. 2558. **MQTT (Message Queuing Telemetry Transport) คืออะไร.** [Online].Available

: <http://www.mindphp.com/บทความ/31-ความรู้ทั่วไป/3343-mqtt.html>

Eurotech, International Business Machines Corporation (IBM). 2558. **MQTT V3.1 Protocol**

Specification.[Online].Available : <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>

Peter R. Egli. 2556. **MQTT - MQ Telemetry Transport for Message Queuing.**

[Online].Available : <http://www.slideshare.net/PeterREgli/mq-telemetry-transport>

Reha Yurdakul, Ates Yurdakul, Omer Sever. **wMQTT Client for M2M Node (weightless**

MQTT Client).[Online].Available : <http://www.mfnode.com/software/wMQTTClient.pdf>

Stephen Nicolas. 2555. **Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on**

Android. [Online].Available : <http://stephendnicholas.com/archives/1217>

Thaieasyelec. **บทความ Arduino คืออะไร? ตอนที่1 แนะนำเพื่อนใหม่ที่ชื่อ Arduino.** [Online].

Available : <http://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>

Engineering Chiang Mai University. **GPRS (General packet radio service).** [Online].Available :

http://www.ee.eng.cmu.ac.th/~tharadol/teach/ee341/341_47/341_47rp/GPRS.pdf

URAIWAN103. 2556. ระบบระบุตำแหน่งบนพื้นโลก (Global Positioning System: GPS).

[Online].Available : <https://yingpew103.wordpress.com/>

Micro Electronics Trade Ltd. and SIMCom. **SIM908**. [Online].Available :

<http://www.simcom.ee/modules/gsm-gprs-gps/sim908/>

Syeda Anila Nusrat. 2553. **Introduction to AT commands and its uses**. [Online].Available :

<http://www.codeproject.com/Articles/85636/Introduction-to-AT-commands-and-its-uses>

นวัตกรรม. 2556. **AT**. [Online].Available :

<http://www.nawattakam.com/talk/index.php?topic=1512.0>

Mr.DP. 2552. **Apache คืออะไร**. [Online].Available : [http://we-](http://we-developer.blogspot.com/2009/09/blog-post.html)

[developer.blogspot.com/2009/09/blog-post.html](http://we-developer.blogspot.com/2009/09/blog-post.html)

The Apache Software Foundation. **Apache**. [Online].Available : <https://httpd.apache.org/>

ธีรภัทร มนต์รีศาสตร์. **Apache เว็บเซิร์ฟเวอร์สารพัดประโยชน์**. [Online]. Available :

<http://www.itdestination.com/articles/idc-apache/>

วิกิพีเดีย. 2558. **ภาษาพีเอชพี**. [Online].Available : <https://th.wikipedia.org/wiki/ภาษาพีเอชพี>

ชมรม MySQL ในประเทศไทย. **MySQL**. [Online].Available : <http://www.mysql-thailand.com>

วิกิพีเดีย. 2557. **มายเอสคิวเอล**. [Online].Available : <https://th.wikipedia.org/wiki/มายเอสคิวเอล>

Oracle Corporation.**MySQL**. [Online].Available : <https://www.mysql.com/>

Google. **Google Map API**. [Online].Available :

<https://www.google.co.th/intx/th/work/mapsearch/products/mapsapi.html>

Swiftlet. **GOOGLE MAP API คืออะไร?**. [Online].Available : <https://swiftlet.co.th/google-api->

[คืออะไร/](https://swiftlet.co.th/google-api-)

Google. **Google Maps JavaScript API**. [Online]. Available :

<https://developers.google.com/maps/documentation/javascript/>

มนัสวิน แสนคำ. 2558. **REST (Representational State Transfer) คืออะไร?**. [Online]. Available :

<http://www.siamhttp.com/site/article/restful-web-service.html>

TMI. 2551. **AJAX (Asynchronous JavaScript and XML)**. [Online]. Available :

http://www.tmi.or.th/index.php?option=com_content&view=article&id=281&catid=30:programmer-corner&Itemid=40

วิกิพีเดีย. 2558. **เอแจ็กซ์**. [Online]. Available : <https://th.wikipedia.org/wiki/เอแจ็กซ์>

Laravel. **Laravel 5.1**. [Online]. Available : <https://laravel.com/docs/5.1>

Laracasts. **Laravel Community**. [Online]. Available : <https://laracasts.com/>

Bundit Nuntates. **LARAVEL ตอนที่ 1 : รู้จัก LARAVEL**. [Online]. Available :

<http://gunoob.com/laravel-ตอนที่-1-รู้จัก-laravel/>

Kamol Chalermviriya. **Laravel - PHP Web Development Framework**. [Online]. Available :

<http://kamolcu.blogspot.com/2014/08/laravel-php-web-development-framework.html>

Thaieasyelec. **Internet of Things**. [Online]. Available :

<http://www.thaieasyelec.com/products/internet-of-things.html>

Ridge Solutions Engineered Software. **Algorithm to calculate speed from two GPS latitude and longitude points and time difference**. [Online]. Available :

<http://www.ridgesolutions.ie/index.php/2013/11/14/algorithm-to-calculate-speed-from-two-gps-latitude-and-longitude-points-and-time-difference/>