แอปพลิเคชั่น บอกเวลาเข้าสถานีของรถไฟ สำหรับแอนดรอยด์
สำหรับบอกเวลาเข้าชานชาลาของรถไฟ
THE APPLICATION INDICATED THE TRAIN ARRIVE TO STATION VIA ANDROID

โดย

นางสาวจิราวรรณ วงศ์ก่ำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

แอปพลิเคชั่น บอกเวลาเข้าสถานีของรถไฟ สำหรับแอนดรอยด์
สำหรับบอกเวลาเข้าชานชาลาของรถไฟ
THE APPLICATION INDICATED THE TRAIN ARRIVE TO STATION VIA ANDROID

โดย

นางสาวจิราวรรณ วงศ์ก่ำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

แอปพลิเคชั่น บอกเวลาเข้าสถานีของรถไฟ สำหรับแอนดรอยด์
สำหรับบอกเวลาเข้าชานชาลาของรถไฟ
THE APPLICATION INDICATED THE TRAIN ARRIVE TO STATION VIA ANDROID

โดย

นางสาวจิราวรรณ วงศ์ก่ำ 55010184

อาจารย์ที่ปรึกษา

รศ.ดร. ไกรสิน ส่งวัฒนา

อาจารย์ที่ปรึกษาร่วม

ผศ.ดร. พิเชฐ ม่วงนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

...../...../.....

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

...../...../.....

ปริญญานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอปพลิเคชั่น บอกเวลาเข้าสถานีของรถไฟ สำหรับแอนดรอยด์

สำหรับบอกเวลาเข้าชานชาลาของรถไฟ

THE APPLICATION INDICATED THE TRAIN ARRIVE TO STATION VIA ANDROID

ผู้จัดทำ

นางสาวจิราวรรณ วงศ์กำ 55010184

..... อาจารย์ที่ปรึกษา

(รศ.ดร. ไกรสิน ส่งวัฒนา)

..... อาจารย์ที่ปรึกษาร่วม

(ผศ.ดร. พิเชฐ ม่วงนวล)

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้รับความอนุเคราะห์จากอาจารย์ที่ปรึกษา และอาจารย์พิเชฐ ม่วงนวล ที่ปรึกษาร่วมซึ่งคอยให้คำแนะนำในการทำงาน ให้ความรู้ที่เป็นประโยชน์ต่อปริญญานิพนธ์ ฉบับนี้เป็นอย่างมากพร้อมทั้งได้ว่ากล่าวตักเตือนสั่งสอนเพื่อให้ปริญญานิพนธ์แล้วเสร็จตามเป้าหมาย ที่ตั้งไว้ และนอกจากนั้นยังมีกลุ่มเพื่อนที่คอยให้คำแนะนำและที่สำคัญที่สุดคือกำลังใจเพราะได้ความช่วยเหลือจากเพื่อนถึงได้ผ่านพ้นช่วงเวลาเหล่านั้นมาได้ และสุดท้ายนี้ขอขอบคุณครอบครัวที่สนับสนุนทางด้านการเงินในการซื้ออุปกรณ์ที่ใช้เพิ่มเติมจนสำเร็จไปได้ด้วยดี

นางสาวจิราวรรณ วงศ์ก่ำ

ผู้จัดทำ

แอปพลิเคชั่น บอกเวลาเข้าสถานีของรถไฟ สำหรับแอนดรอยด์
สำหรับบอกเวลาเข้าชานชาลาของรถไฟ
THE APPLICATION INDICATED THE TRAIN ARRIVE
TO STATION VIA ANDROID

โดย นางสาวจิราวรรณ วงศ์กำ 55010184

อาจารย์ที่ปรึกษา รศ.ดร. ไกรสิน ส่งวัฒนา
อาจารย์ที่ปรึกษาร่วม ผศ.ดร. พิเชฐ ม่วงนวล

บทคัดย่อ

โครงงานนี้นำเสนอระบบบอกเวลาการเข้าสถานีของรถไฟแสดงผลด้วยแอปพลิเคชันแอนดรอยด์ โดยระบบจะแบ่งเป็น 2 ส่วนคือ ระบบที่ติดตั้งบนรถไฟซึ่งจะประกอบด้วยไมโครคอนโทรลเลอร์ GPS และ GPRS ติดตั้งไว้ที่ขบวนรถไฟต้นแบบ และ ระบบเซิร์ฟเวอร์ที่จะรับข้อมูลพิกัดขบวนรถผ่านทางอินเตอร์เน็ต ในทุกๆ ช่วงเวลาจริง และ เก็บไว้ในฐานข้อมูล โดยผู้ใช้งานจะต้องมีแอปพลิเคชันที่พัฒนาขึ้นติดตั้งบนโทรศัพท์แอนดรอยด์ เพื่อดึงข้อมูลจากเซิร์ฟเวอร์มาแสดงพิกัดพร้อมคำนวณเวลาที่จะมาถึงของขบวนรถไฟ เพื่ออำนวยความสะดวกให้กับผู้ใช้บริการ และ สามารถนำข้อมูลการเดินรถเพื่อปรับปรุงการให้บริการให้ตรงต่อเวลาในอนาคตได้

ABSTRACT

This project presents a train arrival system based on Android application. This system is divided in two parts, first part is equipment located in the prototype train which consists contains a microcontroller, GPS and GPRS modules and the second part is server for receiving the train coordinate in real time and storing their information in database. The users have to use the android application for connecting the database and process their information for calculating the train time arrival approximation. The users can be comfortably predicted the time for using the trains. Moreover, the information of trains can also be used for maintain and improve the train services in the future.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 แอนดรอยด์ (Android)	3
	2.2 จีพีเอส (GPS)	6
	2.3 แอร์การ์ด (Air Card)	11
	2.4 ภาษาซีชาร์ป	12
	2.5 Visual Studio	14
	2.6 ระบบฐานข้อมูล Microsoft SQL	18
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**19**
	3.1 การออกแบบ	19
	3.2 เครื่องมือที่ใช้ในการทดลอง	23
	3.3 การจัดเก็บผลการทดลอง	23
บทที่ 4	**ผลการทดลอง**	**25**
	4.1 ผลการทดลอง	25

สารบัญ (ต่อ)

	หน้า
บทที่ 5 ผลการทดลอง	**28**
5.1 สรุปผล	28
5.2 ข้อเสนอแนะ	28
บรรณานุกรม	**29**
ภาคผนวก ก CODE	30
ภาคผนวก ข DATASHEET	43

สารบัญรูป

รูปที่		หน้า
1.1	บล็อคไดอะแกรมการทำงานรวม	2
2.1	โครงสร้างของระบบปฏิบัติการแอนดรอยด์	4
2.2	ระบบพิกัดกริดแบบยูทีเอ็ม	6
2.3	แอร์การ์ด	11
2.4	โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน	13
2.5	โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace	14
2.6	Visual Studio	15
2.7	หน้าจอแสดงองค์ประกอบหลักของโปรแกรม	15
2.8	หน้าต่าง Solution Explorer	17
2.9	หน้าต่าง Properties Window	17
2.10	หน้าต่าง Code Editor	18
3.1	บล็อคไดอะแกรมการทำงานรวม	19
3.2	ส่วนการจัดเก็บข้อมูล	20
3.3	ส่วนประมวลผลข้อมูล	21
3.4	ส่วนแสดงข้อมูล	22
4.1	การเลือกสถานีในการเดินทาง	25
4.2	การเลือกรับการแจ้งเตือน	26
4.3	การแจ้งเตือนก่อนรถไฟมาถึง	26
4.4	การตรวจสอบรอบรถไฟในทุกขบวน	27

สารบัญตาราง

หน้า

ตารางที่

3.1 ค่าเวลาเฉลี่ยการเข้าชานชาลาของรถไฟ 24

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

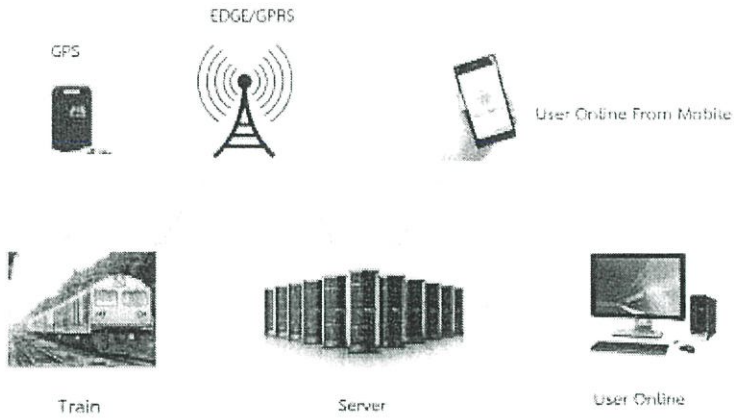
เนื่องจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังนั้นมีเส้นทางรถไฟสายตะวันออก จากสถานีต้นทางกรุงเทพหัวลำโพง ไปยังปลายทางที่สถานีชุมทางฉะเชิงเทราตัดผ่านทางสถาบัน ซึ่งเป็นอีกทางเลือกหนึ่งของการเดินทางของนักศึกษาและบุคคลภายนอก แต่สืบเนื่องจากขบวนรถไฟนั้นเดินทางมาถึงสถานีไม่ตรงตามเวลาที่ระบุไว้ จึงทำให้การวางแผนการเดินทางโดยรถไฟนั้นมีความเป็นไปได้ยาก ดังนั้นทางผู้จัดทำจึงจัดทำแอปพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ขึ้นเพื่อคำนวณและแสดงเวลาที่รถไฟจะเข้าสู่สถานีที่ผู้ใช้งานต้องการทราบ เพื่อสร้างความสะดวกต่อการเดินทางโดยรถไฟมากขึ้นและยังทำให้แผนการเดินทางโดยรถไฟเป็นไปด้วยความสะดวกสบายมากขึ้น

1.2 วัตถุประสงค์

1) เพื่อแสดงเวลาที่รถไฟจะเข้าสู่ชานชาลาแก่ผู้โดยสาร
2) เพื่อความสะดวกในการวางแผนการเดินทาง

1.3 ขอบเขตของปริญญานิพนธ์

เพื่อสร้างแอปพลิเคชั่นในโทรศัพท์โดยระบบปฏิบัติการแอนดรอยด์ ที่จะใช้บอกสถานะของรถไฟว่า ขณะนี้รถไฟอยู่ห่างจากสถานีที่ผู้โดยสารอยู่ จะต้องใช้เวลานานเท่าใดจึงจะเดินทางมาถึง ซึ่งแอปพลิเคชั่นนี้เป็นตัวอย่างของการทดลองที่ใช้กับขบวนรถไฟสายตะวันออก คือจากสถานีต้นทางกรุงเทพหัวลำโพง และสถานีปลายทางที่สถานีชุมทางฉะเชิงเทราเพียงขบวนเดียวเท่านั้น โดยยังไม่รวมสถานีอื่นๆ นอกเหนือจากนี้ และได้กำหนดให้รับข้อมูลมาจากจีพีเอส ที่ติดอยู่กับขบวนรถไฟ



รูปที่ 1.1 บล็อคไดอะแกรมการทำงานรวม

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 แอนดรอยด์ (Android)

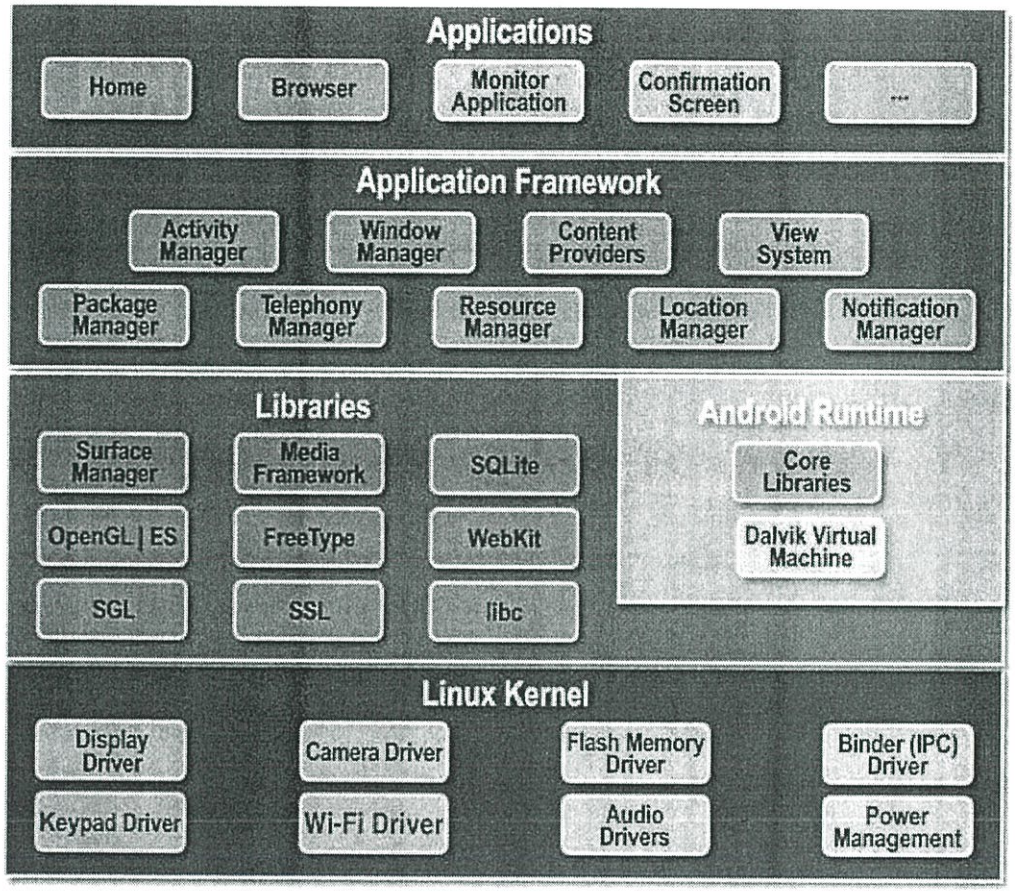
แอนดรอยด์ เป็นระบบปฏิบัติการที่มีพื้นฐานอยู่บนลินุกซ์ (Linux) ในอดีตถูกออกแบบมาสำหรับอุปกรณ์ที่ใช้จอสัมผัส เช่น สมาร์ทโฟนและแท็บเล็ตคอมพิวเตอร์ ปัจจุบันได้แพร่ไปยังอุปกรณ์หลายชนิด เช่น กล้องดิจิตอล, สมาร์ททีวี (Smart TV), ระบบแอนดรอยด์ เป็นต้น

แอนดรอยด์เป็นระบบปฏิบัติการโอเพนซอร์ซ และกูเกิลได้เผยแพร่ภายใต้ลิขสิทธิ์ที่อนุญาตให้ผู้ผลิตปรับแต่งและวางจำหน่ายได้ รวมไปถึงนักพัฒนาและผู้ให้บริการเครือข่ายด้วย อีกทั้งแอนดรอยด์ยังเป็นระบบปฏิบัติการที่รวมนักพัฒนาที่เขียนโปรแกรมประยุกต์มากมาย ภายใต้ภาษาจาวา (Java Language)

ตั้งแต่ปี พ.ศ. 2551 แอนดรอยด์ได้ใช้การอัปเดตแบบเรียงตามเลขรุ่น ซึ่งจะมีการปรับปรุงส่วนต่างๆ ของระบบปฏิบัติการ เพิ่มคุณสมบัติใหม่ และ แก้ไขข้อผิดพลาดในรุ่นก่อนหน้า โดยแต่ละรุ่นจะมีชื่อเฉพาะเรียงตามลำดับตัวอักษรและจะใช้ชื่อจากขนมหวาน เช่น รุ่น 1.5 "คัพเค้ก" รุ่น 1.6 "โดนัท" รุ่น 4.3 "เจลลีบีน" และรุ่น 4.4 "คิทแคท" ซึ่งได้เปิดตัวเมื่อวันที่ 3 กันยายน พ.ศ. 2556 ในปี พ.ศ. 2557 กูเกิลเปิดตัว "Android L" (ต่อมาใช้ชื่อว่าโลลิป็อป) และในปีพ.ศ. 2558 กูเกิลได้เปิดตัวแอนดรอยด์รุ่นใหม่ล่าสุดในชื่อ "Android M" (ต่อมาใช้ชื่อว่ามาร์ชเมลโลว)

2.1.1 โครงสร้างของแอนดรอยด์

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ ถือว่าเป็นสิ่งสำคัญ โดยหากนักพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบทั้งหมด จะสามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนาให้เกิดประสิทธิภาพในการทำงาน โดยโครงสร้างของแอนดรอยด์สามารถสรุปได้ตามรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ มีการแบ่งออกมา เป็นส่วนๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดจะเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งก็คือส่วนของแอปพลิเคชั่น (Applications) จากนั้นก็จะลำดับลงมาเป็นองค์ประกอบอื่นๆตามลำดับ และสุดท้ายจะเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel โครงสร้างของแอนดรอยด์ สามารถอธิบายเป็นส่วนๆ ได้ดังนี้

2.1.1.1 Applications หรือส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ดโปรแกรมเอาไว้

2.1.1.2 Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น

1) Activity Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม

2) Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้

3) View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)

4) Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่นหมายเลขโทรศัพท์ เป็นต้น

5) Resource Manager เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็นข้อความ, รูปภาพ

6) Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์

7) Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรม ต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ (Status Bar) ของหน้าจอ

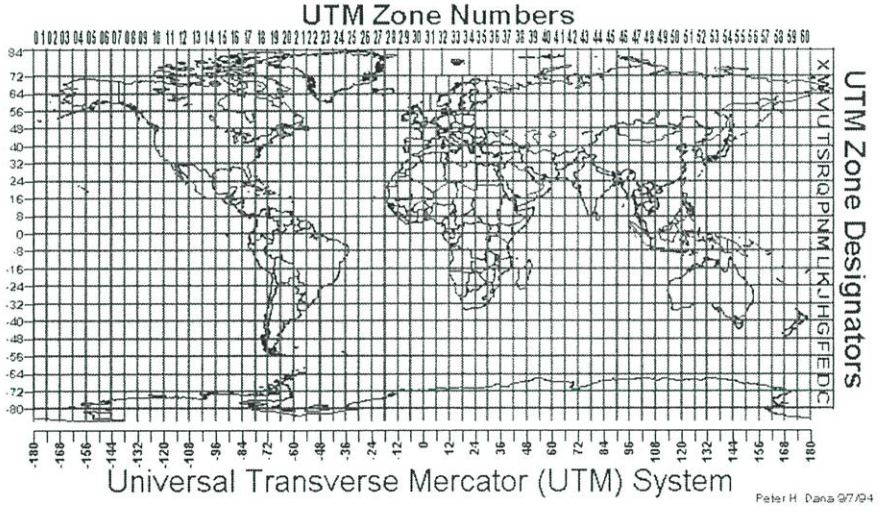
8) Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น Surface Manager จัดการเกี่ยวกับการแสดงผล, Media Framework จัดการเกี่ยวกับการแสดงภาพและเสียง, Open GL ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ, เอสคิวแอลไลท์ (SQLite) จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

Android Runtime จะมี Dalvik Virtual Machine ที่ถูกออกแบบมา เพื่อให้ทำงานบนอุปกรณ์ที่มี หน่วยความจำ (Memory), หน่วยประมวลผลกลาง (CPU) และแบตเตอรี่ (Battery) ที่จำกัด ซึ่งการทำงานของ Dalvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงาน ไปเป็นไฟล์ .DEX ก่อนการทำงาน เหตุผลก็เพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับ หน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาคือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)

Linux Kernel เป็นส่วนที่ทำหน้าที่เป็นหัวใจสำคัญในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ, พลังงาน, ติดต่อกับอุปกรณ์ต่างๆ, ความปลอดภัยเครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์รุ่น 2.6 (Linux 2.6 Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.2 จีพีเอส (GPS)

GPS (Global Positioning System) คือ ระบบกำหนดตำแหน่งบนโลก หมายถึง ระบบบอกตำแหน่งบนพื้นผิวโลกโดยอาศัยการคำนวณระบบพิกัดกริดแบบยูทีเอ็ม (Universal Transverse Mercator: UTM) ทั้งหมด 60 โซนจากนาฬิกาที่ส่งมาจากดาวเทียมที่โคจรอยู่รอบโลก ซึ่งมีตำแหน่งที่แน่นอน ระบบนี้สามารถบอกตำแหน่ง ณ จุดที่สามารถรับสัญญาณได้ทั่วโลก โดยเครื่องรับสัญญาณ จีพีเอส รุ่นใหม่ๆ จะสามารถคำนวณความเร็วและทิศทาง โดยสามารถนำมาใช้ร่วมกับโปรแกรมแผนที่เพื่อใช้ในการนำทางได้ ดังรูปที่ 2.2



รูปที่ 2.2 ระบบพิกัดกริดแบบยูทีเอ็ม

ระบบบอกพิกัดดาวเทียมอื่นๆ ที่คล้ายคลึงกับระบบจีพีเอส ในปัจจุบันมีหลายระบบ ดังนี้

1) GLONASS (Global Navigation Satellite System) เป็นระบบของรัสเซีย ที่พัฒนาเพื่อแข่งขันกับสหรัฐอเมริกา แต่ระบบนี้ยังใช้งานได้ไม่สมบูรณ์ ใช้งานได้เฉพาะในรัสเซีย ยุโรป และแคนาดา

2) Galileo เป็นระบบที่พัฒนาโดยสหภาพยุโรปร่วมกับจีน อิสราเอล อินเดีย โมร็อกโก ซาอุดิอาระเบีย เกาหลีใต้ และยูเครน แล้วเสร็จในปีพ.ศ. 2553

3) Beidou เป็นระบบที่กำลังพัฒนาโดยประเทศจีน โดยให้บริการเฉพาะบางพื้นที่แต่ในอนาคตมีแผนที่จะพัฒนาโดยให้ครอบคลุมทั่วโลกโดยจะใช้ชื่อว่า COMPASS

4) QZSS ระบบดาวเทียมของญี่ปุ่น ทำหน้าที่หลากหลาย ช่วยเสริมการหาตำแหน่งด้วยจีพีเอส โดยเน้นพื้นที่ประเทศญี่ปุ่น ที่มีอาคารสูงบดบังสัญญาณจีพีเอส สำหรับ QZSS ถูกออกแบบให้มีวงโคจรเป็นเลข 8 โดยระบบจะประกอบด้วยดาวเทียม 3-4 ดวง

2.2.1 การทำงานของจีพีเอส

ดาวเทียมทุกดวงที่โคจรอยู่บนท้องฟ้าจะมีการส่งสัญญาณจีพีเอส มาที่พื้นดิน เพื่อที่ จะบอกพิกัดตัวเองต่อสถานีควบคุม ข้อจำกัดคือดาวเทียมไม่สามารถใช้พลังงานสูงมากนัก เนื่องจากพลังงานทั้งหมดได้จากพลังงานแสงอาทิตย์ ดังนั้นกำลังส่งของสัญญาณจีพีเอส จึงมีขนาดต่ำ ส่งผลให้

1) เครื่องรับจีพีเอส ต้องมีความไวสูงมากๆ เพื่อให้สามารถที่จะรับสัญญาณจีพีเอส จากดาวเทียมซึ่งโคจรอยู่และรับสัญญาณจีพีเอส ที่ภาคพื้นดินซึ่งมีขนาดสัญญาณที่เล็กมากๆ

2) จากผลของข้อที่1 ทำให้เครื่องรับจีพีเอส จำเป็นต้องใช้พลังงานไฟฟ้าในส่วนภาครับ เป็นจำนวนมากโดยส่วนนี้จะทำหน้าที่ ทั้งกรองสัญญาณที่รบกวนออกจากสัญญาณจีพีเอส แล้วทำการขยายสัญญาณและนำไปถอดรหัสที่ส่วนของดิจิตอล โดยทั่วไปเครื่องรับจึงมีอุปกรณ์จำนวนมาก จึงเรียกว่าชิปเซต (Chipset)

3) จากข้อที่ 2 ภาคที่เป็นระบบดิจิตอลจะมีไมโครคอนโทรลเลอร์เป็นหัวใจสำคัญในการคำนวณและถอดรหัสต่างๆ โดยปกติจะเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถในการคำนวณอย่างมาก ดังนั้นจึงนิยมใช้ไมโครคอนโทรลเลอร์แบบ 32 บิต

4) หากสัญญาณจีพีเอส ที่เครื่องรับและเครื่องส่งไม่ตรงกัน (Synchronize) งานนี้จะไม่สามารถถอดรหัสได้เลย ดังนั้นหากต้องการให้จีพีเอสบอกพิกัดได้ ต้องห้ามปิดเครื่องจำเป็นต้องเปิดเครื่องตลอดเวลา

5) เนื่องจากสัญญาณจีพีเอส ที่ตกกระทบภาคพื้นดินมีขนาดเล็กมาก ดังนั้นจึงง่ายต่อการถูกรบกวน การที่จะให้สัญญาณที่แรงและดีที่สุด คืออย่าให้มีสิ่งกีดขวาง เครื่องรับจีพีเอส บางประเภทจึงมีสายอากาศแบบภายนอกโดยติดตั้งในที่โล่งแจ้ง

2.2.2 ประเภทของจีพีเอส

จีพีเอสมีใช้ในหลายวัตถุประสงค์ทางธุรกิจแต่ที่ใช้กันอย่างแพร่หลายมี 2 ประเภท คือ อุปกรณ์และระบบนำทาง (GPS Navigator) และ อุปกรณ์และระบบติดตามรถ ยานพาหนะหรือสัตว์เลี้ยง (GPS Tracking System) โดยมีการทำงานที่แตกต่างกันดังนี้

2.2.2.1 อุปกรณ์และระบบนำทาง (GPS NAVIGATOR)

เป็นอุปกรณ์ที่ใช้เพื่อบอกตำแหน่งที่จะเดินทางไป ใช้การบอกตำแหน่ง รถของเราร่วมกับแผนที่ในการเดินทาง ผู้ใช้งานโดยส่วนมากเป็นเจ้าของรถที่ต้องการเดินทางไปในที่ต่างๆ ที่ไม่คุ้นเคย เมื่อสิ้นสุดการเดินทางก็บรรลุวัตถุประสงค์แผนที่ที่ใช้ในระบบนำทางในรถยนต์ที่ใช้กันโดยทั่วไปที่เป็นมาตรฐานอย่างไม่เป็นทางการ (de facto standard) มาจากสองบริษัท ได้แก่ แผนที่จากบริษัทแนฟเทค (NavTeq) และจากบริษัทเทเลแอตลาส (Tele Atlas) นอกจากสองบริษัทนี้แล้วยังมีแผนที่จากบริษัทอื่นๆ อีกแต่ไม่เป็นที่นิยม เนื่องจากมีข้อจำกัดจากฟอร์แมตของแผนที่ที่เป็นลักษณะเฉพาะของแต่ละบริษัท และบริษัทขนาดเล็กไม่สามารถสนับสนุนพื้นที่ครอบคลุมประเทศในทวีปต่างๆ ทำให้บริษัทขนาดใหญ่มีข้อได้เปรียบ ตลอดทั้งขั้นตอนที่ทำให้การประมวลผลแผนที่ ที่จะใช้กับซอฟต์แวร์ระบบนำทางมีปัญหาในการทำแผนที่ เพื่อให้ใช้กับซอฟต์แวร์นั้นๆ ดังนั้นบริษัททำระบบนำร่องส่วนใหญ่จะใช้แผนที่ของบริษัทใดบริษัทหนึ่ง หรือสองบริษัทนี้ เพื่อความรวดเร็วในการประมวลผล เนื่องจากข้อมูลของแผนที่แต่ละประเทศมีขนาดข้อมูลมหาศาลและใช้เนื้อที่ในการเก็บขนาดใหญ่ ทำให้ไม่สามารถนำข้อมูลทุกอย่างมาใช้ในซอฟต์แวร์ของระบบนำร่องได้จึงได้มีการนำข้อมูลแผนที่นั้นมาทำการจัดเรียงใหม่เพื่อความเหมาะสม เพื่อประโยชน์ในเรื่องขนาดของแผนที่ ที่ต้องนำไปใช้ ตลอดจนความรวดเร็วในการเข้าอ่านและประมวลผลข้อมูล

2.2.2.2 อุปกรณ์และระบบติดตามรถ ยานพาหนะหรือสัตว์เลี้ยง

เป็นอุปกรณ์ที่ใช้เพื่อติดตามรถโดยจะเก็บตำแหน่งการเดินทางตลอดเวลา เพื่อนำข้อมูลมาใช้ประโยชน์ในการบริหารจัดการยานพาหนะในเชิงการค้าพาณิชย์ รวมทั้งสิ่งของที่อยู่ในยานพาหนะผู้ใช้งานจะเป็นเจ้าของธุรกิจที่มีรถใช้ในภารกิจต่างๆ โดยจะใช้เป็นเครื่องมือในการบริหารจัดการการขนส่งในกลุ่มธุรกิจขนส่งสินค้า ธุรกิจบริการ ธุรกิจขนส่งผู้โดยสาร และรถสาธารณะต่างๆ โดยมากจะใช้ร่วมกับซอฟแวร์การรายงานประมวลและวิเคราะห์หาความคุ้มค่าสูงสุดในการใช้รถ รวมถึงการป้องกัน การสูญเสียจากการขนส่งในทุกขั้นตอน มีการส่งข้อมูลภาพและข้อมูลอื่นผ่านทางหน้าจอคอมพิวเตอร์ตลอด 24 ชั่วโมง

อุปกรณ์ติดตามรถ (GPS Tracking) มีอยู่หลากหลายแบบ แต่สามารถแบ่งตามประเภทการใช้งานได้เป็น 3 ประเภท คือ ออฟไลน์ กึ่งออฟไลน์ และ ออนไลน์

1) อุปกรณ์ติดตามรถแบบออฟไลน์

แบบนี้จะเก็บข้อมูลตำแหน่งพิกัดเอาไว้ในตัวอุปกรณ์ติดตามรถเมื่อต้องการดูข้อมูล จะต้องนำอุปกรณ์ติดตามรถมาต่อเข้ากับคอมพิวเตอร์ เพื่อถ่ายข้อมูลลงไปโดยมีข้อดี คือ อุปกรณ์แบบนี้ราคาไม่แพง และมีข้อเสียคือไม่สามารถตรวจสอบตำแหน่งปัจจุบันได้ ตรวจสอบได้แต่ประวัติการเดินทางที่ผ่านมาเท่านั้น

2) อุปกรณ์ติดตามรถแบบกึ่งออฟไลน์

อุปกรณ์ติดตามรถแบบนี้จะทำงานร่วมกับผู้ให้บริการโทรศัพท์มือถือ โดยตัวเครื่องจะมีช่องไว้ใส่ซิมการ์ดและจะส่งข้อมูลพิกัดกลับไปให้ผู้ใช้ผ่านระบบ SMS เมื่อผู้ใช้ร้องขอไปที่ตัวเครื่อง GPS Trackingโดยมีข้อดีคือจะไม่มีค่าบริการรายเดือน จะมีแต่ค่าใช้บริการของเครือข่าย โทรศัพท์มือถือ เช่น ค่าส่ง SMS

ข้อเสีย: คือความสะดวกในการใช้งาน เพราะผู้ใช้จะได้รับข้อมูลพิกัดมาเป็นตัวเลข โดยผู้ใช้ต้องเชื่อมต่อ internet และพิมพ์ตัวเลขพิกัดเข้าไป เพื่อทำการหาตำแหน่งปัจจุบันทุกครั้งที่ได้ SMS และ ตรวจสอบข้อมูลการเดินทางย้อนหลังไม่ได้

3) อุปกรณ์ติดตามรถแบบออนไลน์

ตัวอุปกรณ์จะรับข้อมูลพิกัดตำแหน่งปัจจุบันจากดาวเทียม และส่งไปเก็บที่เครื่องแม่จ่ายผ่านระบบ EDGE, GPRS และผู้ใช้สามารถเรียกดูตำแหน่งปัจจุบัน หรือประวัติการเดินทางรายงานการเดินทางแบบออนไลน์ได้ทั้งจากเครื่องคอมพิวเตอร์ และโทรศัพท์มือถือ และยังสามารถส่งคำสั่งต่างๆ กลับไปยังอุปกรณ์ติดตามรถได้ เช่น ให้ดับเครื่องยนต์ โดยมีข้อดีคือข้อมูลพิกัดจะถูกเก็บไว้ที่เครื่องแม่ข่าย ดังนั้นผู้ใช้จะสามารถเรียกดูข้อมูลตำแหน่งปัจจุบันได้แบบเวลาจริง ผ่านเครื่องคอมพิวเตอร์ หรือโทรศัพท์มือถือที่เชื่อมต่ออินเทอร์เน็ต ได้สามารถดูประวัติการเดินทางย้อนหลังและทำรายการต่างๆ ได้ผู้ให้บริการบางรายจะสามารถใช้เทคโนโลยีนี้ในการควบคุมระบบต่างๆ ของรถได้อีกด้วย เช่น ระบบไฟฟ้า ระบบน้ำมัน จึงสามารถใช้ความสามารถที่มีได้อย่างคุ้มค่าและข้อเสียคือจะมีค่าบริการรายเดือนหรือตามเงื่อนไขการให้บริการของแต่ละผู้ให้บริการแต่ละราย เพราะมีการส่งข้อมูลและเก็บข้อมูลต่างๆไว้บนแม่ข่ายตลอดเวลา

2.2.3 GPS Tracker TK-102

เป็นเครื่องติดตามด้วยจีพีเอสที่นำมาใช้ในปริญญานิพนธ์ฉบับนี้ โดยจะใช้งานผ่านการส่ง SMS

2.2.3.1 เริ่มต้นการใช้งาน

การเริ่มต้นให้เครื่องใช้งานทำได้โดยการส่ง SMS ไปที่เครื่องติดตามข้อความสำหรับส่งให้เครื่องเริ่มต้นทำงาน begin123456 (123456 เป็นรหัสผ่านที่ตั้งมาจากโรงงาน)

เมื่อตัวเครื่องติดตามได้รับข้อความเรียบร้อย ตัวเครื่องจะส่งข้อความกลับมาทันทีว่า "begin ok!" ถ้าตอบกลับมาว่า "begin failed" หมายถึงรหัสผ่านผิด

2.2.3.2 การกำหนดรหัสผ่าน

การเปลี่ยนรหัสใหม่ทำได้โดยส่งข้อความ password ตามด้วยรหัสผ่านเดิม เว้นวรรค แล้วตามด้วยรหัสผ่านตัวใหม่ ยกตัวอย่างเช่น รหัสเดิม 123456 รหัสใหม่ 789413 โดยรหัสผ่านต้องใช้จำนวน 6 หลักเท่านั้น ตัวอย่างข้อความสำหรับแก้ไขรหัสผ่าน: "password123456 789413"

2.2.3.3 การกำหนดหมายเลขหลัก

การกำหนดหมายเลขทำได้โดยส่งข้อความ admin+PASSWORD PHONENUMBER ตั้งเบอร์ที่อนุญาต ให้ถามตำแหน่งได้โดยสามารถมีได้ 5 เบอร์ โดยเบอร์แรก จะถูกใช้เป็นเบอร์สำหรับการแจ้งเตือนทาง SMS ตัวอย่าง admin123456 0840012544 (GPS Tracker จะส่ง SMS ของตำแหน่ง ถ้า 0840012544 เป็นคนโทรเข้า) และหาก 0840012544 เป็นเบอร์แรกใน 5 เบอร์ SMS แจ้งเตือนจะถูกส่งไปยัง 0840012544

2.2.3.4 การติดตามเป้าหมายแบบเวลาจริง

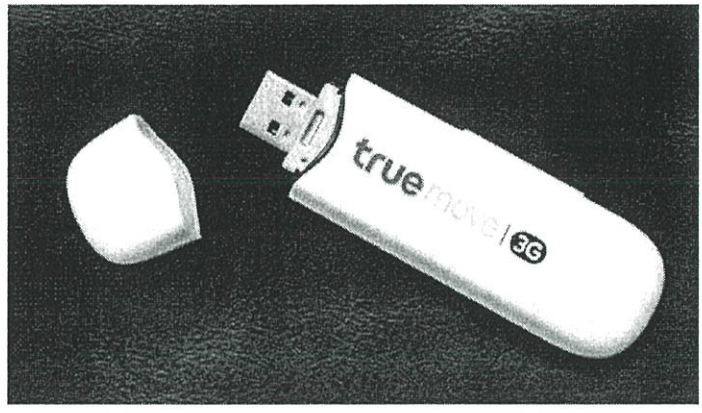
ถ้าไม่ได้กำหนดให้เลขหมายเฉพาะสำหรับติดตาม เครื่องติดตามก็จะรายงานตำแหน่งพิกัดแผนที่มายังเลขหมายนั้นทราบทันที แต่ถ้าได้กำหนดเลขหมายไว้ล่วงหน้าเลขหมายใดๆ นอกจากที่กำหนดไว้จะไม่ได้รับข้อมูลใดๆ

1) เพิ่มเลขหมายหลักโดย ส่งข้อความ admin เว้นวรรครหัสผ่าน เว้นวรรค เลขหมายที่ต้องการ

2) ลบเลขหมายหลักโดยส่งข้อความ noadmin เว้นวรรค รหัสผ่านเว้นวรรค เลขหมายที่ต้องการ

2.3 แอร์การ์ด (Air Card)

แอร์การ์ด (Air Card) คือ อุปกรณ์โมเด็มอย่างหนึ่งที่ใช้เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์ (Desktop หรือ Laptop) เข้าสู่เครือข่ายอินเทอร์เน็ตแบบไร้สายความเร็วสูง โดยผ่านโครงข่ายสัญญาณโทรศัพท์มือถือ ซึ่งในขณะที่เชื่อมต่อเข้าสู่เครือข่ายอินเทอร์เน็ตแล้วยังสามารถใช้โทรศัพท์โทรเข้าออกได้ในเวลาเดียวกัน เพราะระบบมีการใช้ช่องสัญญาณคนละช่องสัญญาณกัน แต่ใช้สถานีฐานเดียวกัน หรือทำหน้าที่เป็นแฟกซ์ไร้สายได้ด้วย



รูปที่ 2.3 แอร์การ์ด

2.3.1 ความแตกต่างระหว่าง แอร์การ์ด กับ ระบบ Wi-Fi

Wi-Fi คือ คุณสมบัติอันหนึ่งที่ทำให้เชื่อมต่ออินเทอร์เน็ตใดๆ ก็ได้แบบไร้สาย (Wireless LAN) ในระยะห่างไม่เกิน 100 เมตร จากตัวแม่ข่ายของ Wi-Fi นั้นๆ หากไม่มีตัวแม่ข่ายการสื่อสารข้อมูลก็จะทำไม่ได้

แอร์การ์ด คือ โมเด็มอย่างหนึ่งที่ใช้เพื่อเชื่อมต่อเครื่องคอมพิวเตอร์เข้าสู่เครือข่ายอินเทอร์เน็ต แบบไร้สาย โดยใช้สัญญาณโทรศัพท์มือถือ ซึ่งมีการเชื่อมสัญญาณเข้ากับสถานีฐานของโทรศัพท์มือถือ ทำให้เข้าถึงเครือข่ายอินเทอร์เน็ตได้ในที่ที่มีสัญญาณโทรศัพท์มือถือ

2.3.2 คุณสมบัติที่ควรมีในแอร์การ์ด

1) สามารถรองรับระบบปฏิบัติการได้หลากหลายระบบ เช่น Window Vista, Window XP, Window ME, Window 2000 หรือ Mac OSX ใช้งานโดยเสียบเข้ากับพอร์ตยูเอสบีได้

2) สามารถอัพเกรดเฟิร์มแวร์ได้ โดยใช้งานได้ทั้งกับเครือข่าย UMTS EDGE GSM

3) สามารถรองรับซิมการ์ดของเครือข่ายโทรศัพท์มือถือได้ทุกเครือข่าย รองรับระบบ 3G, EDGE Class 12 และ GPRS Class 12

4) รองรับการใช้งาน Voice หรือส่ง SMS

5) รองรับการใช้งานด้านโทรศัพท์และการใช้งานโทรสารควรเลือกแอร์การ์ดที่ใช้ไฟน้อย เพราะเหมาะสำหรับการใช้งานกับเครื่องคอมพิวเตอร์แบบพกพา

2.4 ภาษาซีชาร์ป

ภาษา C# (ซี-ชาร์ป) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่ใช้สำหรับเขียนโปรแกรมคอมพิวเตอร์ที่ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ในปัจจุบัน และเป็นภาษาโปรแกรมคอมพิวเตอร์ที่เหมาะสำหรับผู้ที่เริ่มต้นที่สนใจจะเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างยิ่ง ซึ่งภาษา C# ถูกพัฒนามาจากภาษา C++ (ซี-พลัสพลัส) และมีโครงสร้างแบบเชิงวัตถุ (object-oriented programming) โดยใช้ Visual Studio (วิชวล-สตูดิโอ) เป็นเครื่องมือสำหรับพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่ง Visual Studio เป็นเครื่องมือที่คอยอำนวยความสะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมคอมพิวเตอร์ได้ไม่ยากนักมีดังนี้

1) เป็นภาษาที่เขียนง่าย ไม่ซับซ้อนและเรียบง่าย เพราะคล้ายภาษา Java, ภาษา C และ ภาษา C++

2) เป็นภาษาโปรแกรมคอมพิวเตอร์ยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์ ภายใต้แนวคิด .NET Framework ซึ่งเป็นแนวคิดที่ได้รับความนิยมสูงที่สุดในปัจจุบัน

3) เป็นภาษาที่ถูกออกแบบมาให้ทำงานบน .NET Framework โดย .NET Framework เป็นรูปแบบในการพัฒนาโปรแกรมคอมพิวเตอร์สมัยใหม่ ซึ่งบริษัทไมโครซอฟท์ เป็นผู้พัฒนา ซึ่งคุณสมบัติที่สำคัญของ .NET Framework คือ ผู้ใช้งานสามารถใช้งานบนระบบฮาร์ดแวร์หรือ ระบบปฏิบัติการ ที่แตกต่างกันได้อย่างไม่มีปัญหา เช่น ระบบปฏิบัติการ Windows กับระบบปฏิบัติการ MacOS เป็นต้น ดังนั้น ผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมคอมพิวเตอร์ใหม่ๆ ได้โดยง่ายและรวดเร็ว

4) เป็นภาษาที่แข็งแกร่ง เพราะเป็นภาษาที่ได้มีการแก้ไขข้อบกพร่องบางอย่างของภาษา Java, ภาษา C และ ภาษา C++ เหล่านั้น ทำให้ ภาษา C# เป็นภาษาที่มีความสมบูรณ์ตามแบบฉบับของโครงสร้างแบบเชิงวัตถุ

2.4.1 โครงสร้างโปรแกรมภาษา C#

โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐานประกอบด้วยส่วนของโปรแกรมหลักแต่จะไม่มีส่วนของโปรแกรมย่อย (subroutine) โดยแสดงดังรูปที่ 2.4

```
namespace (1)
{
        class (2)
        {
                static void Main ( )
                {
                        (3)
                }
        }
}
```

รูปที่ 2.4 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน

จากรูปที่ 2.4 แสดงโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน โดยมีรายละเอียดดังนี้

หมายเลข (1) เป็นการระบุชื่อของ namespace ซึ่งใช้ในการกำหนดขอบเขตให้กับคลาสต่างๆ รวมถึงใช้ในการจัดโครงสร้างของโปรแกรมขนาดใหญ่ให้เป็นสัดส่วนอีกด้วย โดยเฉพาะอย่างยิ่งในการเขียนโปรแกรมคอมพิวเตอร์ที่ซับซ้อนโดยมีผู้เขียนโปรแกรมหลายคน นอกจากนี้ การกำหนด namespace ยังช่วยป้องกันปัญหาการตั้งชื่อคลาสหรือค่าคงที่อื่นๆ ซ้ำกันได้

หมายเลข (2) เป็นการระบุชื่อของคลาส

หมายเลข (3) เป็นการระบุพื้นที่สำหรับคำสั่งต่างๆ ที่ผู้เขียนโปรแกรมต้องการให้คอมพิวเตอร์ปฏิบัติตาม

นอกจากนี้ ในบางกรณี ผู้เขียนโปรแกรมสามารถที่จะไม่เขียนในส่วนของ namespace ได้ ถ้าโปรแกรมคอมพิวเตอร์นั้นมีขนาดเล็ก และ ไม่ซับซ้อนมากนัก ซึ่งการที่ไม่เขียนในส่วนของ namespace จะถือว่าคลาส ที่ถูกสร้างขึ้นมาอยู่ใน namespace กลาง โดยแสดงดังรูปที่ 2.5

```
class (2)
{
        static void Main ( )
        {
                (3)
        }
}
```

รูปที่ 2.5 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace

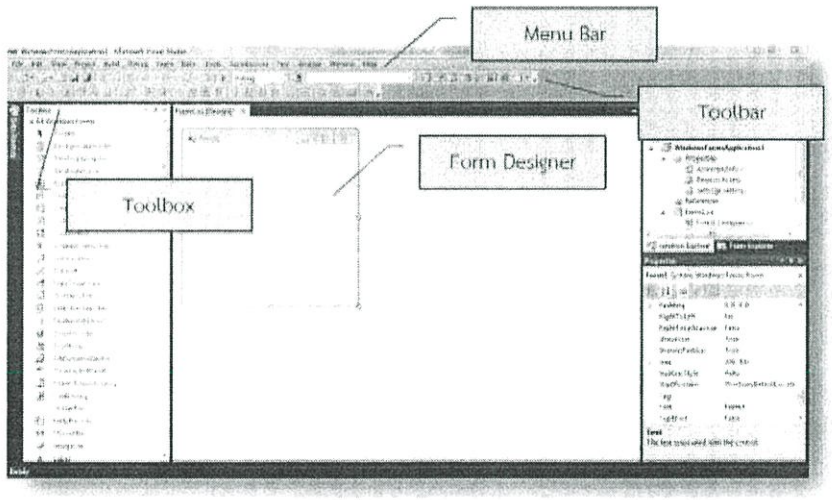
2.5 Visual Studio

Visual Studio (วิชวล-สตูดิโอ) เป็นโปรแกรมหรือเครื่องมือที่ช่วยในการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษา C# ซึ่งโปรแกรมนี้ได้พัฒนาขึ้นครั้งแรกโดยบริษัทไมโครซอฟท์ในชื่อของ Visual C# .NET (วิชวล-ซี-ชาร์ป-ดอตเน็ต)

รูปที่ 2.6 Visual Studio

2.5.1 เครื่องมือสำหรับเขียนโปรแกรม

แสดงแถบเครื่องมือการเขียนโปรแกรม ชื่อและการทำงานของแถบต่างๆ ดังรูปที่ 2.7



รูปที่ 2.7 หน้าจอแสดงองค์ประกอบหลักของโปรแกรม

2.5.1.1 เมนูบาร์ (Menu Bar) เป็นเมนูหลักที่รวบรวมคำสั่งควบคุมการทำงานของ Visual Studio 2010 โดยจัดเป็นกลุ่มคำสั่งแยกตามประเภทการใช้งาน และสามารถเรียกใช้ได้โดยใช้เมาส์คลิกจากเมนู หรือใช้คีย์ลัดจากคีย์บอร์ดก็ได้ ซึ่งมีกลุ่มคำสั่งเรียกใช้งานดังนี้

1) File กลุ่มคำสั่งเกี่ยวกับการสร้างโปรเจคใหม่ เปิดโปรเจค บันทึกโปรเจค ปิดโปรเจค

2) Edit กลุ่มคำสั่งที่ช่วยสร้างและแก้ไขการทำงานของโปรเจคให้ง่ายขึ้น เช่น Copy, Paste และ Undo เป็นต้น

3) View กลุ่มคำสั่งเกี่ยวกับการจัดการรูปแบบการแสดงผล IDE ของ Visual Studio

4) Project กลุ่มคำสั่งเกี่ยวกับการจัดการโปรเจค เช่น การเพิ่ม Form และ Reference เป็นต้น

5) Build กลุ่มคำสั่งเกี่ยวกับการคอมไพล์โปรเจค

6) Debug กลุ่มคำสั่งเกี่ยวกับการหาข้อผิดพลาดของโปรเจค

7) Team กลุ่มคำสั่งเกี่ยวกับการบริหารจัดการโครงการที่พัฒนาโปรแกรมคอมพิวเตอร์เป็นทีม

8) Data กลุ่มคำสั่งเกี่ยวกับการติดต่อกับฐานข้อมูล

9) Format กลุ่มคำสั่งเกี่ยวกับการออกแบบ Form

10) Tools กลุ่มคำสั่งเกี่ยวกับการใช้งานเครื่องมือต่างๆ

11) Test กลุ่มคำสั่งเกี่ยวกับการทดสอบการทำงานของโปรแกรม

12) Window กลุ่มคำสั่งเกี่ยวกับการจัดการรูปแบบการแสดงผล IDE ของ Visual Studio

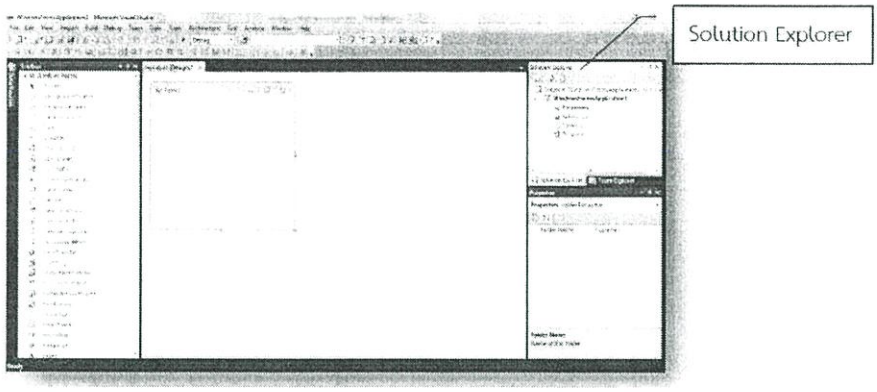
13) Help คำสั่งขอความช่วยเหลือจาก Visual Studio

2.5.1.2 ทูลบาร์ (Toolbar) เป็นการรวบรวมคำสั่งในเมนูบาร์บางคำสั่งที่มีการใช้งานบ่อยๆ มาแสดงไว้เพื่อให้สามารถเรียกใช้งานได้สะดวกขึ้น โดยสร้างเป็นปุ่มให้เรียกใช้งานได้ในคลิกเดียว

2.5.1.3 หน้าต่าง Toolbox เป็นหน้าต่างที่แสดงคอนโทรลและคอมโพเนนต์ต่างๆ เพื่อให้สะดวกในการสร้างโปรแกรมคอมพิวเตอร์ จึงมีการจัดแบ่งคอนโทรลและคอมโพเนนต์ต่างๆ ออกเป็นกลุ่มๆ ตามลักษณะการใช้งาน

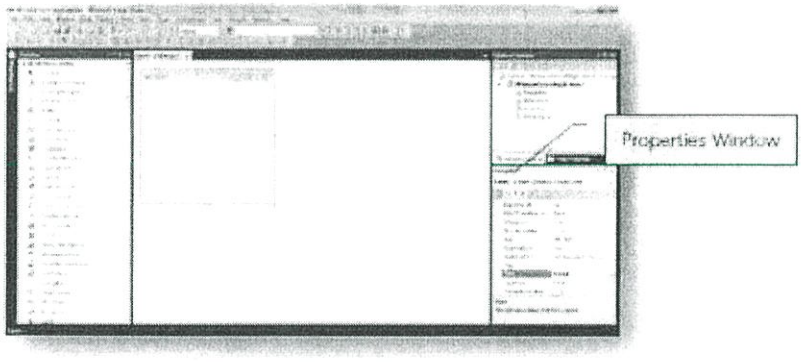
2.5.1.4 หน้าต่าง Form Designer เป็นหน้าต่างที่ใช้สำหรับการออกแบบหน้าตาของโปรแกรมคอมพิวเตอร์ โดยการลากคอนโทรลต่างๆ จาก Toolbox Window มาวางบน Form ตามต้องการ โดยที่ Visual Studio จะสร้างโค้ดการออกแบบให้โดยอัตโนมัติ ซึ่งเป็นความสามารถ ของ Visual Studio ที่ช่วยให้พัฒนาโปรแกรมคอมพิวเตอร์ได้ง่ายและมีประสิทธิภาพมากขึ้น

2.5.1.5 หน้าต่าง Solution Explorer เป็นหน้าต่างแสดงรายการของไอเท็ม (Item) ทั้งหมดที่มีอยู่ในโปรเจค เช่น Form, Module, Component และ Class เป็นต้น ซึ่งในมุมของ Visual Studio นั้น Solution คือ โปรแกรมคอมพิวเตอร์ที่ผู้เขียนโปรแกรมกำลังสร้างขึ้นมา โดยแสดงดังรูปที่ 2.8



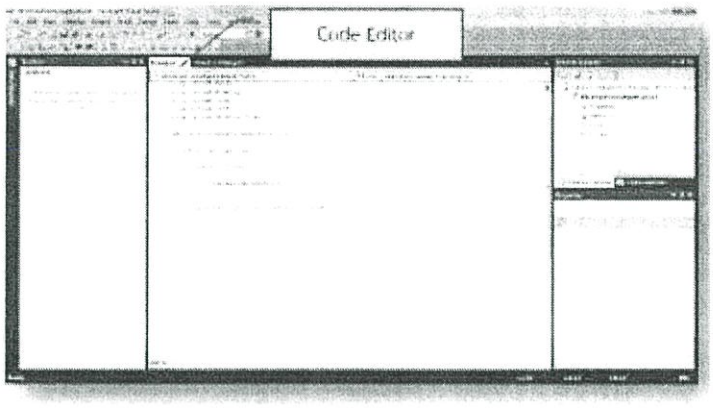
รูปที่ 2.8 หน้าต่าง Solution Explorer

2.5.1.6 หน้าต่าง Properties Window เป็นหน้าต่างที่ใช้แสดงและกำหนดคุณสมบัติของชิ้นส่วนต่างๆ ของโปรแกรมคอมพิวเตอร์ที่กำลังจะสร้างขึ้น เช่น การกำหนดลักษณะหน้าตาของโปรแกรมคอมพิวเตอร์ หมายถึง การกำหนดค่าให้กับชิ้นส่วนต่างๆ ที่ประกอบเป็นหน้าตาของโปรแกรมคอมพิวเตอร์ด้วย เช่น กำหนดแถบข้อความของหน้าต่าง, กำหนดสี และปรับขนาด เป็นต้น นอกจากจะกำหนดผ่านหน้าต่างนี้ ผู้ใช้ยังกำหนดผ่านการเขียนโปรแกรมคอมพิวเตอร์ในภาษา C# ได้อีกด้วย โดยแสดงดังรูปที่ 2.9



รูปที่ 2.9 หน้าต่าง Properties Window

2.5.1.7 หน้าต่าง Code Editor เป็นหน้าต่างที่ใช้สำหรับเขียนคำสั่งในภาษา C# เพื่อกำหนดการทำงานของโปรแกรมคอมพิวเตอร์ตามที่ต้องการ ซึ่งจะทำหลังจากที่ได้ออกแบบหน้าตาของโปรแกรมคอมพิวเตอร์เสร็จเรียบร้อยแล้ว โดยแสดงดังรูปที่ 2.10



รูปที่ 2.10 หน้าต่าง Code Editor

2.6 ระบบฐานข้อมูล Microsoft SQL

Microsoft SQL 6.5 เป็นอีกหนึ่งในระบบฐานข้อมูลที่กำลังได้รับความนิยม เพราะความต้องการระบบจัดการฐานข้อมูลเชิงสัมพันธ์ในระบบฐานข้อมูลมีความจำเป็นอย่างมาก ดังนั้นในหมู่ผู้ใช้ผลิตภัณฑ์ของ Microsoft จึงหันมาใช้ Microsoft SQL เพราะสามารถเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ และรันอยู่บน Window NT ด้วยเหตุที่ข้อมูลส่วนใหญ่ทั่วโลกเก็บไว้ในเครื่องที่ใช้ Microsoft Windows เป็น Operating System จึงทำให้เป็นการง่ายต่อ Microsoft SQL ที่จะนำข้อมูลที่อยู่ในรูป Windows Based มาเก็บและประมวลผล และ ประกอบกับการที่ราคาถูกและหาง่าย จึงเป็นปัจจัยหลักที่ทำให้เป็น Microsoft SQL เป็นระบบฐานขอมูลที่มักจะถูกเลือกใช้นอกจากนี้ข้อมูลส่วนใหญ่ใน โลกปัจจุบันอยู่ในรูปของ Microsoft Windows Format อีกด้วยจึงทำให้ Microsoft SQL 6.5 นั้นเป็นอีกหนึ่งทางเลือกที่เด่นชัดสำหรับองค์กรที่ต้องการความเป็นเลิศในเรื่องข้อมูล

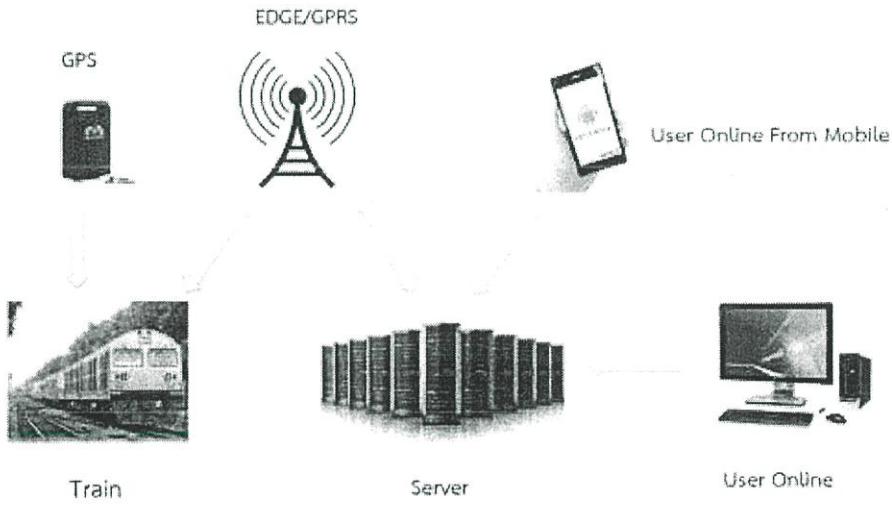
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบด้านแอปพลิเคชั่น

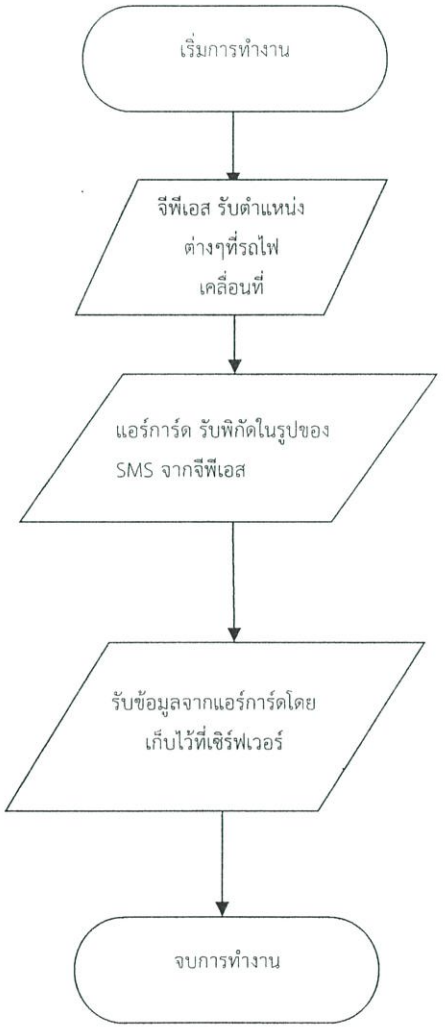
โดยนำระบบติดตามยานพาหนะด้วยดาวเทียมจีพีเอส ติดตามตำแหน่งของรถไฟ ณ เวลาปัจจุบัน ด้วยการทำงานที่ประสานกัน ของเครื่องติดตามจะรับสัญญาณจากดาวเทียมแล้วแจ้งพิกัด ละติจูด กับ ลองจิจูด มายังเครื่องแม่ข่าย ผ่านช่องทางการสื่อสารข้อมูลผ่านเครือข่าย แบบ EDGE/GPRS เพื่อเก็บค่าพิกัดดังกล่าว พร้อมเวลา ซึ่งระบบสามารถรายงานตำแหน่งของรถไฟดังกล่าวให้ผู้ใช้งานผ่านทางแอปพลิเคชั่น ดังรูปที่ 3.1



รูปที่ 3.1 บล็อคไดอะแกรมการทำงานรวม

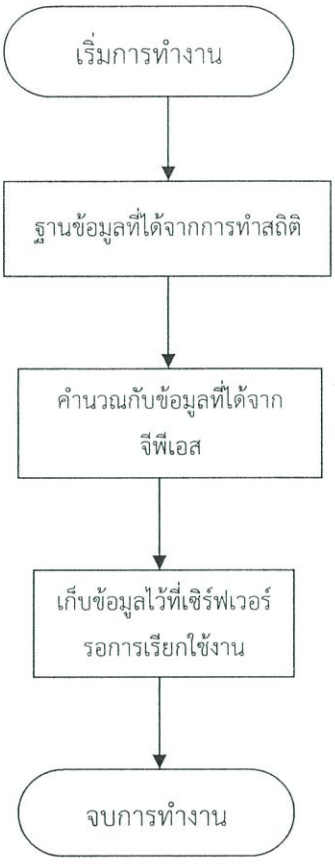
3.1.2 ขั้นตอนการทำงานทุกส่วน

3.1.2.1 ส่วนจัดเก็บข้อมูล โดยเริ่มจาก จีพีเอสรับข้อมูลพิกัดของจุดที่รถไฟอยู่ในแต่ละจุดๆ จากนั้นส่งข้อมูลให้แอร์การ์ดในรูปของ SMS จากนั้นแอร์การ์ดจะทำหน้าที่ส่งข้อมูลนั้นต่อไปเก็บไว้ยังเครื่องแม่ข่าย ดังรูปที่ 3.2



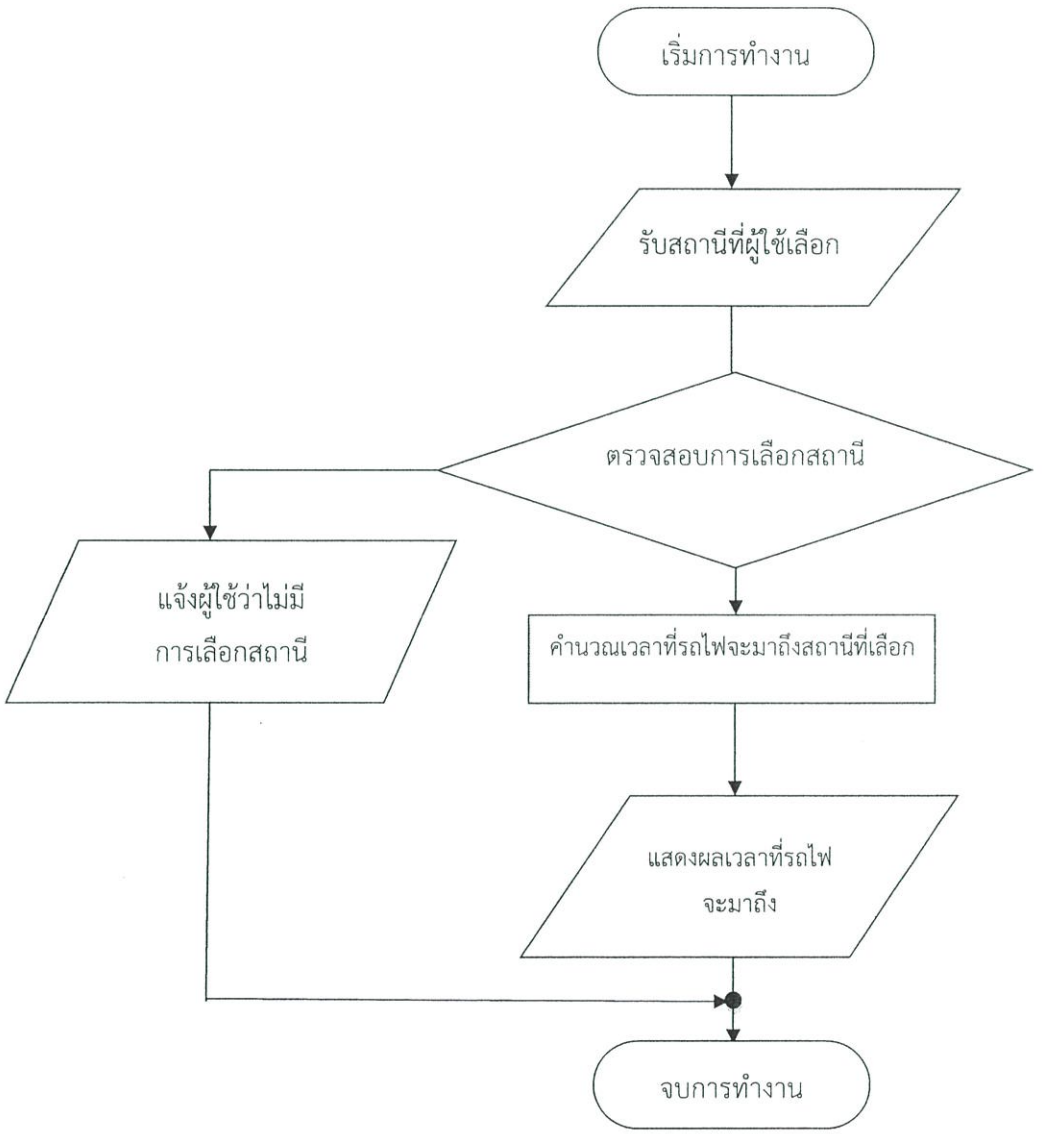
รูปที่ 3.2 ส่วนการจัดเก็บข้อมูล

3.1.2.2 ส่วนประมวลผลข้อมูล โดยนำข้อมูลที่ได้มาจากทั้งจีพีเอสและจากตำแหน่งของผู้ใช้งานนั้น มาใช้ในการคำนวณแล้วเก็บข้อมูลไว้ขณะรอส่งก่อนมีการเรียกใช้ ดังรูปที่ 3.3



รูปที่ 3.3 ส่วนประมวลผลข้อมูล

3.1.1.3 ส่วนแสดงผลข้อมูล เริ่มจากผู้ใช้งานเลือกสถานี เมื่อตัดสินใจเลือกสถานีแล้วจากนั้นจะส่งข้อมูลส่วนนี้ ไปคำนวณ เมื่อได้ผลการคำนวณจะบอกเวลาที่รถไฟจะมาถึงสถานีที่ผู้โดยสารใช้งานอยู่ โดยข้อมูลส่วนนั้นจะมาแสดงในแอปพลิเคชั่น ดังรูปที่ 3.4



รูปที่ 3.4 ส่วนแสดงข้อมูล

3.2 เครื่องมือที่ใช้ในการทดลอง

แบ่งออกเป็น 2 สองส่วนคือ ส่วนของซอฟต์แวร์และส่วนของฮาร์ดแวร์โดยประกอบไปด้วยเครื่องมือต่างๆ ดังนี้

3.2.1 ซอฟต์แวร์

1) Visual Studio

2) App Tracker

3.2.2 ฮาร์ดแวร์

1) GPS

2) แอร์การ์ด

3) โทรศัพท์ระบบแอนดรอยด์

3.3 การจัดเก็บผลการทดลอง

ในการจัดเก็บผลการทดลอง จะทดลองการทำงานของแอปพลิเคชั่นให้คำนวณเวลาที่รถไฟจะมาถึงสถานีที่ผู้ใช้เลือกเปรียบเทียบกับเวลาที่รถไฟเดินทางมาถึงจริงแล้วหาเวลาที่คำนวณผิดพลาด ในการทดลองจะให้ผู้ทดสอบรอรถไฟอยู่ที่สถานีและเปิดแอปพลิเคชั่นพร้อมทั้งเริ่มจับเวลาและบันทึกเวลาที่แอปพลิเคชั่นแสดงผล เมื่อรถไฟเดินทางมาถึงจึงหยุดการจับเวลา และบันทึกเวลาที่รถไฟใช้เดินทางมาถึง

3.4 การจัดเก็บสถิติเวลามาถึงของรถไฟ

ในการจัดเก็บสถิติเวลามาถึงของรถไฟ มีการจัดเก็บโดยจับเวลาเมื่อรถไฟเริ่มออกเดินทางจากสถานีกรุงเทพ เมื่อรถไฟจอดที่สถานีระหว่างทางจึงบันทึกเวลาไว้ จนถึงสถานีชุมทางฉะเชิงเทราทั้งหมด 5 ครั้งจากนั้นจึงนำเวลามาถึงแต่ละครั้งมาหาค่าเฉลี่ยได้ดังตารางที่ 3.1

ชื่อสถานี	เวลามาถึง ครั้งที่ 1 (นาที)	เวลามาถึง ครั้งที่ 2 (นาที)	เวลามาถึง ครั้งที่ 3 (นาที)	เวลามาถึง ครั้งที่ 4 (นาที)	เวลามาถึง ครั้งที่ 5 (นาที)	ค่าเฉลี่ย เวลา มาถึง (นาที)
กรุงเทพ	0	0	0	0	0	0
ยมราช	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	0
อุรุพงษ์	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	0
พญาไท	19	6	8	7	7	9.40
ราชปรารถ	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	ไม่จอด	0
มักกะสัน	4	4	4	5	6	4.60
อโศก	4	3	4	5	4	4.00
คลองตัน	5	3	5	6	6	5.00
หัวหมาก	9	4	5	6	5	5.80
บ้านทับช้าง	4	8	5	4	4	5
ซอยวัดลานบุญ	ไม่จอด	4	4	ไม่จอด	ไม่จอด	4.00
ลาดกระบัง	5	3	6	5	5	4.80
พระจอมเกล้า	5	5	6	7	6	5.80
หัวตะเข้	2	3	2	2	3	2.40
คลองหลวงแพ่ง	7	7	7	8	7	7.20
คลองอุดมชลจร	4	4	4	4	4	4.00
เปรง	4	5	4	4	5	4.40
คลองแขวงกลั่น	5	5	4	4	5	4.60
คลองบางพระ	3	5	3	4	3	3.60
บางเตย	4	3	3	3	4	3.40
ชุมทาง ฉะเชิงเทรา	8	7	7	8	8	7.60

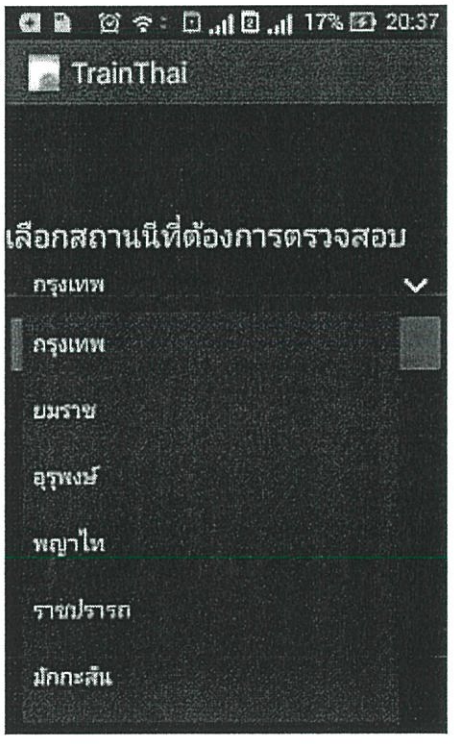
ตารางที่ 3.1 ค่าเวลาเฉลี่ยการเข้าชานชาลาของรถไฟ

บทที่ 4

ผลการทดลอง

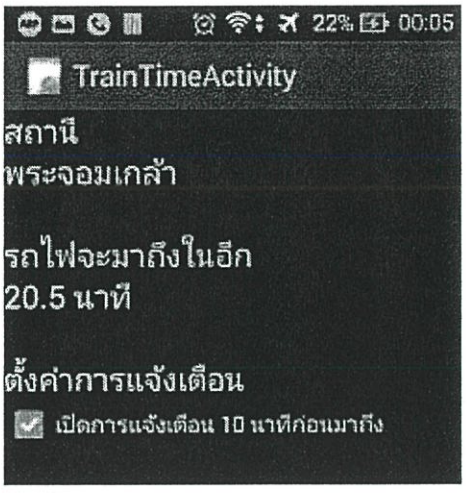
4.1 ผลการทดลอง

แสดงการทำงานในหน้าต่างๆของแอปพลิเคชั่นทั้งหมด จากที่ได้ทำการเก็บผลการทดลอง โดยเริ่มจากการกดเข้าสู่หน้าหลักของแอปพลิเคชั่นแล้ว จากนั้นผู้ใช้จะทำการเลือกสถานีที่ผู้ใช้อยู่ โดยกดเลือกสถานีตามแถบชื่อสถานีที่มีให้เลือก โดยชื่อสถานีทั้งหมดจากสถานีต้นทาง กรุงเทพหัวลำโพง ไปจนถึงสถานีปลายทาง ชุมทางฉะเชิงเทรา ดังรูปที่ 4.1



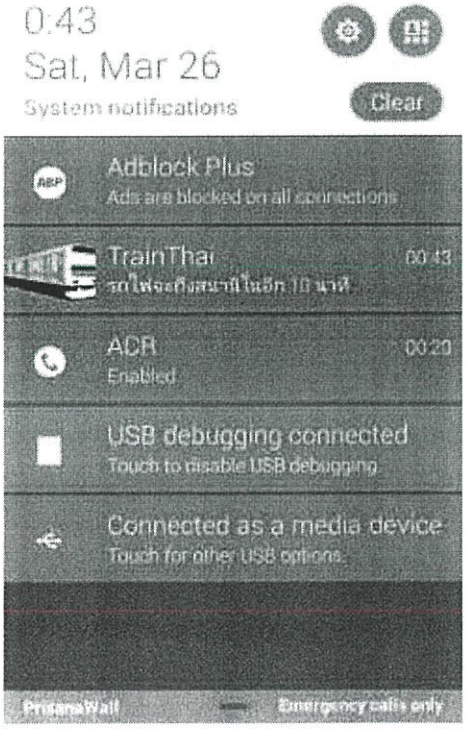
รูปที่ 4.1 การเลือกสถานีในการเดินทาง

เมื่อผู้ใช้เลือกสถานีแล้วแอปพลิเคชั่น จะเข้าสู่หน้าที่แสดงเวลาที่รถไฟจะมาถึงตัวผู้ใช้และในหน้านี้จะมีการตั้งค่าแจ้งเตือน โดยผู้ใช้สามารถเลือกได้ว่าจะรับการแจ้งเตือนหรือไม่ หากรับการแจ้งเตือน ก็จะสามารถเลือกรับการแจ้งเตือนได้ในเวลา 10 นาทีก่อนที่รถไฟจะมาถึงสถานี ดังรูปที่ 4.2



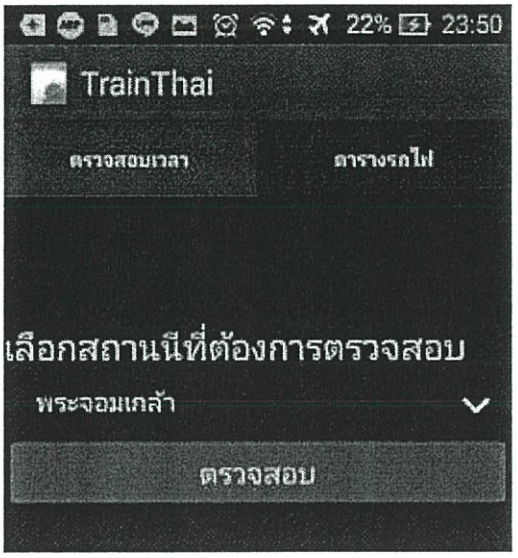
รูปที่ 4.2 การเลือกรับการแจ้งเตือน

หากผู้ใช้เลือกที่จะรับการแจ้งเตือนผู้ใช้สามารถกดออกจากหน้าแอปพลิเคชั่นนั้นได้แล้วรอรับการแจ้งเตือน ดังรูปที่ 4.3



รูปที่ 4.3 การแจ้งเตือนก่อนรถไฟมาถึง

หากผู้ใช้งาน ยังไม่ไปถึงสถานีรถไฟ แต่ต้องการวางแผนไว้ก่อนการเดินทาง ก็สามารถเลือกปุ่ม ตารางรถไฟ แล้วก็สามารถกดตรวจสอบ ตารางรอบรถไฟปกติได้ ดังรูปที่ 4.4



รูปที่ 4.4 การตรวจสอบรอบรถไฟในทุกขบวน

จากการจัดเก็บผลการทดลอง พบว่ารถไฟมาจริงตามที่จีพีเอสแสดง แต่เนื่องจากการคำนวณเวลานั้นได้มาจากการทำสถิติ ดังนั้นแล้วเวลาที่มาถึงของรถไฟจึงอาจคลาดเคลื่อน จากเวลาที่แจ้งเตือน ซึ่งจากการหาค่าเฉลี่ยของการมาไม่ตรงเวลานั้นพบว่า มีความคลาดเคลื่อนอยู่ที่ 5 -10 นาที

บทที่ 5

ผลการทดลอง

5.1 สรุปผล

แอปพลิเคชั่น แสดงเวลาเข้าสถานีของรถไฟสำหรับแอนดรอยด์ ซึ่งสามารถบอกได้ถึงเวลาที่รถไฟจะมาถึงสถานีที่ผู้ใช้งานกำลังใช้งานอยู่ได้ตามต้องการ โดยยังมีส่วนที่คลาดเคลื่อนของการทำงาน คือการบอกเวลาของรถไฟนั้นยังมีความผิดพลาดคือรถไฟมาไม่ตรงตามเวลาที่บอก แต่ยังอยู่ในค่าที่สามารถรับได้และมีส่วนเสริมคือ บอกตารางรถไฟรอบปกติที่ทางสถานีรถไฟได้แจ้งไว้อีกด้วย ทำให้เกิดความสะดวกสบายต่อการวางแผนการเดินทางมากยิ่งขึ้น

5.2 ข้อเสนอแนะ

1) นำไปปรับใช้กับระบบปฏิบัติการอื่นๆ ต่อไปเพื่อผู้ใช้งานที่หลากหลาย เช่น ระบบปฏิบัติการไอโอเอส

2) เพิ่มจำนวนการใช้งานกับรถไฟให้ได้หลายขบวน และหลายสายตามจำนวนเส้นทางที่มีอยู่จริงของรถไฟ

3) เพิ่มภาษาที่ใช้ในการใช้งานของแอปพลิเคชั่นเพื่อความสะดวกสบายแก่ชาวต่างชาติ

บรรณานุกรม

[1] นายทศพล ต้นสมบัติ. “แอนดรอยด์.” https://beerkung.wordpress.com

[2] นายรักพงศ์ แสงหงษ์. “GPS.” www.techincar.com.

[3] Iplate GPS Tracking Center. “คู่มือการใช้Tracker TK-102 โดยใช้คำสั่ง SMS.” http://www.iplateshop.com/store/article/view/_TK_102_SMS-135153-th.html.

[4] Muktahan. “แอร์การ์ดคืออะไร.” http://www.thesystem.co.th.

[5] KrooKung. “ภาษาC#.” https://sites.google.com/site/programmingm42/phasa-c.

[6] KrooKung. “VisualC#.” https://sites.google.com/site/programmingm42/visual-c.

ภาคผนวก ก

CODE

ชุดคำสั่ง

```
using System;
using Android.App;
using Android.Content;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;

namespace TrainThai
{
    [Activity(Label = "TrainThai", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        Fragment[] _fragments;
        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);
            ActionBar.NavigationMode = ActionBarNavigationMode.Tabs;
            // Set our view from the "main" layout resource
            SetContentView(Resource.Layout.Main);
            _fragments = new Fragment[]
            {
                new TrainCheckFragment(),
                new TrainTableFragment()
            };
            AddTabToActionBar("ตรวจสอบเวลา");
            AddTabToActionBar("ตารางรถไฟ");
        }

        void AddTabToActionBar(String tabname)
        {
            ActionBar.Tab tab = ActionBar.NewTab()
                                .SetText(tabname);
            tab.TabSelected += TabOnTabSelected;
            ActionBar.AddTab(tab);
```

```
        }
        void TabOnTabSelected(object sender, ActionBar.TabEventArgs tabEventArgs)
        {
            ActionBar.Tab tab = (ActionBar.Tab)sender;

            Fragment frag = _fragments[tab.Position];
            tabEventArgs.FragmentTransaction.Replace(Resource.Id.frameLayout1, frag);
        }
    }
}

using System;

using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Util;
using Android.Views;
using Android.Widget;

namespace TrainThai
{
    public class TrainCheckFragment : Fragment
    {
        public override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            // Create your fragment here
        }
```

```
        public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState)
        {
            // Use this to return your custom view for this Fragment
            // return inflater.Inflate(Resource.Layout.YourFragment, container, false);
            View view = inflater.Inflate(Resource.Layout.TrainCheck, null);
            Spinner spinner = view.FindViewById<Spinner>(Resource.Id.spinner1);
            Button button = view.FindViewById<Button>(Resource.Id.button1);

            var adapter = ArrayAdapter.CreateFromResource(
                  this.Activity, Resource.Array.stations_array,
Android.Resource.Layout.SimpleSpinnerItem);
adapter.SetDropDownViewResource(Android.Resource.Layout.SimpleSpinnerDropDownItem);
            spinner.Adapter = adapter;

            button.Click += delegate {
                var activity2 = new Intent(this.Activity, typeof(TrainTimeActivity));
                activity2.PutExtra("SelectStation", spinner.SelectedItemId);
                activity2.PutExtra("SelectStationName", spinner.SelectedItem.ToString());
                StartActivity(activity2);
            };

            return view;
        }
    }
                                                    }
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Util;
```

```
using Android.Views;
using Android.Widget;

namespace TrainThai
{
    public class TrainTableFragment : Fragment
    {
        public override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            // Create your fragment here
        }

        public override View OnCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState)
        {
            // Use this to return your custom view for this Fragment
            // return inflater.Inflate(Resource.Layout.YourFragment, container, false);
            View view = inflater.Inflate(Resource.Layout.TrainTable, null);
            return view;
        }
    }
                                                    }

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
```

```
using Android.Views;
using Android.Widget;

namespace TrainThai
{
    [Activity(Label = "TrainTimeActivity")]
    public class TrainTimeActivity : Activity
    {
        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);


            SetContentView(Resource.Layout.TrainTime);
            string selectstationname = Intent.GetStringExtra("SelectStationName") ?? "Data
not available";
            int selectstation = Intent.GetIntExtra("SelectStation",0);
            TextView textview2 = FindViewById<TextView>(Resource.Id.textView2);
            TextView textview4 = FindViewById<TextView>(Resource.Id.textView4);
            textview2.Text = selectstationname;
            //textview4.Text = TimeCal(selectstation).ToString() + " นาที";
        }
    }
                                                    }
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using Android.App;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
```

```
[assembly: AssemblyTitle("TrainThai")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("TrainThai")]
[assembly: AssemblyCopyright("Copyright ©  2016")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
[assembly: ComVisible(false)]

// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

Any raw assets you want to be deployed with your application can be placed in this directory (and child directories) and given a Build Action of "AndroidAsset".

These files will be deployed with you package and will be accessible using Android's AssetManager, like this:

```
public class ReadAsset : Activity
{
```

```
        protected override void OnCreate (Bundle bundle)
        {
                base.OnCreate (bundle);

                InputStream input = Assets.Open ("my_asset.txt");
        }
}
```

Additionally, some Android functions will automatically load asset files:

```
    Typeface tf = Typeface.CreateFromAsset (Context.Assets, "fonts/samplefont.ttf");
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="ApplicationName">TrainThai</string>
  <string name="stations_prompt">สถานี</string>
  <string-array name="stations_array">
    <item>กรุงเทพ</item>
    <item>ยมราช</item>
    <item>อุรุพงษ์</item>
    <item>พญาไท</item>
    <item>ราชปรารถ</item>
    <item>มักกะสัน</item>
    <item>อโศก</item>
    <item>คลองตัน</item>
    <item>หัวหมาก</item>
    <item>บ้านทับช้าง</item>
    <item>พระจอมเกล้า</item>
    <item>หัวตะเข้</item>
    <item>คลองหลวงแพ่ง</item>
    <item>คลองอุดมชลจร</item>
```

```
    <item>เปรง</item>
    <item>คลองแขวงสัน</item>
    <item>คลองบางพระ</item>
    <item>บางเตย</item>
    <item>ชุมทางฉะเชิงเทรา</item>
  </string-array>
</resources>

 [assembly: global::Android.Runtime.ResourceDesignerAttribute("TrainThai.Resource",
IsApplication=true)]

namespace TrainThai
{


            [System.CodeDom.Compiler.GeneratedCodeAttribute("Xamarin.Android.Bui
ld.Tasks", "1.0.0.0")]
            public partial class Resource
            {

                    static Resource()
                    {

            global::Android.Runtime.ResourceIdManager.UpdateIdValues();
                    }

                    public static void UpdateIdValues()
                    {
                    }

                    public partial class Array
                    {
```

```
                // aapt resource value: 0x7f050000
                public const int stations_array = 2131034112;

                static Array()
                {
global::Android.Runtime.ResourceIdManager.UpdateIdValues();
                }

                private Array()
                {
                }
        }

        public partial class Attribute
        {

                static Attribute()
                {

global::Android.Runtime.ResourceIdManager.UpdateIdValues();
                }

                private Attribute()
                {
                }
        }

        public partial class Drawable
        {

                // aapt resource value: 0x7f020000
```

```
            public const int Icon = 2130837504;

            static Drawable()
            {

global::Android.Runtime.ResourceIdManager.UpdateIdValues();
            }

            private Drawable()
            {
            }
        }

        public partial class Id
        {

            // aapt resource value: 0x7f060003
            public const int button1 = 2131099651;

            // aapt resource value: 0x7f060000
            public const int frameLayout1 = 2131099648;

            // aapt resource value: 0x7f060004
            public const int linearLayout1 = 2131099652;

            // aapt resource value: 0x7f060005
            public const int linearLayout2 = 2131099653;

            // aapt resource value: 0x7f060002
            public const int spinner1 = 2131099650;

            // aapt resource value: 0x7f060001
```

```
            public const int textView1 = 2131099649;

            // aapt resource value: 0x7f060006
            public const int textView2 = 2131099654;

            // aapt resource value: 0x7f060007
            public const int textView3 = 2131099655;

            // aapt resource value: 0x7f060008
            public const int textView4 = 2131099656;

            static Id()
            {

global::Android.Runtime.ResourceIdManager.UpdateIdValues();
            }

            private Id()
            {
            }
        }

        public partial class Layout
        {

            // aapt resource value: 0x7f030000
            public const int Main = 2130903040;

            // aapt resource value: 0x7f030001
            public const int TrainCheck = 2130903041;

            // aapt resource value: 0x7f030002
```

```
                public const int TrainTable = 2130903042;
                // aapt resource value: 0x7f030003
                public const int TrainTime = 2130903043;

                static Layout()
                {

global::Android.Runtime.ResourceIdManager.UpdateIdValues();
                }

                private Layout()
                {
                }
        }
        public partial class String
        {
                // aapt resource value: 0x7f040000
                public const int ApplicationName = 2130968576;

                // aapt resource value: 0x7f040001
                public const int stations_prompt = 2130968577;
                static String()
                {
        global::Android.Runtime.ResourceIdManager.UpdateIdValues();
                }

                private String()
                {
                }
        }
}
                        }
```

ภาคผนวก ข

DATASHEET

satellite communication.ch

GSM/GPRS/GPS TRACKER TK102-2 USER MANUAL

Content:

1. Summary
2. Applications
3. Hardware description
4. Specifications

5. User manual - Utilisation
   - 5.1. SIM card installation
   - 5.2. Battery and Charger
   - 5.3. Quick Start-up
   - 5.4. Initialization
   - 5.5. Change the password
   - 5.6. Authorization
   - 5.7. Auto track
   - 5.8. Voice Surveillance
   - 5.9. Modes switch between "track" and "monitor"
   - 5.10. Geo-fence
   - 5.11. Movement Alert
   - 5.12. Speeding Alert
   - 5.13. IMEI Check
   - 5.14. SOS Button
   - 5.15. Low battery alert
   - 5.16. Hidden number tracking
   - 5.17. SMS centre
   - 5.18. GSM ID
   - 5.19. Motion sensor
   - 5.20. SD card function
   - 5.21. GPRS setting

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch

www.satellite-communication.ch www.satellite-telephone.ch www.thuraya-rental.com www.iridium-rental.com
.... subsidiaries of Marine Electronic SA

satellite communication.ch

1 Summary

Operation is based upon existing GSM/GPRS network and GPS satellites, this product can locate and monitor any remote targets by SMS or GPRS.

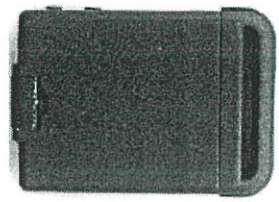
2 Applications

- Rental vehicle, boats / Fleet management etc
- Protection of capital goods and property
- Protect children / elderly / disabled / pets etc
- Provide peace-of-mind for businessmen
- Personnel management
- Covert Tracking
- Etc.

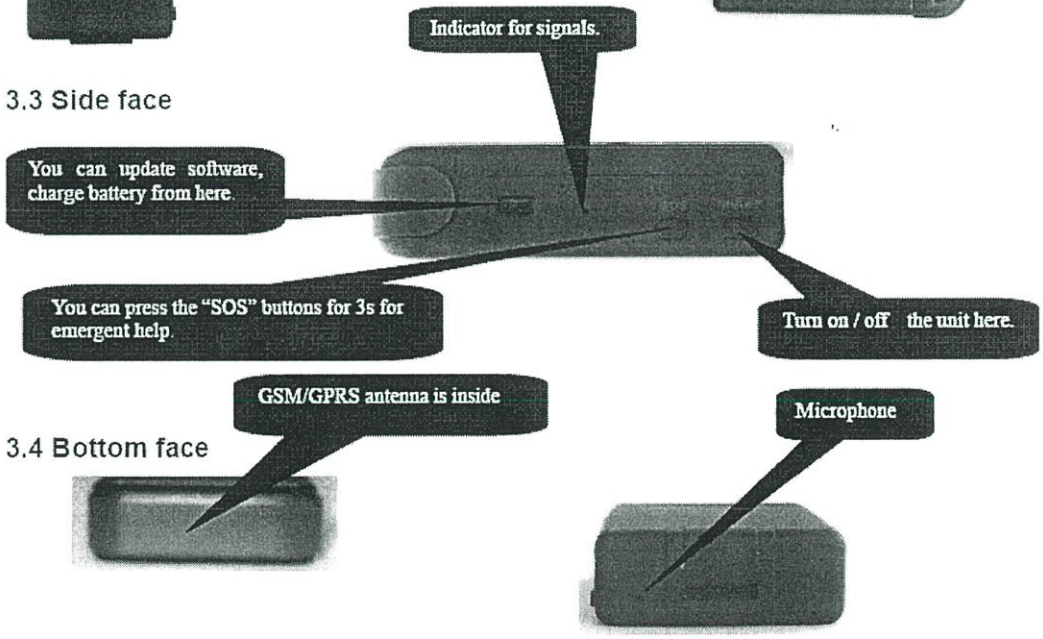
3 Hardware Description

3.1 Front face

3.2 Rear face



3.3 Side face



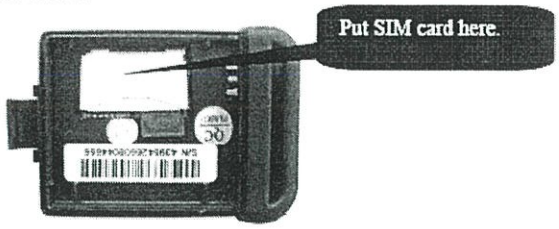
3.4 Bottom face

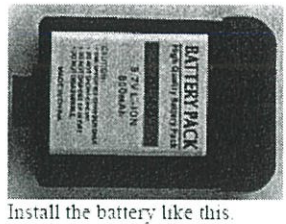
Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

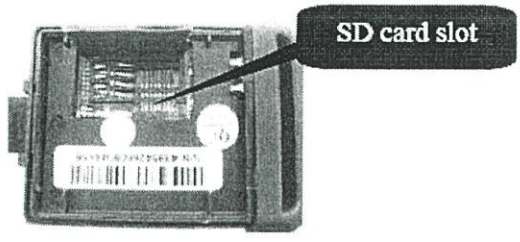
info@satellite-communication.ch

3.5 Interior





Install the battery like this.



Dim.	64mm x 46mm x 17mm(1.8"*2.5"*0.65")
Weight	50g
Network	GSM/GPRS
Band	850/1800/1900Mhz or 900/1800/1900Mhz
GPS chip	SIRF3 chip
GSM/GPRS module	Siemens MC55/Siemens MC56 or Simcom300/Simcom340
GPS sensitivity	-159dBm
GPS accuracy	5m
Time To First Fix	Cold status 45s Warm status 35s Hot status 1s
Car charger	12—24V input 5V output
Wall charger	110 - 220V input 5V output
Battery	Chargeable changeable 3.7V 800mAh Li-ion battery
Standby	48hours
Storage Temp.	-40°C to +85°C
Operation Temp.	-20°C to +55°C
Humidity	5%--95% non-condensing

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

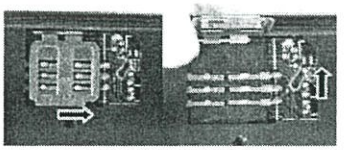
info@satellite-communication.ch

5 Utilisation

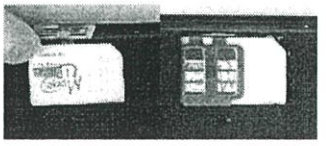
5.1 SIM card installation

Information: Make sure that there is **no call-forward** and **call display is on**, and **PIN code off**..

- 1 Open the rear cover, and pick out the battery.
- 2 Push the foil along and turn it left.



- Put the SIM card into the holder, and turn the foil right and fasten it in place.



5.2 Battery and Charger

Charge the battery to capacity in 8 -12 hours the first time. Use the battery and charger provided by the manufacturer. After first use, the battery can be charged fully in 3-5hours.

5.3 Start-up

- Insert the SIM card and battery
- Go outdoors
- Turn on the unit **outdoors** by pressing the "on/off" button until the LED is on.
- For first use, please initialize the unit and restore it to default setting. (Please refer to **5.4** for instruction.)
- In about 40seconds, the unit will acquire the GPS signals as well as the GSM signals. The LED will flash every 4 seconds when the unit is operative
- When the unit is operative, you can proceed to do all the following settings.

5.4 Initialization

This setting will perform a complete "factory reset"
Please note "lower case" for all SMS texts.

- *To activate send SMS:* "begin+password" by SMS to the unit, it will reply "begin ok" and initialize all the settings. **(default password: 123456)**

5.5 Change the password

If you want to maintain the password 123456 – skip this point

- *To change send SMS:* password+old password+space+new password - to change the password.
- **You can only use six digits as password!**

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch

5.6 Authorization

Information: Maximum 5 GSM numbers can be authorized. However when calling the tracker from a GSM phone only this number will receive the relevant information.
By SOS-alarm all authorized numbers will receive the coordinates!

- Calling the tracker 10 times, will automatically authorize this GSM number
- *To activate send SMS:* admin+password+space+GSM-number
- *Answerback:* "admin ok!" by SMS.
- *To de-activate send SMS:* noadmin+password+space+authorized number
- Make sure to enter country code of GSM e.g. +41 for Switzerland
- Important – the additional four authorized numbers must be programmed using the GSM with the first number – if the additional number is correctly programmed the answerback will be "admin OK"

5.7 Real-time polling

Information: To request precise location of the Tracker – you must call it
If there is no authorized number, then the Tracker will send the Geo-info to the GSM that is making the call.
If there is (are) one or more authorized number(s), then the Tracker will only respond to when called by one of these numbers.
When calling the Tracker it will ring 2 to 5 times, then it will hang up (hence no cost) and report a real-time Geo-info as below:

090723164830,+13616959853,GPRMC,214830.000,A,3017.2558,N,09749.4888,W,26.9,108.8,230709,,,A*61,F, Help me,imei: 359587013388627,05,264.5,F:3.79V,0,122,13990,310,01,0AB0,345A

= 090723164830=Serial no.(year, month, date, hour, minute, second)
= +13616959853=Authorized no.
= GPRMC,214830.000,A,3017.2558,N,09749.4888,W,26.9,108.8,230709,,,A*61=standard GPRMC sentences.
= F=GPS signal is full, if it indicate " L ", means GPS signal is low
= Help me=It is the SOS message, and this section is blank for tracking message
= IMEI=Identify #.of the tracker
= 05=Means you get 5 GPS fix (from 3 to 10)
= 264.5=Altitude
= F: 3.79V =Battery voltage
= 0=Means the tracker is NOT charged. 1 means the tracker is charged
= 122=The length of the GPRS string
= 13990=Checksum
= 310=MCC Mobile Country Code
= 01=MNC Mobile Network Code
= 0AB0=LAC Location area code
= 345A=Cell ID

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch

5.8 Auto-Track

Information: With Auto – Track the Tracker will report Geo-info at determined intervals to the authorized numbers for 5 times:

- *To activate send SMS:* t030s005n+password to the unit - (s: second, m: minute, h: hour). The setting must be in 3 digits and at maximum 255
- *Example:* t005m010n123456 = tracker will report Geo-info every 5 min. 10 times
  t001h020n123456 = tracker will report Geo-info every hour 20 times
  t001h***n123456 = tracker will report Geo-info every hour until switched off
- *To de-activate send SMS:* notn+password (e.g. Notn123456)
- Remark: the interval must not be less than 5s

5.9 Voice Surveillance

In this mode, the user can dial up the unit to monitor the surrounding noise transmitted through the microphone. (please refer to **5.10** for instruction).

5.10 Modes switch between "track" and "monitor"

Information: the default mode is "track". To hear (listen) the tracker has to be set to monitor mode.

- *To activate send SMS:* monitor+password
- *Answer back:* "monitor ok!"
- *To de-activate & restore tracker mode send SMS:* tracker+password, it will reply
- *Answer back*: "tracker ok!"

5.11 Geo-fence

Information: It is possible to set up a geo-fence for the unit to restrict its movements within a specified area. To set the area it is necessary to first program the latitude & longitude of the upper left corner. Followed by the latitude & longitude of the bottom right corner. This will create a square = Geo-fence.
The tracker should be inside of the Geo-fence for 3 to 10 minutes prior to movement. The Geo-zone should not be too small as the accuracy or the GPS signal may cause false alarm.
This function will be out of effect after the unit moves outside the district.

Geo information (latitude & longitude) can be obtained:
http://www.netzwelt.de/geocoding/index.html
http://www.mapsofworld.com/utilities/world-latitude-longitude.htm
http://www.gorissen.info/pierre/maps/googleMaplocationv3.php
http://pagesperso-orange.fr/universimmedia/geo/loc.htm
http://www.travelmath.com

- *To activate send SMS:*
  stockade+password+space+latitude,longitude,latitude,longitude
- *Example* stockade123456 48.389603,10.9122170;48.376549,10.927706
- *To de-activate send SMS:* nostockade+password

5.12 Movement alert

Information: An alarm is sent when the Tracker is moved from a specific location. To avoid false alarm the tracker has to be moved ca. 500m before the alarm is released The Tracker has to remain in the location for 3 to 10 minutes before movement. This setting remains after the tracker is switched off and on
This function will be out of effect after the unit moves outside the district.

- *To activate send SMS:* move+password
- *Answerback:* "move ok!".
- *Alarm if moved* "Move" along with a Geo-info to the authorized numbers.
- *To deactivate:* nomove+password

5.13 Speeding alert:

Information: Set up: This function will advice you when the tracker is moving faster than a predetermined speed

- *To activate send SMS:* speed+password+space+080 (the speed is 80km/h),
- *Answerback*: "speed ok!"
- If speed exceeds 80 km/h, the unit will send SMS speed+080+Geo-info to the authorized numbers. It will alarm one time in each setting.
- *To deactivate:* nospeed+password

5.14 IMEI checking:

- Send SMS imei+password to the unit to check the IMEI number.

5.15 OS button

Information: The SOS button can be used to send an alarm. Press the SOS for 3 second, it will send "help me!+ Geo-info" to all the authorized numbers every 3 minutes.

- *To deactivate:* It will stop sending such a SMS when any authorized number reply SMS "help me!" to the tracker.

5.16 Low battery alert

Information: When the battery is almost empty (below 3.7V) the tracker will send an alert SMS every 30 minutes: low battery+Geo-info.

5.17 Hidden number tracking

Information: Irrespective what number you use to call the tracker, the tracker will reply SMS with the hidden number&Geo-info to all the authorized numbers.

- *To activate send SMS:* "hide number" to set up the hidden number.
- *To deactivate:* "nohide number" to delete the hidden number.

5.18 Time Zone

Information: Every Geo-Info SMS will show date & time – if this information is incorrect you have to set it.

- *To activate send SMS:* time+space+zone+password+timezone
- *Example:* time zone 123456 1 (time zone 1 for Switzerland)
- *Answerback:* "Time OK!

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch

5.19 SMS Center

Information: All coordinates will be sent to a determined GSM number

- *To activate send SMS:* adminsms+password+space+GSM number - to set the SMS center.
- *Example:* adminsms +4179xxxxxxxxxx
- *Answerback:* "ADMINSMS OK!
- *To deactivate:* noadminsms123456

5.20 GPRS setting

Information: The tracker can also be used over the GPRS. In order to use GPRS function, the user needs to set IP, Port and APN (access point name) by either cell phone, or specific PC software. We will be setting up an Online-portal for this purpose.

- *To activate an IPaddress and port send SMS:* adminip+123456+space+IP address+space+port.
- *Example:* adminip123456 80.22.38.12.1000
- *Answerback:"* adminip OK"
- *To deactivate:* noadminip123456

Set APN

- *To activate an APN send SMS:* apn+123456+apn name
- *Example:* apn+123456+swisscom
- *Answerback:* "APN ok"
- *To deactivate:* noadminip123456

As soon as the IP & Port & APN is set up, the user can make a call or send SMS to tracker. The tracker will hang up the telephone and report the coordinates to IP address.

The message sent by tracker to IP address is as following:

- Serial number + authorized number + GPRMC + GPS signal indicator + command + IMEI number + CRC16 checksum.
  - = 090723164830=Serial no.(year, month, date, hour, minute, second)
  - = +13616959853=Authorized no.
  - = GPRMC,214830.000,A,3017.2558,N,09749.4888,W,26.9,108.8,230709,,, A*61=standard GPRMC sentences.
  - = F=GPS signal is full, if it indicate " L ", means GPS signal is low
  - = Help me=It is the SOS message, and this section is blank for tracking message
  - = IMEI=Identify no.of the tracker
  - = 05=Means you get 5 GPS fix (from 3 to 10)
  - = 264.5=Altitude
  - = F: 3.79V =Battery voltage
  - = 0=Means the tracker is NOT charged. 1 means the tracker is charged
  - = 122=The length of the GPRS string
  - = 13990=Checksum
  - = 310=MCC Mobile Country Code
  - = 01=MNC Mobile Network Code
  - = 0AB0=LAC Location area code
  - = 345A=Cell ID

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

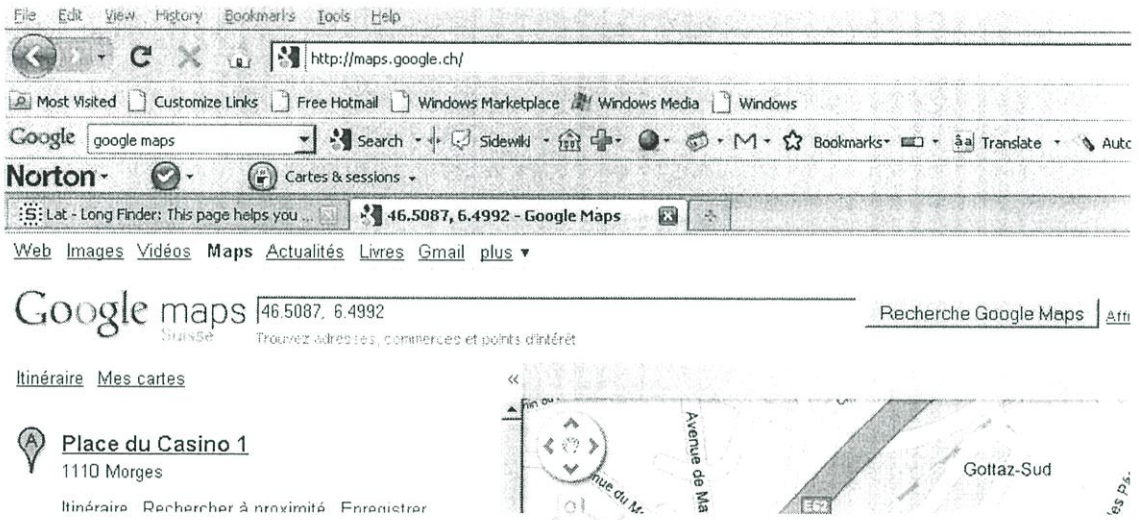
info@satellite-communication.ch

5.21 Goggle Maps and SMS

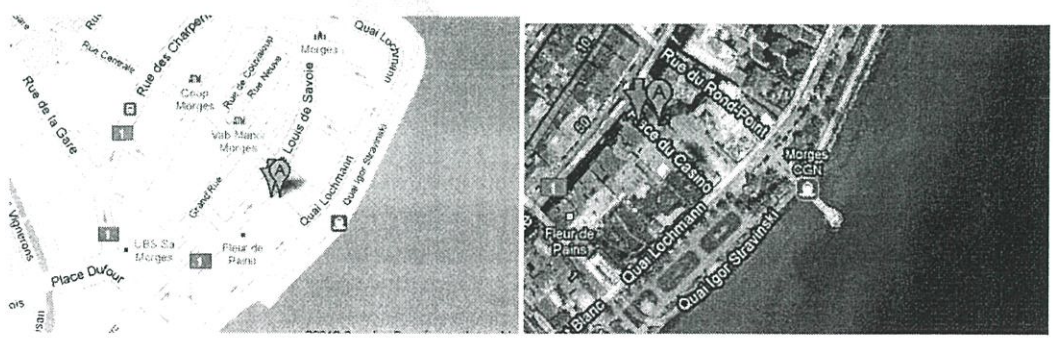
The SMS will be received in following format:
Lat: 46.5110980 long: 6.4999560 speed:043.0 20/01/10 13:42 bat:F imei 32483234343
Lat = latitude // Long = longitude // speed = speed in km/hr // date // time
// Bat F = full, L = low / signal F = full, L = low

Go to http://maps.google.com 1st enter "lat" 2nd enter "long" data (the other data such as speed etc are irrelevant for tracking)

For example: 46.5087, 6.4992



You can then choose between map or satellite image



Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch

satellite communication.ch

Command	Description	SMS format	Example
Password	Change password	Password+old pw+space+ new pw	password123456 000134
Admin	Authorization	Admin+password+space+GSM#	admin123456 +4179xxxxx
adminsms	SMS center	Adminsms+pw+space+SMScenter#	adminsms123456 +4179xxxxx
Adminnip	IP & PORT (GPRS)	Adminip+pw+space+IP+space+port	andminip123456 83.37.1.1.899
Tsn	Auto Track	T030s005n+password	t030s005123456
Tracker	Tracking mode	Tracker+password	tracker123456
Monitor	Listening mode	Monitor+password	monitor123456
Fencing	Geo-fence	Stockade+pw+space+speed	stockade123456 48.567185, 10.051248 ; 48.555666,10.777888
move	Movement alarm	Move+password	move123456
speed	Speed alarm	Speed+password+space+speed	Speed123456 080
Begin	Initilisation	Begin+password	begin123456
IMEI	IMEI checking	IMEI+password	imei123456
Help me	SOS alarm	Press and hold SMS button 3 to 5 sec	
Tracking	Activate tracking in an area where there is no GPS signal	Tracking + password	tracking123456
APN	APN initialisation	apn+password+space+APN	apn123456 swisscom
APN user	APN user	Apnuser+pw+space+username	apnuser123456 joe
APN pw	APN password	Apnpasswd+pw+space+pw	Apnpasswd123456 abc123

Notes:

- The GPRS supports TCP or UDP protocol, so please confirm whether your GPRS webserver supports TCP or UDP.
- Use a SIM card that has GPRS function.
- Pls use the following SMS to set up GMT time in the SMS for the tracker time zone+password+space+ your location GMT TIME
- Pls use the following SMS to set up YOUR website in the SMS for the tracker home+password+space+ your WEBSITE
  home+password+space+space (if there is no website in the SMS)

Marine-Electronic SA
41, rue Louis de Savoie
CH – 1110 Morges
Switzerland

Tel. +4121 8262111
Fax +4121 8262114

info@satellite-communication.ch